Maladaptive plastic responses of flowering time to geothermal heating

Code for analyses in the paper (revised)

Alicia Valdés

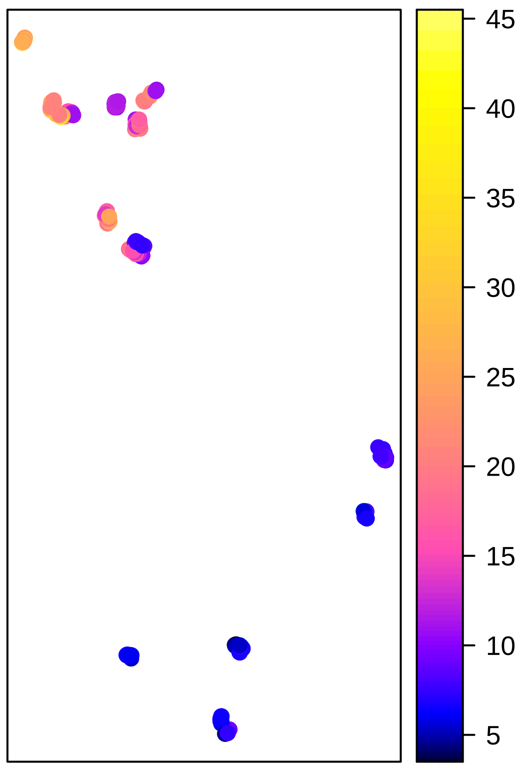
Contents

Read data

The location of these files would need to be changed.

```
data_plants<-read_csv("data/clean/data_plants_coords.csv")
logger_data<-read_csv("data/clean/logger_data.csv")
logger_data_pairs<-read_csv("data/clean/logger_data_pairs.csv")
```

```
#Defining coordinates and coordinate system####
coordinates(data_plants) <- c("x", "y")
project1<-"+proj=utm +zone=27 +ellps=WGS84 +datum=WGS84 +units=m +no_defs"
proj4string(data_plants) = CRS(project1) #assign CRS with projected coordinates

#A plot of all the plants, colored by temperature
spplot(data_plants, "temp", do.log=T, colorkey = TRUE)
```



Correlation between instant measures of soil temperature and mean soil temperature during the period April 1st – June 5th recorded by loggers

For each logger_nr, get mean temperature during April-June and compare with temp_term (which was measured with a thermometer at 10 cm depth on May 2017):

```
with(logger_data%>%
       mutate(month = month(datetime)) %>%
       filter(month==4|month==5|month==6)%>%
```

```r
      filter(above_below=="B")%>%
      mutate(date=date(datetime))%>%
      filter(!is.na(date))%>% # remove records with no info on date
      filter(datetime<"2018-06-06")%>% # keep only data until June 5
      group_by(logger_nr) %>%
      summarize(mean_logger=mean(temp),temp_term=mean(temp_term)),
    cor.test(mean_logger,temp_term))
```

```
##
##  Pearson's product-moment correlation
##
## data:  mean_logger and temp_term
## t = 21.901, df = 139, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8370572 0.9129479
## sample estimates:
##       cor
## 0.8805259
```

Is soil temperature more weakly correlated with air temperature in warmer soils?

For each date and logger pair, calculate mean, max and min of air and soil temperature (from, respectively, the above and belowground logger). Then, calculate the correlation coefficient for air and soil temperatures over the period May or April-May-June. Finally, regress these correlation coefficients on mean soil temperature (from the belowground logger) for the same period (April-May-June).

May

```r
data_corr<-(logger_data_pairs%>%
     mutate(month = month(datetime),date=date(datetime))%>%
              # new variables "month" and "date"
     filter(month==5)%>% # keep data from may
     group_by(date,pair,above_below)%>%
     summarise(mean=mean(temp,na.rm=T),max=max(temp),min=min(temp))%>%
   #calculate mean, max and min of air and soil temperature
     pivot_wider(names_from="above_below",values_from=c("mean","max","min"))%>%
     group_by(pair)%>%
     summarise(corr_airsoil_mean=cor(mean_A,mean_B,use="pairwise.complete.obs"),
               corr_airsoil_max=cor(max_A,max_B,use="pairwise.complete.obs"),
               corr_airsoil_min=cor(min_A,min_B,use="pairwise.complete.obs"))%>%
   # Calculate correlations air-soil temperatures
     pivot_longer(cols=corr_airsoil_mean:corr_airsoil_min,
                  names_to="measure",values_to="corr")%>%
  left_join(logger_data_pairs%>%
                 mutate(month = month(datetime))%>%
                 filter(month==5)%>%
                 filter(above_below=="B")%>%
```

```
                 group_by(pair)%>%
              summarise(meansoiltemp=mean(temp))))
  # calculate mean soil temperature for may
```

```
model_mean<-lm(corr~meansoiltemp,
               data = subset(data_corr,measure=="corr_airsoil_mean"))
model_max<-lm(corr~meansoiltemp,
               data = subset(data_corr,measure=="corr_airsoil_max"))
model_min<-lm(corr~meansoiltemp,
               data = subset(data_corr,measure=="corr_airsoil_min"))
```

Predictions of correlations for minimum and maximum temperatures:

```
ggpredict(model_mean,terms="meansoiltemp[minmax]")
```

```
## # Predicted values of corr
##
## meansoiltemp | Predicted |       95% CI
## ---------------------------------------
##         6.14 |      0.83 | [0.80, 0.86]
##        30.55 |      0.61 | [0.55, 0.68]
```

```
ggpredict(model_max,terms="meansoiltemp[minmax]")
```

```
## # Predicted values of corr
##
## meansoiltemp | Predicted |       95% CI
## ---------------------------------------
##         6.14 |      0.74 | [0.69, 0.79]
##        30.55 |      0.37 | [0.26, 0.48]
```

```
ggpredict(model_min,terms="meansoiltemp[minmax]")
```

```
## # Predicted values of corr
##
## meansoiltemp | Predicted |       95% CI
## ---------------------------------------
##         6.14 |      0.75 | [0.73, 0.78]
##        30.55 |      0.59 | [0.53, 0.65]
```

Appendix S2 (part 1)

Linear models testing the effect of soil temperature on correlations between soil and air temperature (part of table in Appendix S3):

```
(logger_data_pairs%>%
     mutate(month = month(datetime),date=date(datetime))%>%
         # new variables "month" and "date"
     filter(month==5)%>% # keep data from may
     group_by(date,pair,above_below)%>%
```

```
    summarise(mean=mean(temp,na.rm=T),max=max(temp),min=min(temp))%>%
  #calculate mean, max and min of air and soil temperature
    pivot_wider(names_from="above_below",values_from=c("mean","max","min"))%>%
    group_by(pair)%>%
    summarise(corr_airsoil_mean=cor(mean_A,mean_B,use="pairwise.complete.obs"),
              corr_airsoil_max=cor(max_A,max_B,use="pairwise.complete.obs"),
              corr_airsoil_min=cor(min_A,min_B,use="pairwise.complete.obs"))%>%
  # Calculate correlations air-soil temperatures
    pivot_longer(cols=corr_airsoil_mean:corr_airsoil_min,
                 names_to="measure",values_to="corr")%>%
  left_join(logger_data_pairs%>%
               mutate(month = month(datetime))%>%
               filter(month==5)%>%
               filter(above_below=="B")%>%
               group_by(pair)%>%
             summarise(meansoiltemp=mean(temp))))%>%
  # calculate mean soil temperature for may
  group_by(measure)%>%
  do(fitcorr = tidy(lm(corr~meansoiltemp, data = .))) %>%
  unnest(fitcorr)%>%
  kable(digits=5)
```

measure	term	estimate	std.error	statistic	p.value
corr_airsoil_max	(Intercept)	0.83528	0.04012	20.81870	0e+00
corr_airsoil_max	meansoiltemp	-0.01521	0.00287	-5.29894	1e-05
corr_airsoil_mean	(Intercept)	0.88734	0.02275	39.00495	0e+00
corr_airsoil_mean	meansoiltemp	-0.00897	0.00163	-5.50910	0e+00
corr_airsoil_min	(Intercept)	0.79583	0.02147	37.06454	0e+00
corr_airsoil_min	meansoiltemp	-0.00684	0.00154	-4.44851	7e-05

April-May_june

Appendix S2 (part 2)

