Phase 1: Univariate Analysis of BGC

This analysis can help us distinguish hypotheses about underlying mechanisms in the lake (ala Hsieh et al. 2005). It also serves as a foundation for subsequent causal and quasi-mechanistic modeling in two ways. (1) It provides guidance on parameters for the analysis, like degree of time averaging. In this case, we are entertaining 2 month and 3 month temporal averaging of the core BGC variables. (2) It validates the grounding assumption of CCM analysis, which is that there are low-dimensional attractor dynamics that (at least partially) explain changes in the time series variables.

Data Setup

We first set packages.

```
library(rEDM)
```

```
## Warning: package 'rEDM' was built under R version 3.6.2
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.6.2

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1

## Warning: package 'ggplot2' was built under R version 3.6.2

## Warning: package 'tibble' was built under R version 3.6.2

## Warning: package 'tidyr' was built under R version 3.6.2

## Warning: package 'readr' was built under R version 3.6.2

## Warning: package 'purrr' was built under R version 3.6.2

## Warning: package 'dplyr' was built under R version 3.6.2

## Warning: package 'forcats' was built under R version 3.6.2

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
sessionInfo()
```

```
## R version 3.6.0 (2019-04-26)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS  10.15.7
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib
```

```
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] forcats_0.5.1   stringr_1.4.0   dplyr_1.0.7     purrr_0.3.4
##  [5] readr_2.0.2     tidyr_1.1.4     tibble_3.1.6    ggplot2_3.3.5
##  [9] tidyverse_1.3.1 rEDM_1.10.0
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.1.1 xfun_0.28        haven_2.4.3      colorspace_2.0-2
##  [5] vctrs_0.3.8      generics_0.1.1   htmltools_0.5.2  yaml_2.2.1
##  [9] utf8_1.2.2       rlang_0.4.12     pillar_1.6.4     withr_2.4.2
## [13] glue_1.5.0       DBI_1.1.1        dbplyr_2.1.1     modelr_0.1.8
## [17] readxl_1.3.1     lifecycle_1.0.1  munsell_0.5.0    gtable_0.3.0
## [21] cellranger_1.1.0 rvest_1.0.2      evaluate_0.14    knitr_1.36
## [25] tzdb_0.2.0       fastmap_1.1.0    fansi_0.5.0      broom_0.7.10
## [29] Rcpp_1.0.7       backports_1.3.0  scales_1.1.1     jsonlite_1.7.2
## [33] fs_1.5.0         hms_1.1.1        digest_0.6.28    stringi_1.7.5
## [37] grid_3.6.0       cli_3.1.0        tools_3.6.0      magrittr_2.0.1
## [41] crayon_1.4.2     pkgconfig_2.0.3  ellipsis_0.3.2   xml2_1.3.2
## [45] reprex_2.0.1     lubridate_1.8.0  rstudioapi_0.13  assertthat_0.2.1
## [49] rmarkdown_2.11   httr_1.4.2       R6_2.5.1         compiler_3.6.0
```

Then load in data created in “Phase 0” from “_0_data_wrangling.Rmd”

```r
load("./DATA/PROCESSED/_0 2-month BGC.Rdata")
# load("./DATA/PROCESSED/_0 3-month BGC.Rdata")

df_BGC <- df_2mo_BGC_LTP[102:323,]
# gap fill with linear interpolant
df_BGC <- mutate_at(df_BGC,-1, ~ zoo::na.approx(., maxgap = 4))
T_annual <- 6

# df_BGC <- df_3mo_BGC_LTP[68:215,]
# T_season <- 4
L_save_params <- c("columns","target","E","Tp","knn","tau","theta")
```

EDM Analysis Setup

We will replicate the analysis across several variables, using a few tests and statistics to contextualize and interpret the results.

