Hekstra reproduction

Owen

21 July 2015

Introduction

This paper concerns whether population dynamics have a common statistical structure, in the same way that the frequency distribution of earthquake strengths has a characteristic strucutre. Important for this is understanding the role of historical contingency, and this is addressed in the paper also.

The data analysed comes from closed ecosystems containing E.coli, Chlamy., and Tetrahymena. Abundances were estimated via selective plane microscopy, allowing accurate and non-invasive estimates for months with high temporal resolution. The closed ecosystems were 3ml in volume held in fluorimetric cuvettes. Organisms in the ecosystes have been observed to persiste for over 1000 days.

Through time, the ecosystems developed spatial heterogeneity and phenotypic changes, such as large Tetrahymena (that may have beeb able to consume algae) and filamentous E.coli and colonies that may have been resistant to consumption by Tetrahymena. Replicates differed from each other in the development of this complexity.

There were over 50 replicates distributed across temporal blocks. Observations of density were made over ~100 days, every day to every eight weeks.

The data

The authors will provide the data to Owen, but have not given permission to make it public. Owen will put it on his group server, access to which is controlled by University of Zurich administrative systems (not by Owen directly).

In order to minimise the chance of us leaking the data, please follow these rules: - Do not make a copy of the data. Absolutely. Do. Not. - From within R, read the data from the group server. This will require you to be connected to it. If you're off site, you will need to VPN in. - Do not perform any save() or write() operations in R. (We may amend this if analyses are intensive and we need to save intermediate steps. However, these should be stored on the server, and not on local machines.)

This is obviously a bit of hassle compared to if we could make the data public, but we cannot, so must adhere to this practice.

These rules / guidelines are provided and discussed in the wiki.

Text from Hekstra:

> Basis Curated_workspace_20091115.mat in Matlab/Work on "SecondBackup" (LeiblerLab).
>
> Organization for variables: G_S: Group of ecosystems (A-F) and Species (Algae, Bacteria, Ciliates) The first experiment consisted of what is called group F. The second experiment of groups A-E, with A measured twice per week, B once per week, etc.
>
> Within each table, the entries are organized as follows

| —- system 1 —- | | —- system 2 —- | | —- system 3 —- | ...
|---|
| time CPF NF time CPF NF time CPF NF ... |
| ... ... |

With each row a measurement day; CPF = counts per frame; NF: number of frames. Density = CPF * calibration factor. Calibration factors convert counts per frame to cells per mL (not necessary for most analysis): coeff_A = 10^3.92; coeff_B = 10^5.15; coeff_C = 10^4.36;

Total counts: CPF * NF, used for estimation of measurement error (see below).(Note to self: For F_B, I removed the columns tracking alga-related fluorescence "leakage" > into the red channel to conform to the format of the other data sets.

Ecosystems excluded because of excessive leakage (>0.1 mg/day) [Bootstrap_sigma_20090714_2.m]:
Group A: 7, 9 Group B: 2, 8, 9
Group C: 2, 6 Group D: 10 Group E: 5, 10 Group F: none.

For some analysis, replicate 4 of set F was excluded (this ecosystem was used for a number of other experiments after day 70).

Grouping of ecosystems with approximately the same (sub) measurement schedule. Group A, rows (measurement days) 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 21, 23, 25, 26 Group B rows 2-15. Group F: rows: 4, 11, 18, 25, 31, 37, 42, 46, 50, 54, 58, 59, 59, 60. (time point 59 was duplicated for this analysis)

Calculated Escape Rates Motivation: the counts of individuals are not entirely independent between time points as cells often spend a few seconds within the observation volume. A correction is > thus necessary with respect to Poisson counting statistics.

Probability that object (identified as A, B or C, respectively) is not observed in the next frame (at 1 Hz): p_e_A = 0.237; p_e_B = 0.3; p_e_C = 0.9;

Correlation times are defined as 1/p_e. The number of independent counts goes as N*p_e. The error model is described in my thesis as well (https://dspace.rockefeller.edu/bitstream/10209/413/1/DoekeHekstraFinalThesis.pdf).

Other • Figure 4: to determine correspondence of eigenvectors between time points, I worked backwards in time, using the dot products of (normalized) eigenvectors as a measure of their similarity. • Treatment of zero counts: as 0.5 counts over the total number of frames.

Import and tidy the data

Read the data and tidy:

```
library(tidyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:stats':
##
##     filter, lag
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(stringr)
library(ggplot2)

library(boot)
library(Hmisc)
```

```
## Loading required package: grid
## Loading required package: lattice
##
## Attaching package: 'lattice'
##
## The following object is masked from 'package:boot':
##
##     melanoma
##
## Loading required package: survival
##
## Attaching package: 'survival'
##
## The following object is masked from 'package:boot':
##
##     aml
##
## Loading required package: Formula
##
## Attaching package: 'Hmisc'
##
## The following objects are masked from 'package:dplyr':
##
##     combine, src, summarize
##
## The following objects are masked from 'package:base':
##
##     format.pval, round.POSIXt, trunc.POSIXt, units
```

```
revised <- read.csv("/Volumes/Petchey/Gr_Petchey/6 Secure data/Hekstra_and_Leibler_2012_Cell/revised.csv

## Create variable names
var_names <- c("Block_Species",
               paste(rep(c("Time", "CPF", "NF"), 10),
                     rep(1:10, each=3), sep="-"))
colnames(revised) <- var_names

## Remove empty rows
revised <- filter(revised, Block_Species!="")


## Stack the data
dd <- select(revised, 1:4)
dd$Ecosystem <- 1
names(dd) <- c("Block_Species", "Time", "CPF", "NF", "Replicate")
for(i in 2:10) {
  sub_data <- select(revised, 1, (3*(i-1)+2:4))
  sub_data$Replicate <- i
  names(sub_data) <- c("Block_Species", "Time", "CPF", "NF", "Replicate")
  dd <- rbind(dd, sub_data)
}

## Split the Group_Species data
dd <- separate(dd, Block_Species, c("Block", "Species"), sep="_")
```