Linear models testing the effect of soil temperature on correlations between soil and air temperature (part of table in Appendix S3):

```
(logger_data_pairs%>%
    mutate(month = month(datetime),date=date(datetime))%>%
         # new variables "month" and "date"
    filter(month==4|month==5|month==6)%>% # keep data from april-june
    group_by(date,pair,above_below)%>%
    summarise(mean=mean(temp,na.rm=T),max=max(temp),min=min(temp))%>%
  #calculate mean, max and min of air and soil temperature
    pivot_wider(names_from="above_below",values_from=c("mean","max","min"))%>%
    group_by(pair)%>%
    summarise(corr_airsoil_mean=cor(mean_A,mean_B,use="pairwise.complete.obs"),
              corr_airsoil_max=cor(max_A,max_B,use="pairwise.complete.obs"),
              corr_airsoil_min=cor(min_A,min_B,use="pairwise.complete.obs"))%>%
  # Calculate correlations air-soil temperatures
    pivot_longer(cols=corr_airsoil_mean:corr_airsoil_min,
                 names_to="measure",values_to="corr")%>%
```

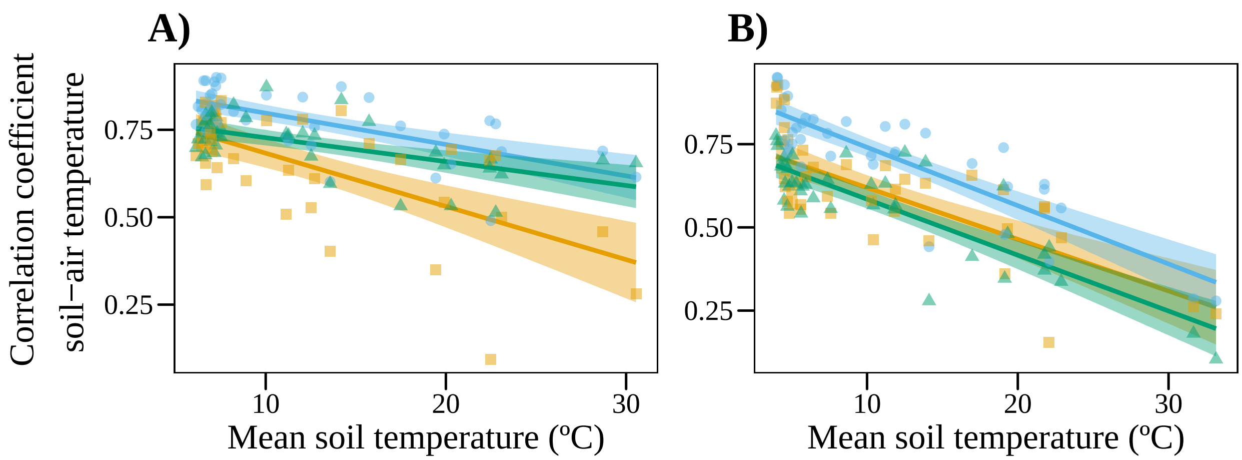
```
left_join(logger_data_pairs%>%
               mutate(month = month(datetime))%>%
               filter(month==4|month==5|month==6)%>%
               filter(above_below=="B")%>%
               group_by(pair)%>%
            summarise(meansoiltemp=mean(temp))))%>%
# calculate mean soil temperature for april-june
group_by(measure)%>%
do(fitcorr = tidy(lm(corr~meansoiltemp, data = .))) %>%
unnest(fitcorr)%>%
kable(digits=5)
```

measure	term	estimate	std.error	statistic	p.value
corr_airsoil_max	(Intercept)	0.77449	0.03184	24.32387	0
corr_airsoil_max	meansoiltemp	-0.01552	0.00236	-6.57111	0
corr_airsoil_mean	(Intercept)	0.91574	0.02391	38.29900	0
corr_airsoil_mean	meansoiltemp	-0.01752	0.00177	-9.88060	0
corr_airsoil_min	(Intercept)	0.75246	0.02361	31.87482	0
corr_airsoil_min	meansoiltemp	-0.01679	0.00175	-9.59056	0

Appendix S1: Correlations soil-air temperature vs soil temperature

```
cbPalette <- c("#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2",
               "#D55E00", "#CC79A7")
AppS1A<-(logger_data_pairs%>%
          mutate(month = month(datetime),date=date(datetime))%>%
         # new variables "month" and "date"
          filter(month==5)%>% # keep data from may
          group_by(date,pair,above_below)%>%
          summarise(mean=mean(temp,na.rm=T),max=max(temp),min=min(temp))%>%
          #calculate mean, max and min of air and soil temperature
          pivot_wider(names_from="above_below",
                      values_from=c("mean","max","min"))%>%
          group_by(pair)%>%
          summarise(corr_airsoil_mean=cor(mean_A,mean_B,
                                         use="pairwise.complete.obs"),
                    corr_airsoil_max=cor(max_A,max_B,
                                        use="pairwise.complete.obs"),
                    corr_airsoil_min=cor(min_A,min_B,
                                        use="pairwise.complete.obs"))%>%
          # Calculate correlations air-soil temperatures
          pivot_longer(cols=corr_airsoil_mean:corr_airsoil_min,
                       names_to="measure",values_to="corr")%>%
          left_join(logger_data_pairs%>%
                      mutate(month = month(datetime))%>%
                      filter(month==5)%>%
                      filter(above_below=="B")%>%
                      group_by(pair)%>%
                      summarise(meansoiltemp=mean(temp))))%>%
  # calculate mean soil temperature for may
  ggplot(.,aes(x=meansoiltemp,y=corr,color=measure,fill=measure,shape=measure))+
```

```
  geom_smooth(method="lm",size=1)+geom_point(size=2,alpha=0.5)+
  xlab("Mean soil temperature (ºC)")+
  ylab(NULL)+
  my_theme()+scale_fill_manual(values=cbPalette)+
  scale_colour_manual(values=cbPalette)+
  scale_shape_manual(values=c(15,16,17)) +
  geom_text_repel(data=. %>% filter(corr<0),aes(label=pair))+
  ggtitle("A)")
AppS1B<-(logger_data_pairs%>%
          mutate(month = month(datetime),date=date(datetime))%>%
          # new variables "month" and "date"
          filter(month==4|month==5|month==6)%>% # keep data from april-june
          group_by(date,pair,above_below)%>%
          summarise(mean=mean(temp,na.rm=T),max=max(temp),min=min(temp))%>%
          #calculate mean, max and min of air and soil temperature
          pivot_wider(names_from="above_below",
                      values_from=c("mean","max","min"))%>%
          group_by(pair)%>%
          summarise(corr_airsoil_mean=cor(mean_A,mean_B,
                                          use="pairwise.complete.obs"),
                    corr_airsoil_max=cor(max_A,max_B,
                                         use="pairwise.complete.obs"),
                    corr_airsoil_min=cor(min_A,min_B,
                                         use="pairwise.complete.obs"))%>%
          # Calculate correlations air-soil temperatures
          pivot_longer(cols=corr_airsoil_mean:corr_airsoil_min,
                       names_to="measure",values_to="corr")%>%
          left_join(logger_data_pairs%>%
                      mutate(month = month(datetime))%>%
                      filter(month==4|month==5|month==6)%>%
                      filter(above_below=="B")%>%
                      group_by(pair)%>%
                      summarise(meansoiltemp=mean(temp))))%>%
  # calculate mean soil temperature for april-june
  ggplot(.,aes(x=meansoiltemp,y=corr,color=measure,fill=measure,shape=measure))+
  geom_smooth(method="lm",size=1)+geom_point(size=2,alpha=0.5)+
  xlab("Mean soil temperature (ºC)")+
  ylab(NULL)+
  my_theme()+scale_fill_manual(values=cbPalette)+
  scale_colour_manual(values=cbPalette)+
  scale_shape_manual(values=c(15,16,17)) +
  geom_text_repel(data=. %>% filter(corr<0),aes(label=pair))+
  ggtitle("B)")
AppS1<-grid.arrange(AppS1A,AppS1B,ncol=2,
                    left=textGrob("Correlation coefficient\nsoil-air temperature",just="center",
                      hjust=0.42,
                      gp=gpar(fontsize=16,fontfamily="serif"),
                      rot = 90))
```




```
ggsave(filename="output/figures/AppS1.tiff",plot=AppS1,
       width=24,height=10,units="cm",dpi=300)
```

Hypothesis 1: Effect of temperature on FFD

Models including quadratic effects of ffd.

```
data_plants$year_fct<-as.factor(data_plants$year)
FFD_1<-lm(ffd~(temp+I(temp^2))*year_fct,data_plants)
summ(FFD_1,vif=T)
```

Observations	349
Dependent variable	ffd
Type	OLS linear regression

F(5,343)	28.500
R^2	0.294
Adj. R^2	0.283

	Est.	S.E.	t val.	p	VIF
(Intercept)	183.736	2.052	89.536	0.000	NA
temp	-0.779	0.254	-3.065	0.002	17.405
I(temp^2)	0.012	0.007	1.832	0.068	16.329
year_fct2018	18.016	4.167	4.323	0.000	16.426
temp:year_fct2018	-1.383	0.595	-2.322	0.021	76.002
I(temp^2):year_fct2018	0.021	0.017	1.219	0.224	35.733