Null Hypothesis 1: Signal is due to random chance. Null Hypothesis 2: Signal is due to serial autocorrelation in the time series. Null Hypothesis 3: Signal is due to seasonal cycling.

```
num_surr <- 500
```

Univariate Analysis

```
do_univariate_1_var <- function(df,target_col){

  E_list <- 1:15
  theta_list <- c(0,10^seq(-2,1,by=.075))

  lib <- paste(1,NROW(df_BGC))
  pred <- paste(1,NROW(df_BGC))

  ## Simplex
  stats_simplex <- map_df(E_list,function(E_i){
    out_simplex_i <- Simplex(dataFrame=df,
                             target=target_col,
                             columns=target_col,
                             lib=lib,pred=pred,
                             E=E_i,
                             parameterList=TRUE)

    stats_i <- compute_stats(out_simplex_i$predictions$Predictions,
                             out_simplex_i$predictions$Observations)

    stats_i <- bind_cols(out_simplex_i$parameters[L_save_params],
              stats_i
              )

    return(stats_i)

  })

  stats_simplex <- suppressMessages(type_convert(stats_simplex))

  E_star <- as.integer(stats_simplex$E[which.max(stats_simplex$rho)])

  ## S-map
  stats_smap <- map_df(theta_list,function(theta_i){

    out_smap_i <- SMap(dataFrame=df,
                       target=target_col,
                       columns=target_col,
                       lib=lib,pred=pred,
                       E=E_star,
                       theta=theta_i,
                       parameterList = TRUE)

    stats_i <- compute_stats(out_smap_i$predictions$Predictions,out_smap_i$predictions$Observations)

    stats_i <- bind_cols(
      out_smap_i$parameters[L_save_params],
              stats_i
              )

    return(stats_i)

  })

  stats_smap <- suppressMessages(type_convert(stats_smap))

  return(list(simplex=stats_simplex,smap=stats_smap))

}
```

```
summarise_univars <- function(edm_stats){

  out <- data.frame(
    target = edm_stats$simplex$target[1],
    rho_simplex = max(edm_stats$simplex$rho),
    mae_simplex = min(edm_stats$simplex$mae),
    rmse_simplex = min(edm_stats$simplex$rmse),
    rho_smap_0 = edm_stats$smap$rho[1],
    mae_smap_0 = edm_stats$smap$mae[1],
    rmse_smap_0 = edm_stats$smap$rmse[1],
    rho_smap = max(edm_stats$smap$rho),
    mae_smap = min(edm_stats$smap$mae),
    rmse_smap = min(edm_stats$smap$rmse)
  )

  return(out)
}
```

Null Hypothesis 1

Null Hypothesis 1: Signal is due to random chance.

```
do_shuffle_surrogates <- function(df,target_col,n_surr=500){

  df_surr <- SurrogateData(df %>% pull(!!target_col),method="random_shuffle",num_surr=n_surr)

  map_df(1:n_surr,function(i_surr){

    df_i <- cbind(df[,1],data.frame(surr=df_surr[,i_surr]))
    out <- do_univariate_1_var(df_i,target_col="surr")

    out_summary <- summarise_univars(out)

    return(out_summary)

  })

}
```

Null Hypothesis 2

Null Hypothesis 2: Signal is due to serial autocorrelation in the time series.

```
do_ebi_surrogates <- function(df,target_col,n_surr=500){

  df_surr <- SurrogateData(df %>% pull(!!target_col),method="ebisuzaki",num_surr=n_surr)

  map_df(1:n_surr,function(i_surr){

    df_i <- cbind(df[,1],data.frame(surr=df_surr[,i_surr]))
    out <- do_univariate_1_var(df_i,target_col="surr")

    out_summary <- summarise_univars(out)

    return(out_summary)

  })
}
```

Null Hypothesis 3

This we will test with a seasonal surrogate method. Essentially, we ask if the EDM statistics of the real time series are significantly improved from a time series formed from the same seasonal cycle with randomized residuals– essentially a time series that is an equivalent noisy measure of the seasonal cycle as the true time series.