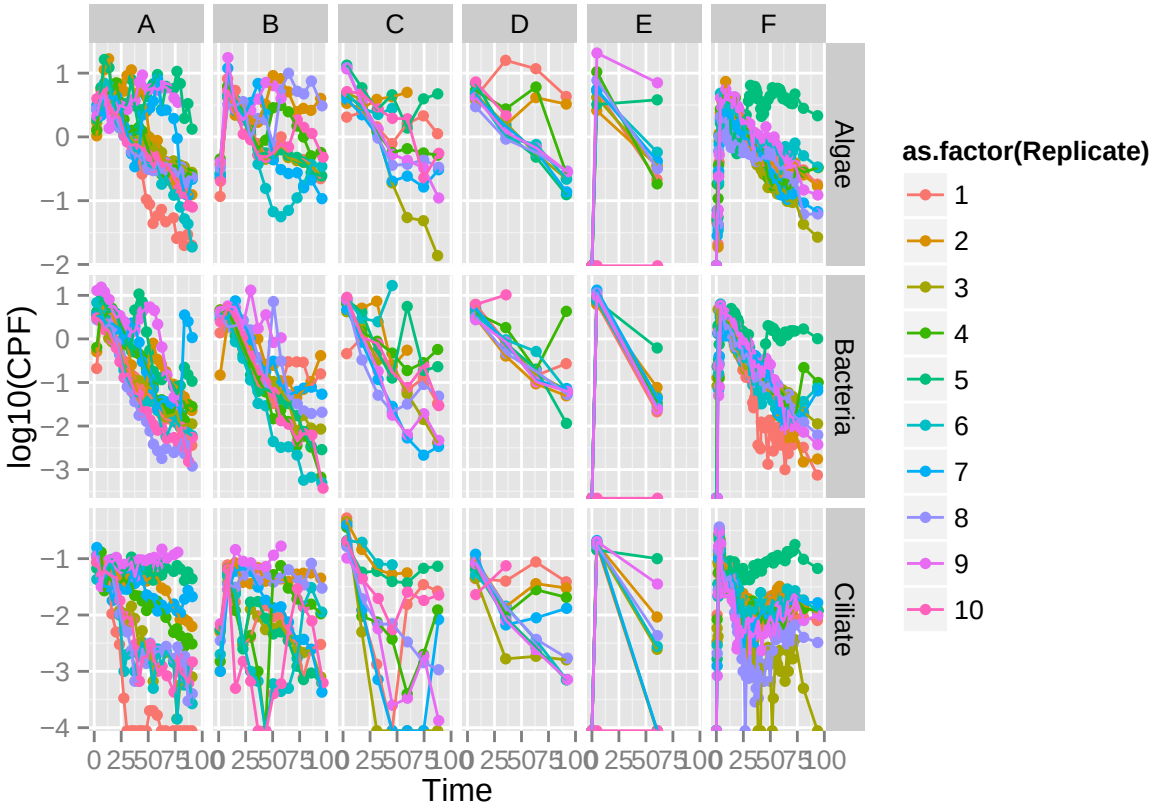
```
## Full species name
dd$Species <- ifelse(dd$Species=="A", "Algae",
                     ifelse(dd$Species=="B", "Bacteria", "Ciliate"))

#str(dd)
```