Standard errors: OLS

Quadratic terms of ffd not significant. Refit models without quadratic terms of ffd.

```r
FFD_1<-lm(ffd~temp*year_fct,data_plants)
summ(FFD_1,vif=T)
```

Observations	349
Dependent variable	ffd
Type	OLS linear regression

F(3,345)	44.212
R^2	0.278
Adj. R^2	0.271

	Est.	S.E.	t val.	p	VIF
(Intercept)	180.626	1.162	155.505	0.000	NA
temp	-0.332	0.072	-4.620	0.000	1.369
year_fct2018	14.478	2.082	6.955	0.000	4.033
temp:year_fct2018	-0.738	0.140	-5.250	0.000	4.164

Standard errors: OLS

Predictions of ffd for minimum and maximum temperatures:

```r
range(subset(data_plants,year==2017)$temp)
```

```
## [1]  4.1 45.5
```

```r
range(subset(data_plants,year==2018)$temp)
```

```
## [1]  3.5 34.0
```

```r
ggpredict(FFD_1,terms=c("temp[4.1,45.5]","year_fct[2017]"))
```

```
## # Predicted values of ffd
##
##  temp | Predicted |           95% CI
## ------------------------------------
##  4.10 |    179.26 | [177.47, 181.06]
## 45.50 |    165.52 | [160.97, 170.08]
```

```r
# 179.26-165.52=14 days earlier on warmer soils
ggpredict(FFD_1,terms=c("temp[3.5,34.0]","year_fct[2018]"))
```

```
## # Predicted values of ffd
##
##  temp | Predicted |           95% CI
## ------------------------------------
##  3.50 |    191.36 | [188.66, 194.06]
## 34.00 |    158.74 | [153.35, 164.13]
```

```
# 191.36-158.74=45 days earlier on warmer soils
```

Hypothesis 2: Effect of temperature on fitness

```
fitness_1<-glm.nb(n_seed_round~(temp+I(temp^2))*year_fct+log(nfl),
                  data_plants) # Quadratic term significant, keep
summ(fitness_1,vif=T)
```

Observations	349
Dependent variable	n_seed_round
Type	Generalized linear model
Family	Negative Binomial(2.0752)
Link	log

χ^2()	0.747	0.097	4472.025	4502.865
Pseudo-R² (Cragg-Uhler)	0.747	0.097	4472.025	4502.865
Pseudo-R² (McFadden)	0.747	0.097	4472.025	4502.865
AIC	0.747	0.097	4472.025	4502.865
BIC	0.747	0.097	4472.025	4502.865

	Est.	S.E.	z val.	p	VIF
(Intercept)	4.227	0.168	25.104	0.000	NA
temp	-0.051	0.021	-2.498	0.012	17.929
I(temp^2)	0.001	0.001	1.064	0.287	16.480
year_fct2018	-1.158	0.334	-3.464	0.001	16.560
log(nfl)	0.983	0.036	27.345	0.000	1.102
temp:year_fct2018	0.088	0.048	1.829	0.067	76.148
I(temp^2):year_fct2018	-0.003	0.001	-2.142	0.032	35.522

Standard errors: MLE

Predictions of fitness for minimum and maximum temperatures:

```
range(subset(data_plants,year==2017)$temp)
```

```
## [1]  4.1 45.5
```

```
range(subset(data_plants,year==2018)$temp)
```

```
## [1]  3.5 34.0
```

```
ggpredict(fitness_1,terms=c("temp[4.1,45.5]","year_fct[2017]"))
```

```
## # Predicted counts of n_seed_round
##
```

```
##  temp | Predicted |           95% CI
## ------------------------------------
##  4.10 |    810.91 | [659.59, 996.93]
## 45.50 |    311.40 | [142.72, 679.46]
##
## Adjusted for:
## * nfl = 15.17
```

```
ggpredict(fitness_1,terms=c("temp[3.5,34.0]","year_fct[2018]"))
```

```
## # Predicted counts of n_seed_round
##
##  temp | Predicted |           95% CI
## ------------------------------------
##  3.50 |    343.18 | [246.14, 478.48]
## 34.00 |     63.33 | [ 29.27, 137.00]
##
## Adjusted for:
## * nfl = 15.17
```

Figure 2: Effects of temperature on ffd and fitness

Model prediction ffd : based on model FFD_1 (without quadratic term of ffd)

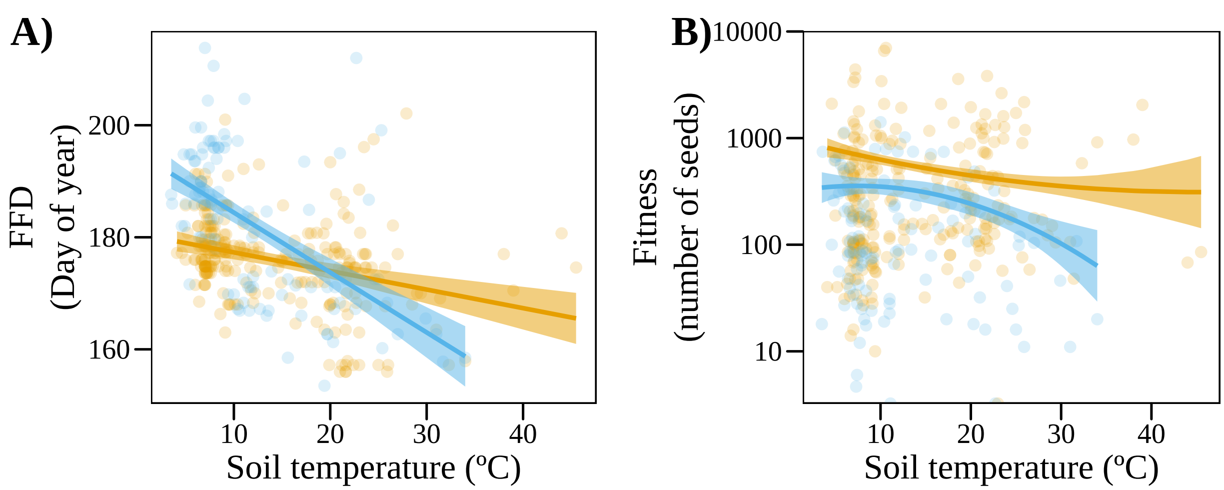
```
predict_FFD<-ggpredict(FFD_1,terms=c("temp [all]","year_fct"))
```

Model prediction fitness : based on model fitness_1

```
predict_fitness<-ggpredict(fitness_1,terms=c("temp [all]","year_fct"))
```

```
fig2<-
  grid.arrange(
    # ffd
    ggplot(data.frame(data_plants),aes(x=temp,y=ffd))+
  xlab("Soil temperature (ºC)")+ylab("FFD\n(Day of year)")+my_theme()+
  geom_point(aes(color=year_fct),size=2,alpha=0.2)+
  geom_ribbon(data=subset(predict_FFD,group==2017&
                        x>=min(subset(data_plants,year==2017)$temp)&
                        x<=max(subset(data_plants,year==2017)$temp)),
            aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high,fill=group),
            alpha=0.5)+
  geom_line(data=subset(predict_FFD,group==2017&
                        x>=min(subset(data_plants,year==2017)$temp)&
                        x<=max(subset(data_plants,year==2017)$temp)),
          aes(x=x,y=predicted,color=group),size=1)+
  geom_ribbon(data=subset(predict_FFD,group==2018&
                          x>=min(subset(data_plants,year==2018)$temp)&
                          x<=max(subset(data_plants,year==2018)$temp)),
            aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high,fill=group),
            alpha=0.5)+
  geom_line(data=subset(predict_FFD,group==2018&
```

```
                          x>=min(subset(data_plants,year==2018)$temp)&
                          x<=max(subset(data_plants,year==2018)$temp)),
            aes(x=x,y=predicted,color=group),size=1)+
  ggtitle("A)")+theme(plot.title=element_text(hjust=-0.35,vjust=-3))+
  theme(plot.margin = unit(c(-0.6,0.3,0,0.3), "cm"))+
  scale_fill_manual(values=cbPalette)+scale_color_manual(values=cbPalette),
    # fitness
    ggplot(data.frame(data_plants),aes(x=temp,y=nseed))+
      xlab("Soil temperature (ºC)")+ylab("Fitness\n(number of seeds)")+
      my_theme()+
      geom_point(aes(color=year_fct),size=2,alpha=0.2)+
      geom_ribbon(data=subset(predict_fitness,group==2017&
                          x>=min(subset(data_plants,year==2017)$temp)&
                          x<=max(subset(data_plants,year==2017)$temp)),
                  aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high,fill=group),
                  alpha=0.5)+
      geom_line(data=subset(predict_fitness,group==2017&
                          x>=min(subset(data_plants,year==2017)$temp)&
                          x<=max(subset(data_plants,year==2017)$temp)),
                aes(x=x,y=predicted,color=group),size=1)+
    geom_ribbon(data=subset(predict_fitness,group==2018&
                          x>=min(subset(data_plants,year==2018)$temp)&
                          x<=max(subset(data_plants,year==2018)$temp)),
                  aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high,fill=group),
                  alpha=0.5)+
      geom_line(data=subset(predict_fitness,group==2018&
                          x>=min(subset(data_plants,year==2018)$temp)&
                          x<=max(subset(data_plants,year==2018)$temp)),
                aes(x=x,y=predicted,color=group),size=1)+
      ggtitle("B)")+theme(plot.title=element_text(hjust=-0.35,vjust=-3))+
      theme(plot.margin = unit(c(-0.6,0.3,0,0.3), "cm"))+
      scale_fill_manual(values=cbPalette)+scale_color_manual(values=cbPalette)+
      scale_y_continuous(trans='log10'),
    ncol=2)
```




```
ggsave(filename="output/figures/fig2.tiff",plot=fig2,
       width=22,height=8,units="cm",dpi=300)
```