In another situation, we could consider instead comparison to non-parametric predictions based solely on the phase of season. However, since we are looking at 2-3mo temporal averaging, this variable only takes on 4-6 possible values. On the other hand, this is a perhaps overconservative, since the translation of the seasonal sinusoid into a noisy nonlinear oscillation itself could be considered a sign of deterministic dynamics.

```
do_seasonal_surrogates <- function(df,target_col,n_surr=500,T_period=4){

  df_surr <- SurrogateData(df %>% pull(!!target_col),method="seasonal",num_surr=n_surr,T_period = T_peri

  map_df(1:n_surr,function(i_surr){

    df_i <- cbind(df[,1],data.frame(surr=df_surr[,i_surr]))
    out <- do_univariate_1_var(df_i,target_col="surr")

    out_summary <- summarise_univars(out)

    return(out_summary)

  })

}
```

EDM Analysis

```
out_all_var_univar <- map_dfr(names(df_BGC)[-1],function(var_name){
  out <- do_univariate_1_var(df_BGC,var_name)
  return(summarise_univars(out))
  })
```

I will add standard "rho-vs-E" and "rho-vs-theta" plots here for at least Secchi_Ave.

```
out_all_var_univar %>% select(target,contains("rho"))
```

```
##                target rho_simplex rho_smap_0  rho_smap
## 1          Secchi_Ave   0.5611225  0.5990398 0.6040749
## 2          Chla_sechi   0.5784655  0.5613425 0.5666599
## 3 Chla_deep_euphotic   0.4388003  0.4816683 0.4822009
## 4           NO3_sechi   0.8061744  0.7478747 0.7883192
## 5  NO3_deep_euphotic   0.5770626  0.6005863 0.6036487
```

In general, we see the biogeochemistry variables have predictable short-term dynamics. The lowest forecast skill is seen for the deep chlorophyll concentration, "Chla_deep_euphotic", which shows simplex forecast skill of about rho = 0.44, which is still highly significant for > 200 data points under the parametric derivations of Pearson's correlation for Gaussian random variables. These data, of course, do not follow Gaussian distributions, so it is better to assess significance using null surrogate methods.

The evidence of nonlinear dynamics is not particularly strong at least at this view. The improvement in forecast skill from S-map (theta=0), i.e. a global linear model, to the optimal nonlinearly tuned S-map (theta>0) is most pronounced for NO3 in the "secchi zone".

Surrogates

We rerun the univariate analysis on surrogate date for each variable under each null hypothesis (i.e. using shuffle surrogates, seasonal surrogates, and phase-randomized fourier surrogates).

```
do_all_surrogates <- function(target_variable,file_surr){
  if(!file.exists(file_surr)){
    out_ebi <- do_ebi_surrogates(df_BGC,target_variable,n_surr = num_surr) %>%
      mutate(target=target_variable)
    out_seasonal <- do_seasonal_surrogates(df_BGC,target_variable,n_surr = num_surr,T_period=T_annual) %
      mutate(target=target_variable)
    out_shuffle <- do_shuffle_surrogates(df_BGC,target_variable,n_surr = num_surr) %>%
      mutate(target=target_variable)

    save(out_ebi,out_seasonal,out_shuffle,file=file_surr)
  }else{
    load(file_surr)
  }
}
```

```
file_Sechi_surr <- "./RESULTS/_1_sechi_ave_univar_surr_2mo.Rdata"
do_all_surrogates("Secchi_Ave",file_Sechi_surr)
```

```
file_Chla_sechi_surr <- "./RESULTS/_1_Chla_sechi_univar_surr_2mo.Rdata"
do_all_surrogates("Chla_sechi",file_Chla_sechi_surr)
```

```
file_Chla_deep_euphotic <- "./RESULTS/_1_Chla_deep_euphotic_univar_surr_2mo.Rdata"
do_all_surrogates("Chla_deep_euphotic",file_Chla_deep_euphotic)
```

```
file_NO3_sechi_surr <- "./RESULTS/_1_NO3_sechi_univar_surr_2mo.Rdata"
do_all_surrogates("NO3_sechi",file_NO3_sechi_surr)
```

```
file_NO3_deep_euphotic <- "./RESULTS/_1_NO3_deep_euphotic_univar_surr_2mo.Rdata"
do_all_surrogates("NO3_deep_euphotic",file_NO3_deep_euphotic)
```

```
# sum(out_ebi$delta_rho_smap < max(out_univar$smap$rho) - out_univar$smap$rho[1])
# sum(out_ebi$rho_simplex < max(out_univar$simplex$rho))
#
# sum(out_seasonal$delta_rho_smap < max(out_univar$smap$rho) - out_univar$smap$rho[1])
# sum(out_seasonal$rho_simplex < max(out_univar$simplex$rho))
#
# sum(out_shuffle$delta_rho_smap < max(out_univar$smap$rho) - out_univar$smap$rho[1])
# sum(out_shuffle$rho_simplex < max(out_univar$simplex$rho))
```