Try a preliminary graph:

```
ggplot(dd, aes(x=Time, y=log10(CPF), col=as.factor(Replicate))) +
  facet_grid(Species~Block, scales="free_y") +
  geom_point() + geom_line()
```

```
## Warning: Removed 14 rows containing missing values (geom_point).

## Warning: Removed 15 rows containing missing values (geom_point).

## Warning: Removed 15 rows containing missing values (geom_point).

## Warning: Removed 12 rows containing missing values (geom_point).

## Warning: Removed 60 rows containing missing values (geom_point).

## Warning: Removed 14 rows containing missing values (geom_point).

## Warning: Removed 15 rows containing missing values (geom_point).

## Warning: Removed 15 rows containing missing values (geom_point).

## Warning: Removed 12 rows containing missing values (geom_point).

## Warning: Removed 61 rows containing missing values (geom_point).

## Warning: Removed 14 rows containing missing values (geom_point).

## Warning: Removed 15 rows containing missing values (geom_point).

## Warning: Removed 15 rows containing missing values (geom_point).

## Warning: Removed 12 rows containing missing values (geom_point).

## Warning: Removed 60 rows containing missing values (geom_point).

## Warning: Removed 14 rows containing missing values (geom_path).

## Warning: Removed 15 rows containing missing values (geom_path).

## Warning: Removed 15 rows containing missing values (geom_path).
```

```
## Warning: Removed 12 rows containing missing values (geom_path).

## Warning: Removed 14 rows containing missing values (geom_path).

## Warning: Removed 15 rows containing missing values (geom_path).

## Warning: Removed 15 rows containing missing values (geom_path).

## Warning: Removed 12 rows containing missing values (geom_path).

## Warning: Removed 14 rows containing missing values (geom_path).

## Warning: Removed 15 rows containing missing values (geom_path).

## Warning: Removed 15 rows containing missing values (geom_path).

## Warning: Removed 12 rows containing missing values (geom_path).
```




```
# NaN in CPF column replaced by NAs
dd$CPF <- ifelse(is.nan(dd$CPF), NA,  dd$CPF)

# > Calibration factors convert counts per frame to cells per mL (not necessary for most analysis):
# > coeff_A = 10^3.92;  coeff_B = 10^5.15;  coeff_C = 10^4.36;

#Density = CPF * calibration factor.
```

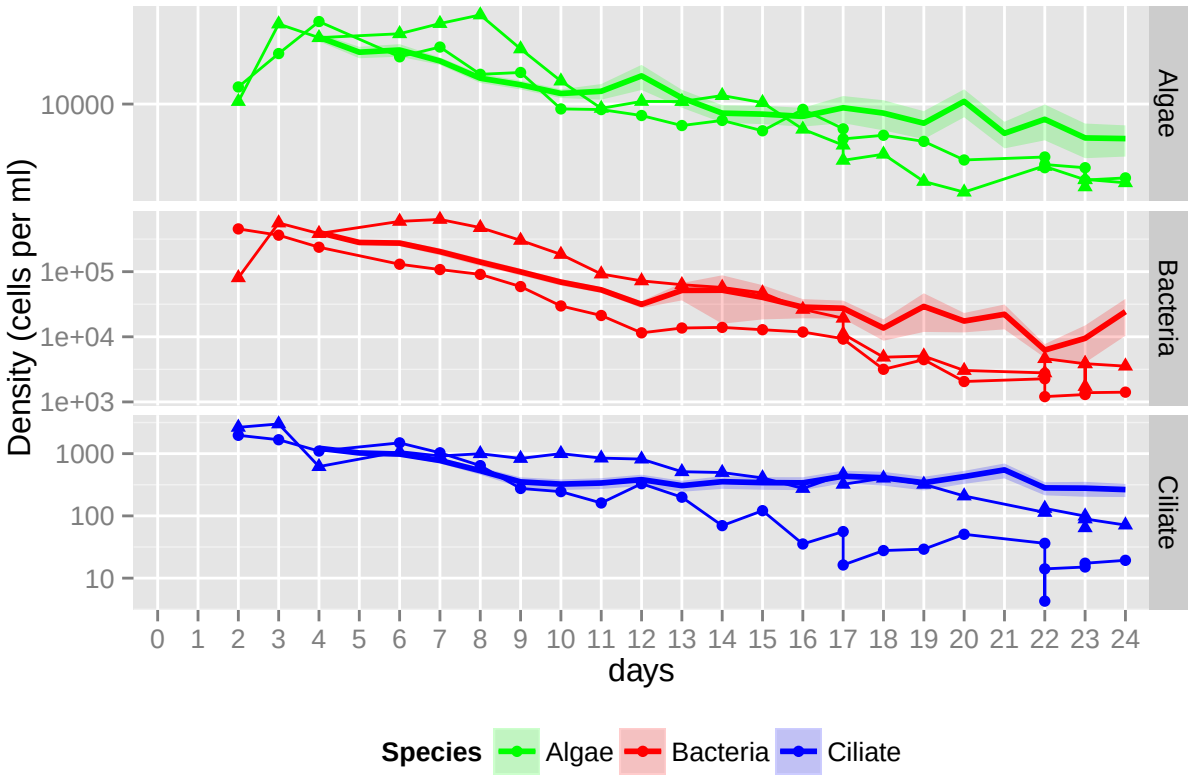
```r
dd <- dd %>% mutate(coef = ifelse(Species == "Algae", 10^3.92,
                            ifelse(Species == "Bacteria", 10^5.1, 10^4.36)),
                            density = CPF * coef)


# Ecosystems to be excluded because of excessive leakage (>0.1 mg/day)
# > Group A:    7, 9
# > Group B:  2, 8, 9
# > Group C:    2, 6
# > Group D:    10
# > Group E:    5, 10

dd_clean <- subset(dd, !((Block == "A" & Replicate %in% c(7,9)) |
            (Block == "B" & Replicate %in% c(2,8,9)) |
            (Block == "C" & Replicate %in% c(2,6) ) |
            (Block == "D" & Replicate %in% c(10) )|
            (Block == "E" & Replicate %in% c(5,10) )))
```