Hypothesis 3: Effect of temperature on selection on FFD

Models including quadratic effects of temp.

```
selection_1<-lm(nseed_rel~ffd_std*(temp+I(temp^2))*year_fct+nfl_std,
                data_plants)
summ(selection_1,vif=T)
```

Observations	349
Dependent variable	nseed_rel
Type	OLS linear regression

F(12,336)	26.559
R^2	0.487
Adj. R^2	0.468

	Est.	S.E.	t val.	p	VIF
(Intercept)	1.612	0.277	5.812	0.000	NA
ffd_std	0.430	0.359	1.200	0.231	36.292
temp	-0.062	0.033	-1.845	0.066	18.854
I(temp^2)	0.001	0.001	1.040	0.299	17.109
year_fct2018	-0.227	0.716	-0.317	0.751	30.420
nfl_std	1.155	0.071	16.333	0.000	1.411
ffd_std:temp	-0.030	0.042	-0.718	0.473	174.924
ffd_std:I(temp^2)	0.001	0.001	0.581	0.562	76.286
ffd_std:year_fct2018	-0.851	0.738	-1.153	0.250	45.614
temp:year_fct2018	0.037	0.111	0.336	0.737	165.435
I(temp^2):year_fct2018	-0.001	0.004	-0.180	0.857	96.685
ffd_std:temp:year_fct2018	0.090	0.101	0.890	0.374	248.539
ffd_std:I(temp^2):year_fct2018	-0.002	0.003	-0.503	0.615	132.433

Standard errors: OLS

Quadratic terms of temp not significant. Refit models without quadratic terms of temp

```
selection_1<-lm(nseed_rel~ffd_std*temp*year_fct+nfl_std,
                data_plants)
summ(selection_1,vif=T)
```

Observations	349
Dependent variable	nseed_rel
Type	OLS linear regression

F(8,340)	39.925
R^2	0.484
Adj. R^2	0.472

	Est.	S.E.	t val.	p	VIF
(Intercept)	1.392	0.154	9.043	0.000	NA
ffd_std	0.285	0.189	1.510	0.132	10.148
temp	-0.029	0.010	-3.045	0.003	1.563
year_fct2018	-0.116	0.299	-0.388	0.699	5.357
nfl_std	1.146	0.070	16.436	0.000	1.382
ffd_std:temp	-0.008	0.009	-0.819	0.414	8.727
ffd_std:year_fct2018	-0.466	0.312	-1.496	0.136	8.189
temp:year_fct2018	0.016	0.022	0.721	0.472	6.547
ffd_std:temp:year_fct2018	0.035	0.018	1.909	0.057	8.373

Standard errors: OLS

BCa intervals

Used for assessing significance.

```
# ffd
slp <- function(selection_1) coef(selection_1)[2]
b <- car::Boot(selection_1,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
ffd_ci <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# temp
slp <- function(selection_1) coef(selection_1)[3]
b <- car::Boot(selection_1,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
temp_ci <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# year
slp <- function(selection_1) coef(selection_1)[4]
b <- car::Boot(selection_1,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
year_ci <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# nfl
slp <- function(selection_1) coef(selection_1)[5]
b <- car::Boot(selection_1,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
nfl_ci <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# ffd:temp
slp <- function(selection_1) coef(selection_1)[6]
b <- car::Boot(selection_1,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
ffd_temp_ci <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```r
# ffd:year
slp <- function(selection_1) coef(selection_1)[7]
b <- car::Boot(selection_1,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
ffd_year_ci <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```r
# temp:year
slp <- function(selection_1) coef(selection_1)[8]
b <- car::Boot(selection_1,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
temp_year_ci <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```r
# ffd:temp:year
slp <- function(selection_1) coef(selection_1)[9]
b <- car::Boot(selection_1,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
ffd_temp_year_ci <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```r
# Save confidence intervals as a table
BCIs_selection_1 <- cbind(
  rbind(ffd_ci[1,] ,temp_ci[1,], year_ci[1,], nfl_ci[1,], ffd_temp_ci[1,],
        ffd_year_ci[1,],temp_year_ci[1,],ffd_temp_year_ci[1,]),
  rbind(ffd_ci[2,] ,temp_ci[2,], year_ci[2,], nfl_ci[2,], ffd_temp_ci[2,],
        ffd_year_ci[2,],temp_year_ci[2,],ffd_temp_year_ci[2,])
)
colnames(BCIs_selection_1)<-c("lower","upper")
rownames(BCIs_selection_1) <- c("ffd","temp","year","nfl","ffd:temp",
                                "ffd:year","temp:year","ffd:temp:year")
save(BCIs_selection_1,file="output/BCIs_selection_1.RData")
```

```r
BCIs_selection_1
```

```
##                     lower        upper
## ffd           -0.02091869  0.656360362
## temp          -0.05206476 -0.012639518
## year          -0.80886531  0.420408148
## nfl            0.92081413  1.524992693
## ffd:temp      -0.02352751  0.006110947
## ffd:year      -1.05684449  0.101172982
## temp:year     -0.01556584  0.059183120
## ffd:temp:year  0.00606531  0.065951760
```

Figure 3: Effects of temperature on selection

```r
quantile(subset(data_plants,year==2017)$temp,probs=c(0.05,0.95))
```