Visualization

We wish to look at EDM benchmarks across the BGC variables, comparing each to the null distributions. This requires re-organizing the outputs somewhat.

```
collect_null_results <- function(outputs,labels){

  outputs_collected <- map2_dfr(outputs,labels,function(x,y) x %>% mutate(method=y))
  outputs_collected <- outputs_collected %>%
    mutate(method=as_factor(method))
  return(outputs_collected)

}
```

```
source("./FUNCTIONS/_1_funs_plotting.R")
```

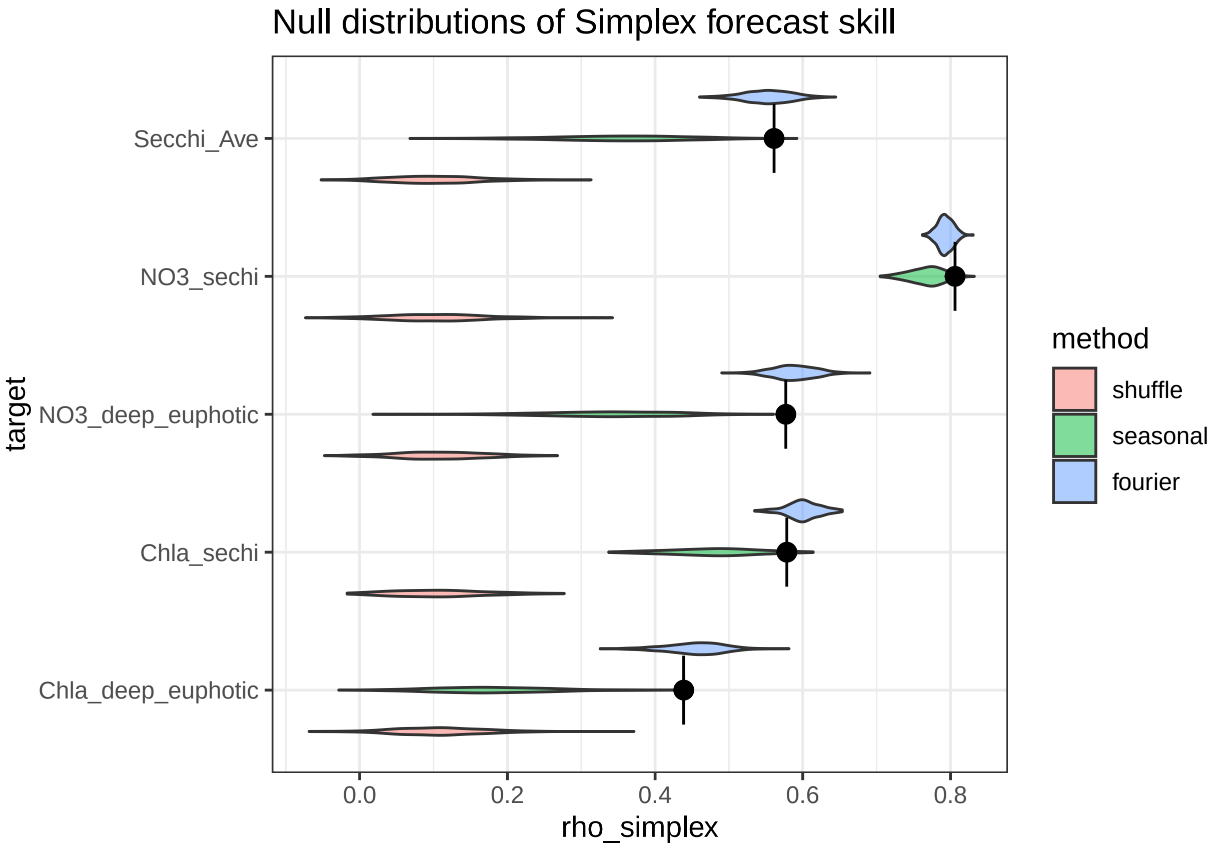
Collect across the list of output files:

```
file_list <- list(file_Sechi_surr,
                  file_Chla_sechi_surr,
                  file_Chla_deep_euphotic,
                  file_NO3_sechi_surr,
                  file_NO3_deep_euphotic)


out_all_null_all_var <- map_dfr(file_list,function(fpath){

  env_i <- new.env()
  load(fpath,env_i)

  out_i <- collect_null_results(outputs=list(env_i$out_shuffle,env_i$out_seasonal,env_i$out_ebi),
                       labels=list("shuffle","seasonal","fourier"))

  return(out_i)

})
```