Reproduce figure 2E and Figure 3

```r
# calculate bootstrap sd of mean
#dd_clean$Time_bin <- ceiling(dd_clean$Time)


dd_clean$Time_bin <- as.numeric(cut2(dd_clean$Time, g=25))
dd_clean$split_var <- paste0(dd_clean$Species, "_", dd_clean$Time_bin)

species_time <- split(dd_clean, dd_clean$split_var)
mean_boot <- function(x, d) {
  return(mean(x[d], na.rm=T))
}

botstrapped_sds <- lapply(1:length(species_time), function(x)
  sd(boot(species_time[[x]]$density, mean_boot, R=1000)$t))

combi <- lapply(1:length(species_time), function(x)
  unique(paste0(species_time[[x]]$Species, "_", species_time[[x]]$Time_bin)))

# merge sd into df
sd_df <- data.frame(split_var=as.character(unlist(combi)),
                    boot_sd=unlist(botstrapped_sds), stringsAsFactors=F)


# merge back to original data
dd_clean <- dd_clean[order(dd_clean$split_var),]
dd_clean$boot_sd <- rep(unlist(botstrapped_sds), lapply(1:length(species_time), function(x) nrow(species


# crunch data into mean
mean_densities <- dd_clean %>% group_by(Species, Time_bin) %>% dplyr::summarize(mean_density = mean(dens
mean_densities$split_var <- paste0(mean_densities$Species, "_", mean_densities$Time_bin)
```

```r
# join with sd data
mean_densities <- merge(mean_densities, sd_df, by=c("split_var"))

ggplot() +
geom_line(data=subset(mean_densities, Time_bin > 4), aes(y=mean_density+1, x=Time_bin, group=Species, c
scale_colour_manual(values = c("green","red","blue")) +
geom_ribbon(data=subset(mean_densities, Time_bin > 4),
            aes(x=Time_bin, ymin=mean_density-boot_sd, ymax=mean_density+boot_sd, group=Species, fill=Sp
geom_line(data=subset(dd_clean, Replicate %in% 3:4 & Block == "A"),
          aes(y=density+1, x=Time_bin, group=interaction(Species, Replicate, Block), colour=Species))+
  geom_point(data=subset(dd_clean, Replicate %in% 3:4 & Block == "A"),
             aes(y=density+1, x=Time_bin, group=interaction(Species, Replicate, Block), colour=Species,
  scale_fill_manual(values = c("green","red","blue")) +
  theme(legend.position="bottom") +
  facet_grid(Species~., scales="free") +
  guides(shape=F) +
  scale_y_log10() +
  xlab("days")+
  ylab("Density (cells per ml)") +
  scale_x_discrete(labels=seq(0,100))
```

```
## Warning: Removed 2 rows containing missing values (geom_path).

## Warning: Removed 2 rows containing missing values (geom_path).

## Warning: Removed 2 rows containing missing values (geom_path).

## Warning: Removed 2 rows containing missing values (geom_point).

## Warning: Removed 2 rows containing missing values (geom_point).

## Warning: Removed 2 rows containing missing values (geom_point).
```



"Population dynamics in our replicate CESs under constant light and temperature. Shown are the mean ±bootstrap standard deviation over ecosystems with at least weekly measurements (n = 24), as well as data for two replicates (circles and triangles, respectively). Green (A, right axis): C. reinhardtii; red (B): E. coli; and blue (C): T. thermophila."

The figures shows a common trend across replicates, but with considerable fluctuations around this trend.

Reproduce analyses of variability in space and time

Variability across replicates (i.e., in space)

The aim here is to characterise across replicate variability in (species log transformed) densities. I.e., is it the case, at a particular point in time, that replicates with high Tetrahymena abundance

Calculate covariance among species abundances across replicates. E.g,. correlation of abundance of algae in replicate 1 at time 1.

Should be done at "weekly time points", and separately for experiment 1 and 2 (The first experiment consisted of what is called group F. The second experiment of groups A-E, with A measured twice per week, B once per week, etc.).

```
dd_clean <- mutate(dd_clean, Experiment=ifelse(Block=="F", 1, 2))
dd_clean <- mutate(dd_clean, Day=ceiling(Time))
dd_clean <- na.omit(dd_clean)
```

An example of an eigen analysis of one time slice, across replicates:

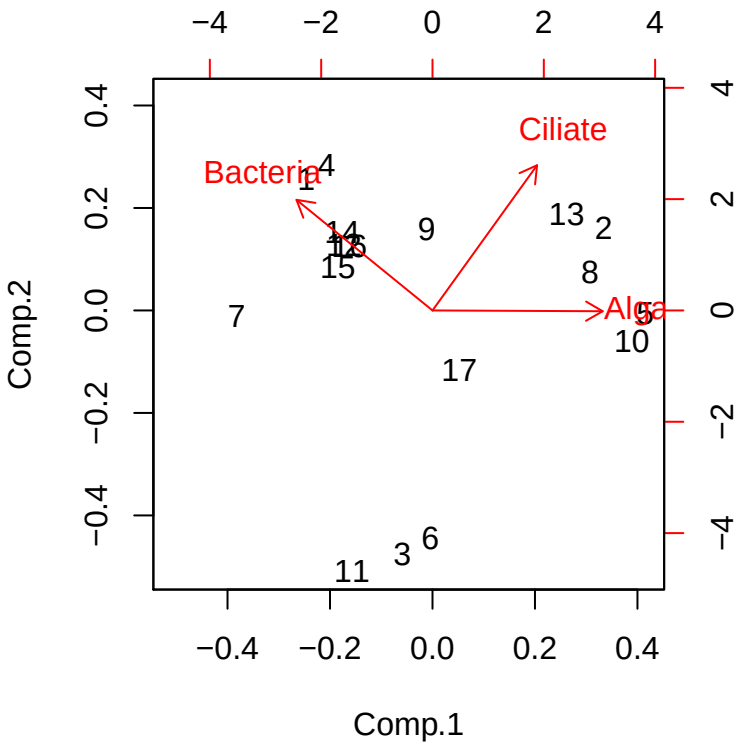
```
xxx <- filter(dd_clean, Experiment==1, Block=="F", Time_bin==2) %>%
  select(Experiment, Block, Time_bin, Replicate, Time, Species, density) %>%
  group_by(Experiment, Block, Time_bin) %>%
  spread(key=Species, value=density)
xxx[,c("Algae", "Bacteria", "Ciliate")] <- log(xxx[,c("Algae", "Bacteria", "Ciliate")])
xxx <- xxx[!apply(sapply(xxx[,c("Algae", "Bacteria", "Ciliate")], is.infinite), 1, function(x) any(x)),]
xxx <- na.omit(xxx)
#pairs(xxx[,c("Algae", "Bacteria", "Ciliate")])
eig <- eigen(cor(xxx[,c("Algae", "Bacteria", "Ciliate")]))
unlist(eig)
```

```
##      values1      values2      values3     vectors1     vectors2
##  1.732399145  0.988572350  0.279028505  0.704314856 -0.562824362
##     vectors3     vectors4     vectors5     vectors6     vectors7
##  0.432630699 -0.004330141  0.606017906  0.795439217 -0.709874520
##     vectors8     vectors9
## -0.562113009  0.424390305
```

```
eig$values
```

```
## [1] 1.7323991 0.9885724 0.2790285
```

```
pca1 <- princomp(xxx[,c("Algae", "Bacteria", "Ciliate")], cor=T)
biplot(pca1)
```




```
summary(pca1)
```

```
## Importance of components:
##                           Comp.1    Comp.2    Comp.3
## Standard deviation     1.3162063 0.9942698 0.5282315
## Proportion of Variance 0.5774664 0.3295241 0.0930095
## Cumulative Proportion  0.5774664 0.9069905 1.0000000
```

```
str(pca1)
```

```
## List of 7
##  $ sdev    : Named num [1:3] 1.316 0.994 0.528
##   ..- attr(*, "names")= chr [1:3] "Comp.1" "Comp.2" "Comp.3"
##  $ loadings: loadings [1:3, 1:3] 0.70431 -0.56282 0.43263 -0.00433 0.60602 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:3] "Algae" "Bacteria" "Ciliate"
##   .. ..$ : chr [1:3] "Comp.1" "Comp.2" "Comp.3"
##  $ center  : Named num [1:3] 6.42 11.08 6.43
##   ..- attr(*, "names")= chr [1:3] "Algae" "Bacteria" "Ciliate"
##  $ scale   : Named num [1:3] 0.9 1.24 1.31
##   ..- attr(*, "names")= chr [1:3] "Algae" "Bacteria" "Ciliate"
##  $ n.obs   : int 17
##  $ scores  : num [1:17, 1:3] -1.33 1.81 -0.32 -1.12 2.25 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:3] "Comp.1" "Comp.2" "Comp.3"
##  $ call    : language princomp(x = xxx[, c("Algae", "Bacteria", "Ciliate")], cor = T)
##  - attr(*, "class")= chr "princomp"
```

```
pca1$loadings
```

```
##
## Loadings:
##          Comp.1 Comp.2 Comp.3
## Algae     0.704         0.710
## Bacteria -0.563  0.606  0.562
## Ciliate   0.433  0.795 -0.424
##
##                Comp.1 Comp.2 Comp.3
## SS loadings     1.000  1.000  1.000
## Proportion Var  0.333  0.333  0.333
## Cumulative Var  0.333  0.667  1.000
```