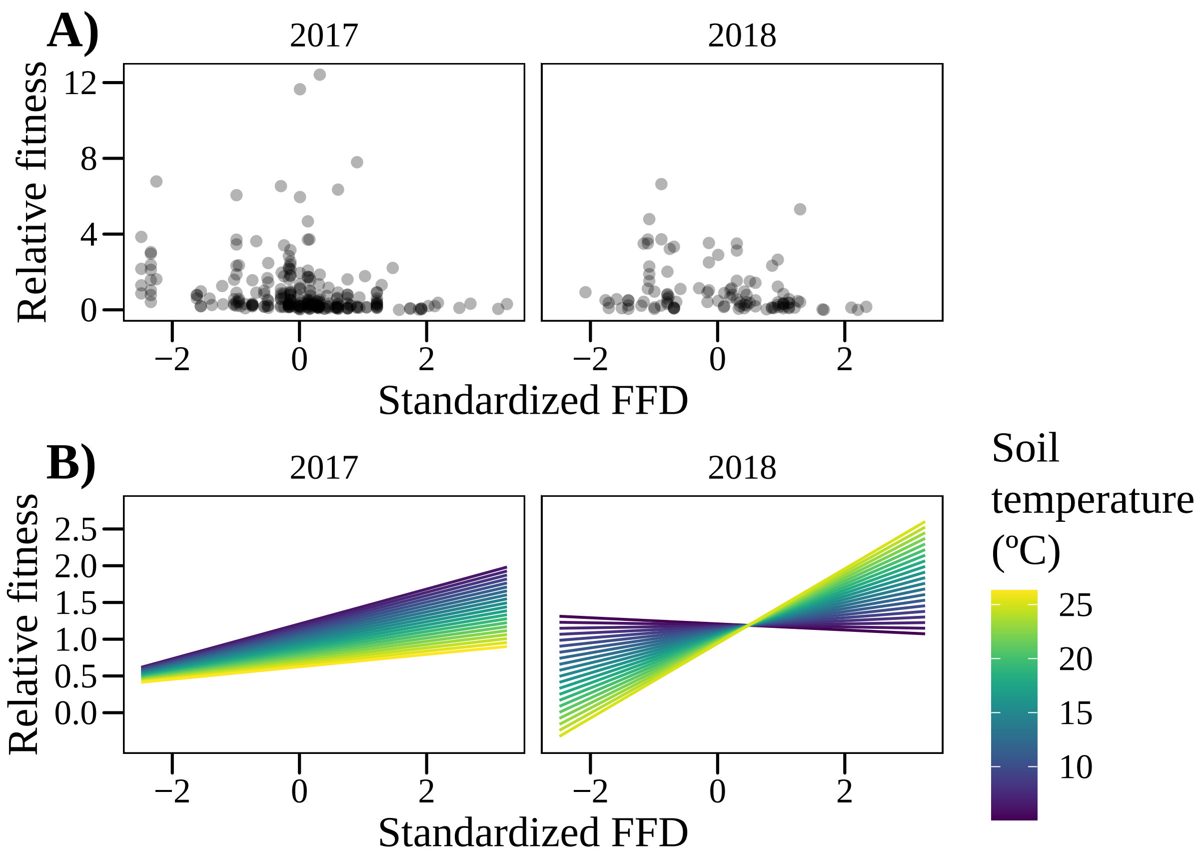
```
##    5%   95%
##  6.32 26.90
```

```
quantile(subset(data_plants,year==2018)$temp,probs=c(0.05,0.95))
```

```
##     5%    95%
##  5.060 25.825
```

```
pred_fitness_17<-ggpredict(selection_1,
                           terms = c("ffd_std[all]","temp[6.32:26.90]",
                                     "year_fct[2017]"))
pred_fitness_18<-ggpredict(selection_1,
                           terms = c("ffd_std[all]","temp[5.060:25.825]",
                                     "year_fct[2018]"))
```

```
fig3<-cowplot::plot_grid(ggplot(data.frame(data_plants),
                                aes(x=ffd_std,y=nseed_rel))+
                           facet_grid(~year,scales="fixed")+
                           geom_point(size=1.5,alpha=0.3)+
                           my_theme()+ggtitle("A)")+
                           xlab("Standardized FFD")+ylab("Relative fitness")+
                           theme(plot.title=element_text(hjust=-0.1,vjust=-5))+
                           theme(plot.margin = unit(c(-0.6,0.3,0,0.3), "cm")),
                         ggplot(rbind(data.frame(pred_fitness_17),
                                      data.frame(pred_fitness_18)),
                                aes(x,predicted,colour=group,fill=group))+
                           facet_grid(~facet,scales="free")+
                           geom_line(aes(color=as.numeric(as.character(group))),
                                     size=0.5)+
                           my_theme_legend()+
                           theme(legend.position="right")+ggtitle("B)")+
                           scale_color_viridis()+
                           labs(colour="Soil\ntemperature\n(ºC)")+
                           xlab("Standardized FFD")+ylab("Relative fitness")+
                           theme(plot.title=element_text(hjust=-0.1,vjust=-5))+
                           theme(plot.margin = unit(c(-0.6,0.3,0,0.3), "cm"))+
                           scale_y_continuous(limits=c(-0.4,2.8),
                                              breaks=c(0,0.5,1,1.5,2,2.5)),
                         ncol=1,align="v",axis="lr")
fig3
```




```r
ggsave(filename="output/figures/fig3.tiff",plot=fig3,
       width=18,height=16,units="cm",dpi=300)
```

Appendix S4

```r
quantile(subset(data_plants,year==2017)$temp)
```

```
##   0%  25%  50%  75% 100%
##  4.1  7.4 10.7 20.6 45.5
```

```r
mean(subset(data_plants,year==2017&temp<=7.4)$temp)
```

```
## [1] 6.714286
```

```r
# Mean cat 1 = 6.714286
mean(subset(data_plants,year==2017&temp>7.4&temp<=10.7)$temp)
```

```
## [1] 8.748333
```

```r
# Mean cat 2 = 8.748333
mean(subset(data_plants,year==2017&temp>10.7&temp<=20.6)$temp)
```

```
## [1] 16.2129
```

```r
# Mean cat 3 = 16.2129
mean(subset(data_plants,year==2017&temp>20.6)$temp)
```

```
## [1] 25.10667
```

```r
# Mean cat 4 = 25.10667

quantile(subset(data_plants,year==2018)$temp)
```

```
##     0%    25%    50%    75%   100%
##  3.500  6.875  9.300 17.075 34.000
```

```r
mean(subset(data_plants,year==2018&temp<=6.875)$temp)
```

```
## [1] 5.784615
```

```r
# Mean cat 1 = 5.784615
mean(subset(data_plants,year==2018&temp>6.875&temp<=9.300)$temp)
```

```
## [1] 7.95
```

```r
# Mean cat 2 = 7.95
mean(subset(data_plants,year==2018&temp>9.300&temp<=17.075)$temp)
```

```
## [1] 12.45
```

```r
# Mean cat 3 = 12.45
mean(subset(data_plants,year==2018&temp>17.075)$temp)
```

```
## [1] 23.28846
```