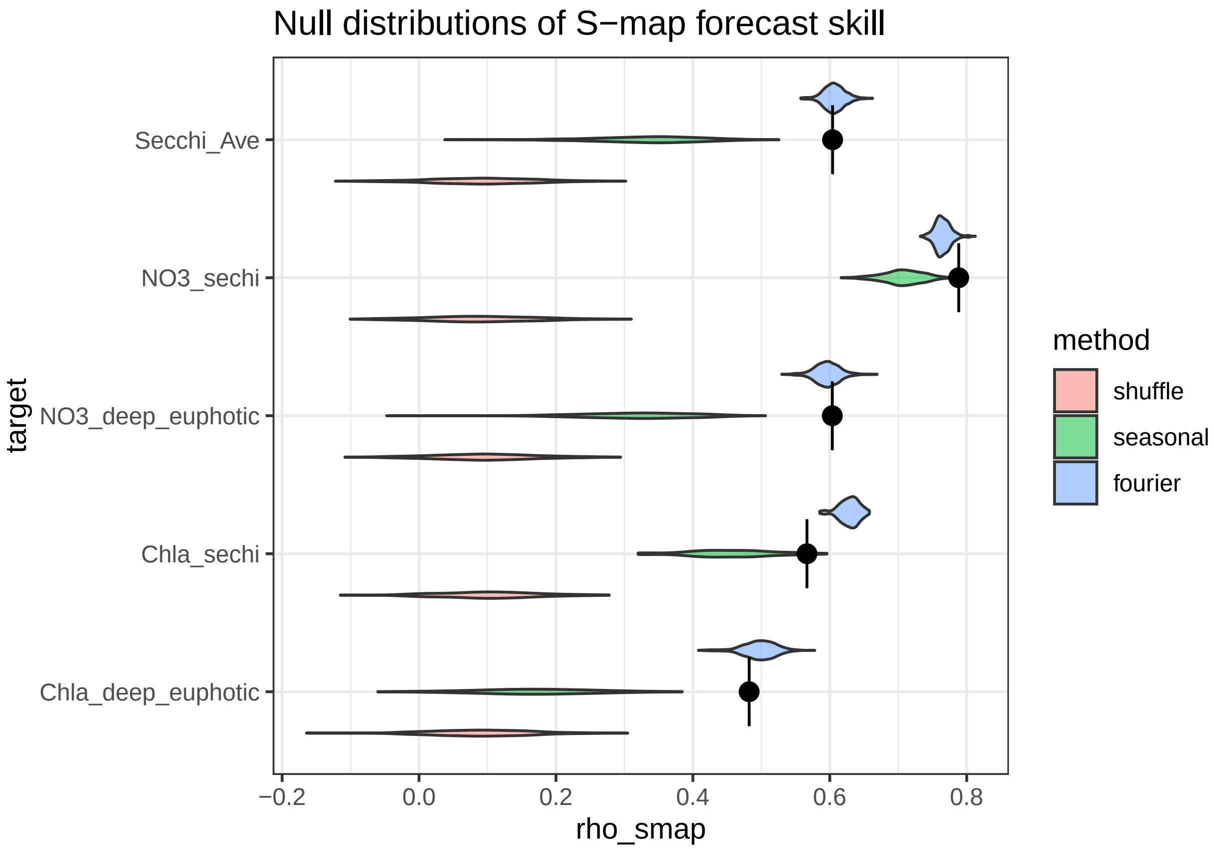
```
# g_jit <- plot_null_jitters(out_all_null_all_var,"rho_simplex")
# g_vio <- plot_null_violins(out_all_null_all_var,"rho_simplex")
#
# g_vio + theme_bw()
#
```

```
# g_group_box <- out_all_null_all_var %>% ggplot(aes(x=target,y=rho_simplex,fill=method)) + geom_boxplo
# g_group_violin <- out_all_null_all_var %>% ggplot(aes(x=target,y=rho_simplex,fill=method)) + geom_vio
```

```
out_all_null_all_var %>% ggplot(aes(x=target,y=rho_simplex)) +
  geom_violin(alpha=0.5,aes(fill=method)) +
  geom_point(data=out_all_var_univar,size=3) +
  geom_spoke(data=out_all_var_univar,aes(angle = 0, radius = 0.25)) +
geom_spoke(data=out_all_var_univar,aes(angle = pi, radius = 0.25)) +
  coord_flip() +
  theme_bw() +
  labs(title="Null distributions of Simplex forecast skill")
```



```
out_all_null_all_var %>% ggplot(aes(x=target,y=rho_smap)) +
  geom_violin(alpha=0.5,aes(fill=method)) +
  geom_point(data=out_all_var_univar,size=3) +
  geom_spoke(data=out_all_var_univar,aes(angle = 0, radius = 0.25)) +
geom_spoke(data=out_all_var_univar,aes(angle = pi, radius = 0.25)) +
  coord_flip() +
  theme_bw() +
  labs(title="Null distributions of S-map forecast skill")
```



In all cases, prediction skills are well outside of what would be expected for random data with the same distribution of values (i.e. the "shuffle" surrogates). The seasonal patterns in all the time series do create more prediction, but in general the EDM results appear to lie outside the expected prediction skill of the "seasonal" null as well. By and large, however, the phase-randomized Fourier surrogates (i.e. the Ebisuzaki method) produce forecast skill distributions that include the empirical result in most cases.

The difference between the seasonal null distributions and Fourier null distributions is potentially interesting. The Fourier null will generally capture prediction due to seasonal cycling but also longer time-scale fluctuations, including secular trends. It is in many ways a very conservative null model, since there is potentially a lot of interesting limnology in the Fourier spectra of these data.

Interpretation

Endogenous versus exogenous drivers of change

Audience: Us

For data sets in limnological settings, how well can we use these kinds of approach to untangle causality?

Audience: Colleagues doing parametric analyses

Parametric models can't really deal with long-term causality.

Audience: Management Agencies

Forecasting and decision-making.

Exogenous contexts/climate patterns: - El nino - Drought

Legacy effects on the way the system behaves based on relationships with climate.

Questions: - Do climatic drivers have simple or complex effects?

Audience: Activity Report

Next-Steps