Get eigen values and vectors for all time slices for both experiments:

```
f1 <- function(xxx) {

  num_data_required <- 6

  if(nrow(xxx)>num_data_required) {
    #print(xxx)
    xxx[,c("Algae", "Bacteria", "Ciliate")] <- log(xxx[,c("Algae", "Bacteria", "Ciliate")])
    xxx1 <- xxx[!apply(sapply(xxx[,c("Algae", "Bacteria", "Ciliate")], is.infinite), 1, function(x) any
    xxx1 <- na.omit(xxx1)
```

```
    if(nrow(xxx1)>num_data_required)
      res <- unlist(eigen(cor(xxx1[,c("Algae", "Bacteria", "Ciliate")])))
    if(nrow(xxx1)<=num_data_required)
      res <- rep(NA, 12)
  }

  if(nrow(xxx)<=num_data_required)
    res <- rep(NA, 12)
  res <- data.frame(val_L_null=res[1], val_M_null=res[2], val_S_null=res[3],
                    vect_L_Algae=res[4], vect_L_Bacteria=res[5], vect_L_Ciliate=res[6],
                    vect_M_Algae=res[7], vect_M_Bacteria=res[8], vect_M_Ciliate=res[9],
                    vect_S_Algae=res[10], vect_S_Bacteria=res[11], vect_S_Ciliate=res[12])

}


ccc <- select(dd_clean, Experiment, Block, Day, Replicate, Time, Species, density) %>%
 # filter(Day>0) %>%
  spread(key=Species, value=density) %>%
  na.omit(.) %>%
  group_by(Experiment, Block, Day) %>%
  do(f1(.))

## make the signs of vectors consistent relative to Algae
str(ccc)
```

```
## Classes 'grouped_df', 'tbl_df', 'tbl' and 'data.frame':  129 obs. of  15 variables:
##  $ Experiment     : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ Block          : chr  "F" "F" "F" "F" ...
##  $ Day            : num  0 1 2 3 4 5 6 7 8 9 ...
##  $ val_L_null     : num  NA NA 1.41 2.06 1.8 ...
##  $ val_M_null     : num  NA NA 1.285 0.7 0.958 ...
##  $ val_S_null     : num  NA NA 0.31 0.238 0.246 ...
##  $ vect_L_Algae   : num  NA NA -0.363 -0.643 -0.693 ...
##  $ vect_L_Bacteria: num  NA NA -0.492 -0.5 -0.684 ...
##  $ vect_L_Ciliate : num  NA NA 0.791 -0.58 0.23 ...
##  $ vect_M_Algae   : num  NA NA 0.748 0.121 0.122 ...
##  $ vect_M_Bacteria: num  NA NA -0.66 -0.814 0.203 ...
##  $ vect_M_Ciliate : num  NA NA -0.0674 0.5674 0.9716 ...
##  $ vect_S_Algae   : num  NA NA -0.555 0.756 0.711 ...
##  $ vect_S_Bacteria: num  NA NA -0.567 -0.295 -0.701 ...
##  $ vect_S_Ciliate : num  NA NA -0.608 -0.5847 0.0569 ...
##  - attr(*, "vars")=List of 3
##   ..$ : symbol Experiment
##   ..$ : symbol Block
##   ..$ : symbol Day
##  - attr(*, "drop")= logi TRUE
##  - attr(*, "indices")=List of 129
##   ..$ : int 0
##   ..$ : int 1
##   ..$ : int 2
##   ..$ : int 3
##   ..$ : int 4
```

```
##   ..$ : int 5
##   ..$ : int 6
##   ..$ : int 7
##   ..$ : int 8
##   ..$ : int 9
##   ..$ : int 10
##   ..$ : int 11
##   ..$ : int 12
##   ..$ : int 13
##   ..$ : int 14
##   ..$ : int 15
##   ..$ : int 16
##   ..$ : int 17
##   ..$ : int 18
##   ..$ : int 19
##   ..$ : int 20
##   ..$ : int 21
##   ..$ : int 22
##   ..$ : int 23
##   ..$ : int 24
##   ..$ : int 25
##   ..$ : int 26
##   ..$ : int 27
##   ..$ : int 28
##   ..$ : int 29
##   ..$ : int 30
##   ..$ : int 31
##   ..$ : int 32
##   ..$ : int 33
##   ..$ : int 34
##   ..$ : int 35
##   ..$ : int 36
##   ..$ : int 37
##   ..$ : int 38
##   ..$ : int 39
##   ..$ : int 40
##   ..$ : int 41
##   ..$ : int 42
##   ..$ : int 43
##   ..$ : int 44
##   ..$ : int 45
##   ..$ : int 46
##   ..$ : int 47
##   ..$ : int 48
##   ..$ : int 49
##   ..$ : int 50
##   ..$ : int 51
##   ..$ : int 52
##   ..$ : int 53
##   ..$ : int 54
##   ..$ : int 55
##   ..$ : int 56
##   ..$ : int 57
##   ..$ : int 58
```

```
##   ..$ : int 59
##   ..$ : int 60
##   ..$ : int 61
##   ..$ : int 62
##   ..$ : int 63
##   ..$ : int 64
##   ..$ : int 65
##   ..$ : int 66
##   ..$ : int 67
##   ..$ : int 68
##   ..$ : int 69
##   ..$ : int 70
##   ..$ : int 71
##   ..$ : int 72
##   ..$ : int 73
##   ..$ : int 74
##   ..$ : int 75
##   ..$ : int 76
##   ..$ : int 77
##   ..$ : int 78
##   ..$ : int 79
##   ..$ : int 80
##   ..$ : int 81
##   ..$ : int 82
##   ..$ : int 83
##   ..$ : int 84
##   ..$ : int 85
##   ..$ : int 86
##   ..$ : int 87
##   ..$ : int 88
##   ..$ : int 89
##   ..$ : int 90
##   ..$ : int 91
##   ..$ : int 92
##   ..$ : int 93
##   ..$ : int 94
##   ..$ : int 95
##   ..$ : int 96
##   ..$ : int 97
##   ..$ : int 98
##   .. [list output truncated]
##  - attr(*, "group_sizes")= int  1 1 1 1 1 1 1 1 1 1 ...
##  - attr(*, "biggest_group_size")= int 1
##  - attr(*, "labels")='data.frame':   129 obs. of  3 variables:
##   ..$ Experiment: num  1 1 1 1 1 1 1 1 1 1 ...
##   ..$ Block     : chr  "F" "F" "F" "F" ...
##   ..$ Day       : num  0 1 2 3 4 5 6 7 8 9 ...
##   ..- attr(*, "vars")=List of 3
##   .. ..$ : symbol Experiment
##   .. ..$ : symbol Block
##   .. ..$ : symbol Day
```

```
ccc$L_mult <- ifelse(ccc$vect_L_Algae>0, 1, -1)
ccc[,c("vect_L_Algae", "vect_L_Bacteria", "vect_L_Ciliate")] <- ccc[,c("vect_L_Algae", "vect_L_Bacteria
```