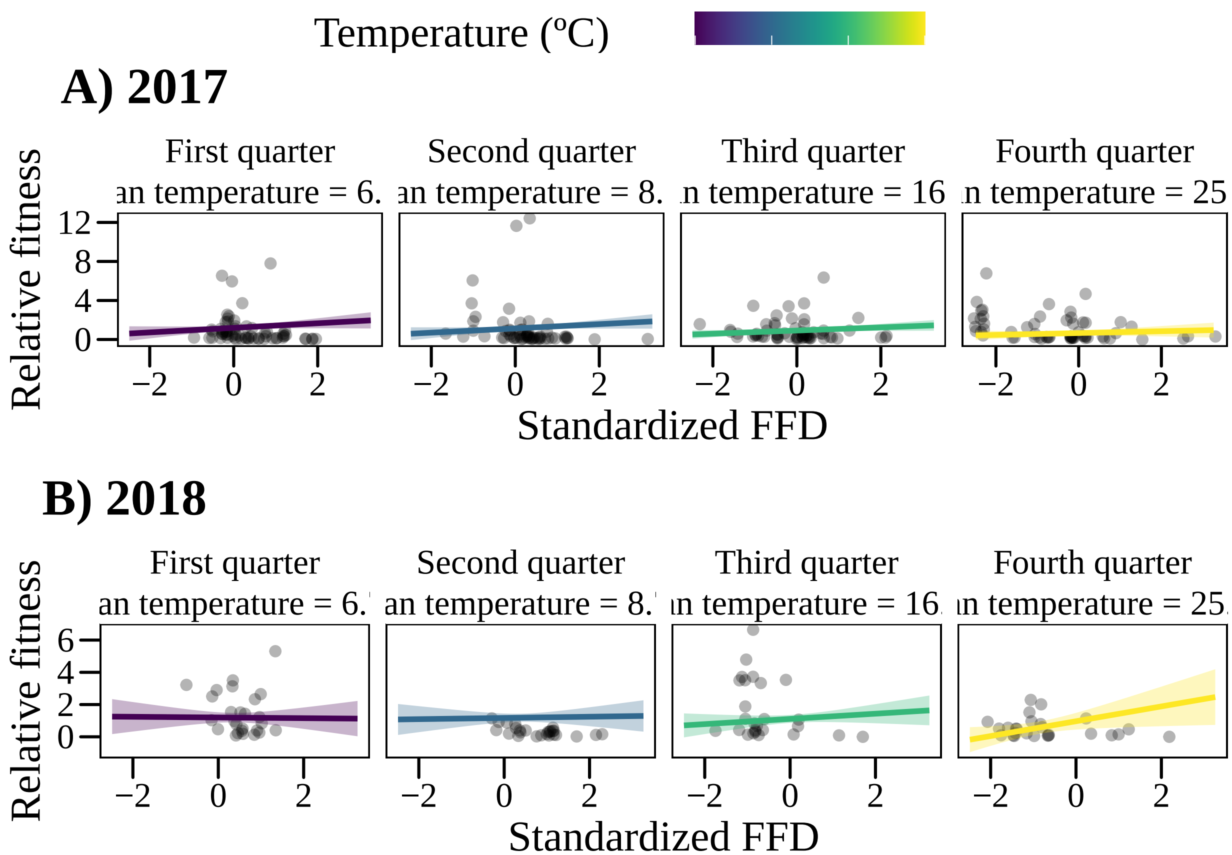
```r
# Mean cat 4 = 23.28846

pred_fitness_17_cats<-rbind(
  (data.frame(ggpredict(selection_1,
                         terms = c("ffd_std[all]","temp[6.714286]",
                                   "year_fct[2017]")))%>%
     mutate(temp_cat=1)%>%
     dplyr::rename(FFD_std=x, fitness=predicted,temp=group)),
  (data.frame(ggpredict(selection_1,
                       terms = c("ffd_std[all]","temp[8.748333]",
                                 "year_fct[2017]")))%>%
     mutate(temp_cat=2)%>%
```

```r
    dplyr::rename(FFD_std=x, fitness=predicted,temp=group)),
  (data.frame(ggpredict(selection_1,
                        terms = c("ffd_std[all]","temp[16.2129]",
                                  "year_fct[2017]")))%>%
    mutate(temp_cat=3)%>%
    dplyr::rename(FFD_std=x, fitness=predicted,temp=group)),
  (data.frame(ggpredict(selection_1,
                        terms = c("ffd_std[all]","temp[25.10667]",
                                  "year_fct[2017]")))%>%
    mutate(temp_cat=4)%>%
    dplyr::rename(FFD_std=x, fitness=predicted,temp=group)))

pred_fitness_18_cats<-rbind(
  (data.frame(ggpredict(selection_1,
                          terms = c("ffd_std[all]","temp[5.784615]",
                                  "year_fct[2018]")))%>%
    mutate(temp_cat=1)%>%
    dplyr::rename(FFD_std=x, fitness=predicted,temp=group)),
  (data.frame(ggpredict(selection_1,
                        terms = c("ffd_std[all]","temp[7.95]",
                                  "year_fct[2018]")))%>%
    mutate(temp_cat=2)%>%
    dplyr::rename(FFD_std=x, fitness=predicted,temp=group)),
  (data.frame(ggpredict(selection_1,
                        terms = c("ffd_std[all]","temp[12.45]",
                                  "year_fct[2018]")))%>%
    mutate(temp_cat=3)%>%
    dplyr::rename(FFD_std=x, fitness=predicted,temp=group)),
  (data.frame(ggpredict(selection_1,
                        terms = c("ffd_std[all]","temp[23.28846]",
                                  "year_fct[2018]")))%>%
    mutate(temp_cat=4)%>%
    dplyr::rename(FFD_std=x, fitness=predicted,temp=group)))


label_names1 <- list(
  '1'="First quarter\nMean temperature = 6.7ºC",
  '2'="Second quarter\nMean temperature = 8.7ºC",
  '3'="Third quarter\nMean temperature = 16.2ºC",
  '4'="Fourth quarter\nMean temperature = 25.1ºC"
)

labeller_function1 <- function(variable,value){
  return(label_names1[value])
}
```

```r
leg <- as_ggplot(get_legend(ggplot(subset(data.frame(data_plants),
                                          year==2017)%>%
         # Define 4 temp categories based on quartiles
         mutate(temp_cat=as.factor(
           ifelse(temp<=7.4,1,
                  ifelse(temp>7.4&temp<=10.7,2,
                         ifelse(temp>10.7&temp<=20.6,3,4))))),
```

```
      aes(x=ffd_std,y=nseed_rel))+
  facet_grid(~temp_cat,scales="free",
             labeller=labeller(temp_cat=labeller_function1))+
  geom_jitter(size=1.5,alpha=0.3,width=0.05)+
  geom_line(data=pred_fitness_17_cats,
            aes(x=FFD_std,y=fitness,color=temp_cat),size=1)+
  geom_ribbon(data=pred_fitness_17_cats,aes(x=FFD_std,y=fitness,
                                   ymin=conf.low,ymax=conf.high,
                                   fill=temp_cat),alpha=0.3)+
  my_theme()+scale_color_viridis(labels=NULL)+scale_fill_viridis(labels=NULL)+
  theme(legend.position="top")+labs(colour="Temperature (ºC)      ")+
  xlab("Standardized FFD")+
  ylab("Relative fitness")+
  #scale_x_continuous(breaks=c(-4,-2,0,2,4,6,8))+
  theme(strip.text.x=element_text(margin=margin(2,0,2,0)))+
  guides(fill=FALSE)+ggtitle("A) 2017")))

AppS4<-grid.arrange(leg,
             ggplot(subset(data.frame(data_plants),year==2017)%>%
         # Define 4 temp categories based on quartiles
         mutate(temp_cat=as.factor(
           ifelse(temp<=7.4,1,
                  ifelse(temp>7.4&temp<=10.7,2,
                         ifelse(temp>10.7&temp<=20.6,3,4))))),
      aes(x=ffd_std,y=nseed_rel))+
  facet_grid(~temp_cat,scales="free",
             labeller=labeller(temp_cat=labeller_function1))+
  geom_jitter(size=1.5,alpha=0.3,width=0.05)+
  geom_line(data=pred_fitness_17_cats,
            aes(x=FFD_std,y=fitness,color=temp_cat),size=1)+
  geom_ribbon(data=pred_fitness_17_cats,aes(x=FFD_std,y=fitness,
                                   ymin=conf.low,ymax=conf.high,
                                   fill=temp_cat),alpha=0.3)+
  my_theme()+scale_color_viridis(labels=NULL)+scale_fill_viridis(labels=NULL)+
  labs(colour="Temperature (ºC)      ")+
  xlab("Standardized FFD")+
  ylab("Relative fitness")+
  #scale_x_continuous(breaks=c(-4,-2,0,2,4,6,8))+
  theme(strip.text.x=element_text(margin=margin(2,0,2,0)))+
  guides(fill=FALSE)+ggtitle("A) 2017"),
  ggplot(subset(data.frame(data_plants),year==2018)%>%
         # Define 4 temp categories based on quartiles
         mutate(temp_cat=as.factor(
           ifelse(temp<=6.875,1,
                  ifelse(temp>6.875&temp<=9.300,2,
                         ifelse(temp>9.300&temp<=17.075,3,4))))),
      aes(x=ffd_std,y=nseed_rel))+
  facet_grid(~temp_cat,scales="free",
             labeller=labeller(temp_cat=labeller_function1))+
  geom_jitter(size=1.5,alpha=0.3,width=0.05)+
  geom_line(data=pred_fitness_18_cats,
            aes(x=FFD_std,y=fitness,color=temp_cat),size=1)+
  geom_ribbon(data=pred_fitness_18_cats,aes(x=FFD_std,y=fitness,
```

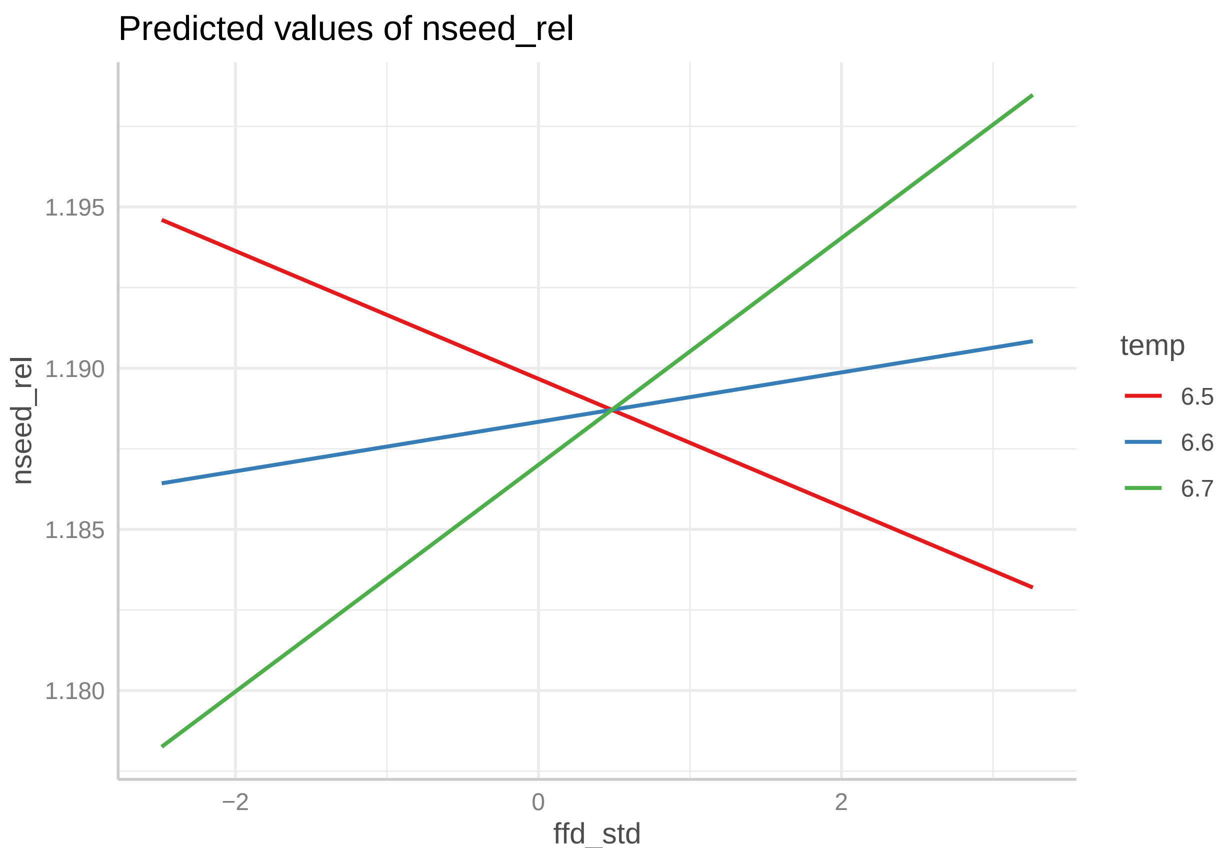
```
                                  ymin=conf.low,ymax=conf.high,
                                  fill=temp_cat),alpha=0.3)+
  my_theme()+scale_color_viridis(labels=NULL)+scale_fill_viridis(labels=NULL)+
  labs(colour="Temperature (ºC)      ")+
  xlab("Standardized FFD")+
  ylab("Relative fitness")+
  #scale_x_continuous(breaks=c(-4,-2,0,2,4,6,8))+
  theme(strip.text.x=element_text(margin=margin(2,0,2,0)))+
  guides(fill=FALSE)+ggtitle("B) 2018"),
                        ncol=1,heights=c(0.1,1,1))
```




```
ggsave(filename="output/figures/AppS4.tiff",plot=AppS4,
       width=26,height=20,units="cm",dpi=300)
```

Predictions of fitness:

```
plot(ggpredict(selection_1,
               terms=c("ffd_std[all]","temp[6.5:6.7 by=0.1]","year_fct[2018]")),
     ci=F)
```



In 2018, the model predicted that selection favoured earlier flowering at soil temperatures up to 6.5 ºC, while later flowering was favoured at higher soil temperatures.

Effect of temperature on the relationship absolute fitness-FFD

Keep separate models for both years here!

```
selectionabs_2017_1<-lm(nseed~ffd*temp+ffd*I(temp^2)+log(nfl),
                        subset(data_plants,year==2017))
selectionabs_2018_1<-lm(nseed~ffd*temp+ffd*I(temp^2)+log(nfl),
                        subset(data_plants,year==2018))
summ(selectionabs_2017_1)
```

Observations	245
Dependent variable	nseed
Type	OLS linear regression

F(6,238)	34.508
R^2	0.465
Adj. R^2	0.452

	Est.	S.E.	t val.	p
(Intercept)	-6682.670	4879.792	-1.369	0.172
ffd	35.520	27.405	1.296	0.196
temp	394.423	560.245	0.704	0.482
I(temp^2)	-8.887	15.022	-0.592	0.555
log(nfl)	625.843	45.793	13.667	0.000
ffd:temp	-2.451	3.181	-0.770	0.442
ffd:I(temp^2)	0.054	0.086	0.627	0.531

Standard errors: OLS

```
summ(selectionabs_2018_1)
```

Observations	104
Dependent variable	nseed
Type	OLS linear regression

F(6,97)	22.809
R^2	0.585
Adj. R^2	0.560

	Est.	S.E.	t val.	p
(Intercept)	1263.606	1411.947	0.895	0.373
ffd	-7.074	7.767	-0.911	0.365
temp	-134.639	192.293	-0.700	0.485
I(temp^2)	1.104	5.710	0.193	0.847
log(nfl)	197.641	20.943	9.437	0.000
ffd:temp	0.676	1.101	0.614	0.541
ffd:I(temp^2)	-0.005	0.033	-0.146	0.884

Standard errors: OLS

Quadratic terms of ffd not significant. Refit models withouth quadratic terms of ffd.

```
selectionabs_2017_2<-lm(nseed~ffd*temp+log(nfl),subset(data_plants,year==2017))
selectionabs_2018_2<-lm(nseed~ffd*temp+log(nfl),subset(data_plants,year==2018))
summ(selectionabs_2017_2)
```

Observations	245
Dependent variable	nseed
Type	OLS linear regression

F(4,240)	51.549
R^2	0.462
Adj. R^2	0.453

	Est.	S.E.	t val.	p
(Intercept)	-4647.341	2606.895	-1.783	0.076
ffd	23.284	14.531	1.602	0.110
temp	88.601	124.771	0.710	0.478
log(nfl)	619.124	45.314	13.663	0.000
ffd:temp	-0.601	0.709	-0.848	0.397

Standard errors: OLS

```
summ(selectionabs_2018_2)
```

Observations	104
Dependent variable	nseed
Type	OLS linear regression

F(4,99)	34.660
R^2	0.583
Adj. R^2	0.567

	Est.	S.E.	t val.	p
(Intercept)	755.873	607.739	1.244	0.217
ffd	-4.561	3.188	-1.431	0.156
temp	-81.835	33.624	-2.434	0.017
log(nfl)	198.339	20.492	9.679	0.000
ffd:temp	0.425	0.190	2.241	0.027

Standard errors: OLS

BCa intervals

Used for assessing significance.

2017

```
# ffd
slp <- function(selectionabs_2017_2) coef(selectionabs_2017_2)[2]
b <- car::Boot(selectionabs_2017_2,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
ffd_ci_17_abs <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# temp
slp <- function(selectionabs_2017_2) coef(selectionabs_2017_2)[3]
b <- car::Boot(selectionabs_2017_2,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
temp_ci_17_abs <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# nfl
slp <- function(selectionabs_2017_2) coef(selectionabs_2017_2)[4]
b <- car::Boot(selectionabs_2017_2,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
nfl_ci_17_abs <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# ffd:temp
slp <- function(selectionabs_2017_2) coef(selectionabs_2017_2)[5]
b <- car::Boot(selectionabs_2017_2,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
ffd_temp_ci_17_abs <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# Save confidence intervals as a table
BCIs_selection_2017_abs <- cbind(
  rbind(ffd_ci_17_abs[1,] ,temp_ci_17_abs[1,], nfl_ci_17_abs[1,],
        ffd_temp_ci_17_abs[1,]),
  rbind(ffd_ci_17_abs[2,] ,temp_ci_17_abs[2,], nfl_ci_17_abs[2,],
        ffd_temp_ci_17_abs[2,])
)
colnames(BCIs_selection_2017_abs)<-c("lower","upper")
rownames(BCIs_selection_2017_abs) <- c("ffd","temp","nfl","ffd:temp")
save(BCIs_selection_2017_abs,file="output/BCIs_selection_2017_abs.RData")
```

```
BCIs_selection_2017_abs
```

```
##                 lower       upper
## ffd         0.5810225  54.7350047
## temp      -77.8599588 299.5825479
## nfl       463.9592583 860.4155048
## ffd:temp   -1.7992499   0.3743141
```

2018

```
# ffd
slp <- function(selectionabs_2018_2) coef(selectionabs_2018_2)[2]
b <- car::Boot(selectionabs_2018_2,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
ffd_ci_18_abs <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# temp
slp <- function(selectionabs_2018_2) coef(selectionabs_2018_2)[3]
b <- car::Boot(selectionabs_2018_2,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
temp_ci_18_abs <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```r
# nfl
slp <- function(selectionabs_2018_2) coef(selectionabs_2018_2)[4]
b <- car::Boot(selectionabs_2018_2,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
nfl_ci_18_abs <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```r
# ffd:temp
slp <- function(selectionabs_2018_2) coef(selectionabs_2018_2)[5]
b <- car::Boot(selectionabs_2018_2,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
ffd_temp_ci_18_abs <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```r
# Save confidence intervals as a table
BCIs_selection_2018_abs <- cbind(
  rbind(ffd_ci_18_abs[1,] ,temp_ci_18_abs[1,], nfl_ci_18_abs[1,],
        ffd_temp_ci_18_abs[1,]),
  rbind(ffd_ci_18_abs[2,] ,temp_ci_18_abs[2,], nfl_ci_18_abs[2,],
        ffd_temp_ci_18_abs[2,])
)
colnames(BCIs_selection_2018_abs)<-c("lower","upper")
rownames(BCIs_selection_2018_abs) <- c("ffd","temp","nfl","ffd:temp")
save(BCIs_selection_2018_abs,file="output/BCIs_selection_2018_abs.RData")
```

```r
BCIs_selection_2018_abs
```

```
##                 lower       upper
## ffd       -11.7641534   3.1112504
## temp     -153.4487867 -12.8579565
## nfl       149.9824774 256.8988748
## ffd:temp    0.0577084   0.8070981
```

Tests of residual spatial autocorrelation

Run separate models for each year and test for residual spatial autocorrelation in those.