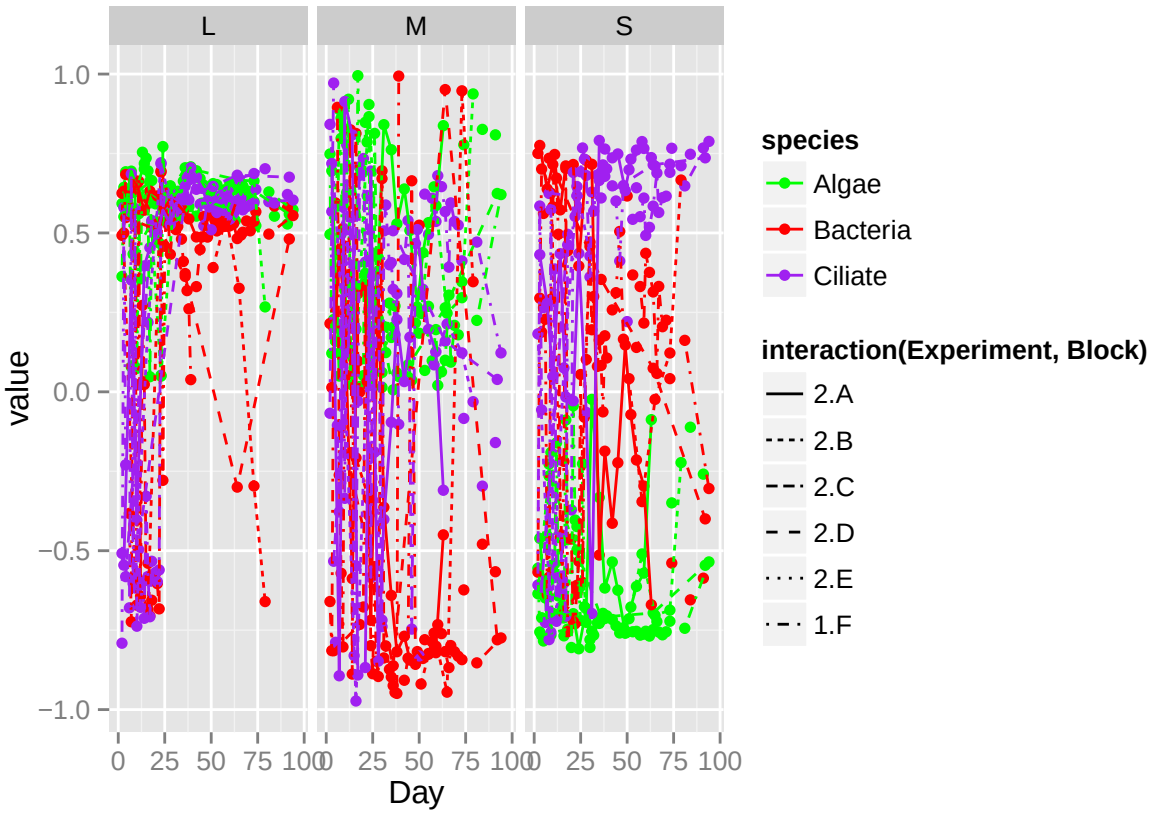
```
ccc$M_mult <- ifelse(ccc$vect_M_Algae>0, 1, -1)
ccc[,c("vect_M_Algae", "vect_M_Bacteria", "vect_M_Ciliate")] <- ccc[,c("vect_M_Algae", "vect_M_Bacteria
ccc$S_mult <- ifelse(ccc$vect_S_Algae>0, -1, 1)
ccc[,c("vect_S_Algae", "vect_S_Bacteria", "vect_S_Ciliate")] <- ccc[,c("vect_S_Algae", "vect_S_Bacteria
```

```
ccc <- gather(ccc, key=variable, value=value, 4:15)
ccc <- separate(ccc, variable, sep="_", into=c("quantity", "component", "species"))
```

Reproduce original article figure 4a-c:

```
ggplot(filter(ccc, quantity=="vect"),
       aes(x=Day, y=value, col=species, linetype=interaction(Experiment, Block))) +
  geom_point() + geom_path() +
  facet_wrap(~component, nrow=1) +
  scale_color_manual(values=c("green","red","purple"))
```

```
## Warning: Removed 114 rows containing missing values (geom_point).

## Warning: Removed 114 rows containing missing values (geom_point).

## Warning: Removed 114 rows containing missing values (geom_point).

## Warning: Removed 54 rows containing missing values (geom_path).

## Warning: Removed 54 rows containing missing values (geom_path).

## Warning: Removed 54 rows containing missing values (geom_path).
```

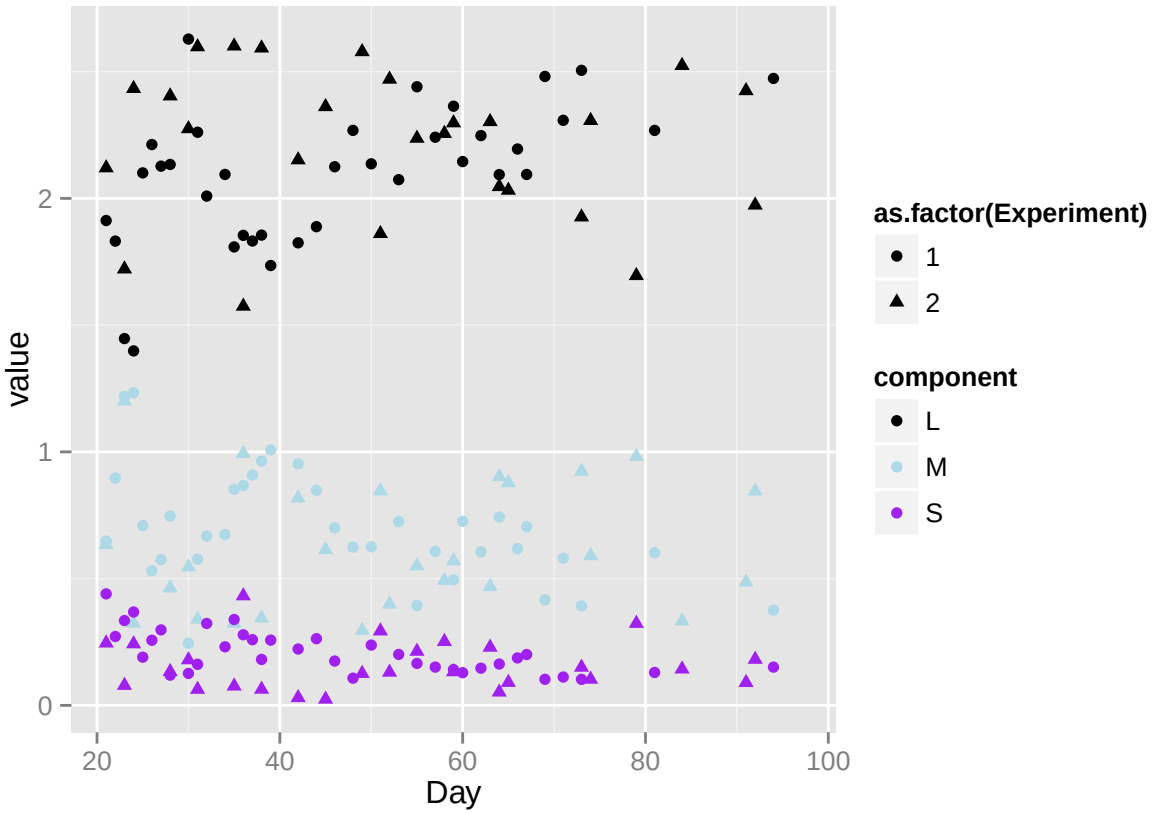


Qualitative patterns reproduced...

Reproduce original article figure 4b:

```
ggplot(filter(ccc, quantity=="val", Day>20),
       aes(x=Day, y=value, col=component, shape=as.factor(Experiment))) +
  geom_point(size=2) +
  scale_color_manual(values=c("black","lightblue","purple"))
```

```
## Warning: Removed 99 rows containing missing values (geom_point).
```



Be sure to understand the "surprising" correspondence of the ecomodes and the eigen values of the interaction matrix.

Analysis of growth rate fluctuations (Figure S3)

Analysis of Hurst exponent

Taylor law analysis

Effect of difference in gas seal quality among replicates

Conclusions

"establish two simple statistical results describing the nature of random fluctuations around the average dynamics"

1: "the variations of the three species were corre- lated. Well-defined ecomodes that describe these correlations emerged and stabilized after an initial period of about 3 weeks. The existence of these ecomodes reflects the fact that fluctua- tions of the three species' densities around the replicate-average dynamics are coupled through ecological interactions that are common to all replicates."

2: "despite the large complexity of biological phenomena observed in individual ecosystems, it was, remarkably, possible to describe the resulting fluctuations in population dynamics by simple quantitative laws. Specifically, local population dynamics displayed power-law behavior close to a geometric random walk around the average

dynamics. Underlying these random walks is a single dominant ecomode, along which density fluctuations do not revert to the mean."

"recently observed extreme repeatability of temporal dynamics and spatial patterns in similar experiments involving long-term single-species dynamics (Frentz et al., 2010) shows that the number of interacting components, the nature of their interactions, or the details of starting or external conditions may play a crucial role (cf. Jiang et al., 2011)."

Ideas

Need to think about what the analyses above assume about linearity of interactions / relationship among species. A non-linear approach (e.g., multispecies GAM) might give interesting insight.