Hypothesis 1

```r
FFD_2017<-lm(ffd~temp,subset(data_plants,year==2017))
FFD_2018<-lm(ffd~temp,subset(data_plants,year==2018))
summ(FFD_2017)
```

Observations	245
Dependent variable	ffd
Type	OLS linear regression

F(1,243)	29.049
R^2	0.107
Adj. R^2	0.103

	Est.	S.E.	t val.	p
(Intercept)	180.626	0.996	181.397	0.000
temp	-0.332	0.062	-5.390	0.000

Standard errors: OLS

```
summ(FFD_2018)
```

Observations	104
Dependent variable	ffd
Type	OLS linear regression

F(1,102)	48.102
R^2	0.320
Adj. R^2	0.314

	Est.	S.E.	t val.	p
(Intercept)	195.104	2.206	88.424	0.000
temp	-1.070	0.154	-6.936	0.000

Standard errors: OLS

Spatial correlograms

```
res_FFD_2017<-residuals(FFD_2017)
res_FFD_2018<-residuals(FFD_2018)
```

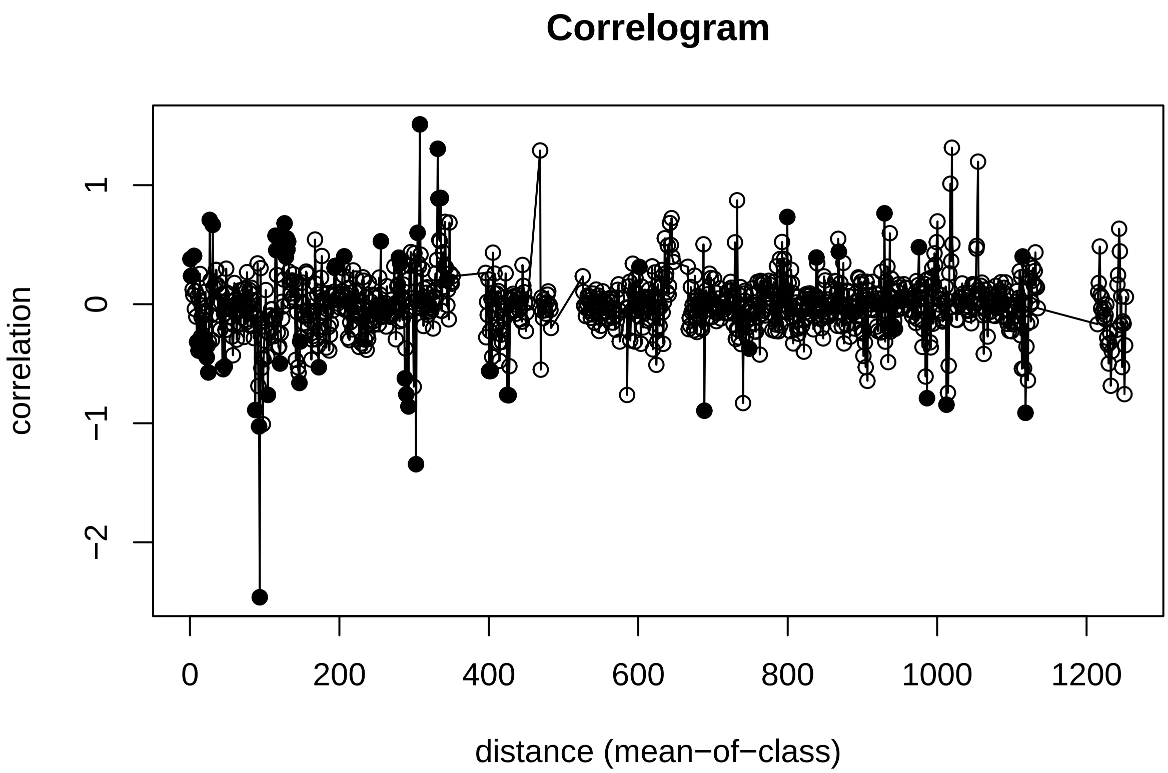
```
correlog_FFD_2017 <- correlog(x=subset(data_plants,year==2017)$x,
                              y=subset(data_plants,year==2017)$y,
                              res_FFD_2017,increment=1, resamp=100,quiet=F)
```

```
## 10  of  100 20  of  100 30  of  100 40  of  100 50  of  100 60  of  100 70  of  100 80  of  100 90  o
```

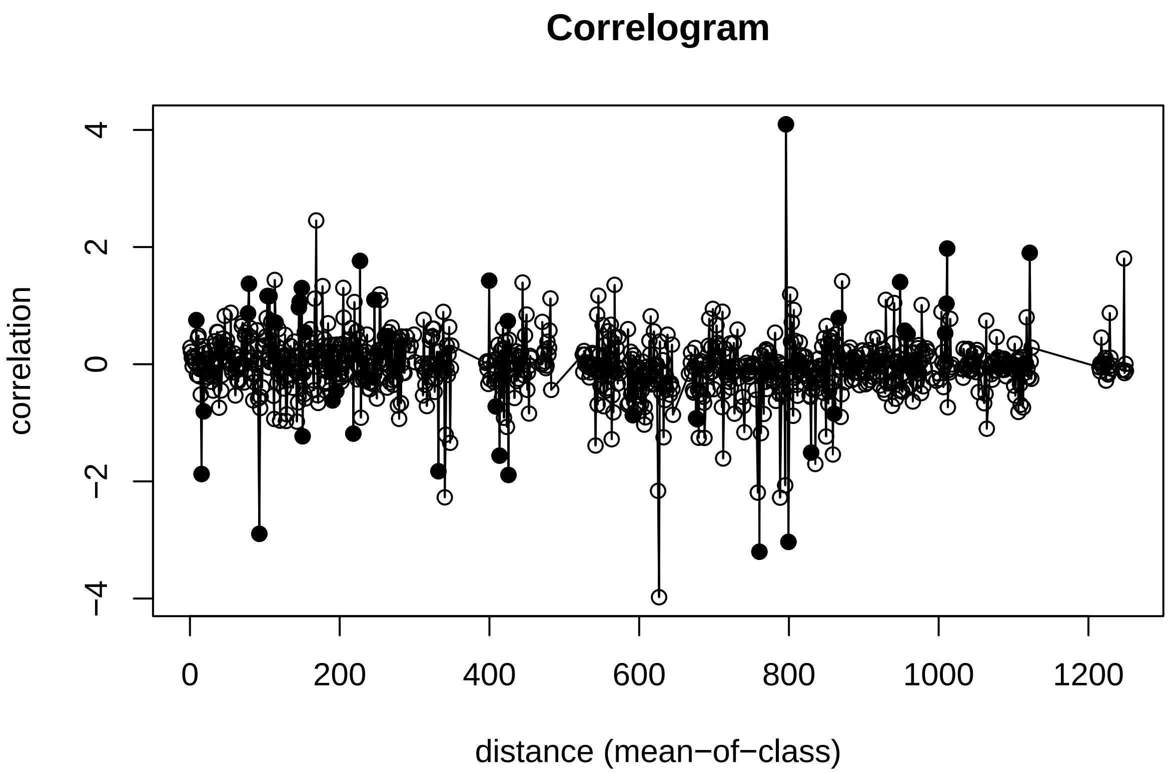
```
correlog_FFD_2018 <- correlog(x=subset(data_plants,year==2018)$x,
                              y=subset(data_plants,year==2018)$y,
                              res_FFD_2018,increment=1, resamp=100,quiet=F)
```

```
## 10  of  100 20  of  100 30  of  100 40  of  100 50  of  100 60  of  100 70  of  100 80  of  100 90  o
```

```
plot(correlog_FFD_2017)
```




```
plot(correlog_FFD_2018)
```



Moran's I

Calculation of global Moran's I with a permutation test (1000 random permutations), based on a connectivity matrix of pairwise Euclidean distances among the plants up to a distance of 30 m.

```
# Create neighbours matrix (30 m)
data_plants.listw_2017 <- nb2listw(dnearneigh(subset(data_plants,year==2017),
                                               0,30))
data_plants.listw_2018 <- nb2listw(dnearneigh(subset(data_plants,year==2018),
                                               0,30))
moran_FFD_2017<- moran.mc(res_FFD_2017, listw=data_plants.listw_2017,nsim=999)
moran_FFD_2018<- moran.mc(res_FFD_2018, listw=data_plants.listw_2018,nsim=999)
```

```
moran_FFD_2017 # Significant autocorrelation in the residuals
```

```
##
##  Monte-Carlo simulation of Moran I
##
## data:  res_FFD_2017
## weights: data_plants.listw_2017
## number of simulations + 1: 1000
##
## statistic = 0.04303, observed rank = 985, p-value = 0.015
## alternative hypothesis: greater
```

```r
moran_FFD_2018 # Significant autocorrelation in the residuals
```

```
##
##  Monte-Carlo simulation of Moran I
##
## data:  res_FFD_2018
## weights: data_plants.listw_2018
## number of simulations + 1: 1000
##
## statistic = 0.082696, observed rank = 956, p-value = 0.044
## alternative hypothesis: greater
```

Moran's eigenvector mapping

```r
ME.FFD_2017 <-ME(FFD_2017,listw=data_plants.listw_2017,
               data=subset(data_plants,year==2017),
               alpha=0.1,verbose=T)
```

```
## eV[,11], I: 0.01463088 ZI: NA, pr(ZI): 0.18
```

```r
ME.FFD_2018 <-ME(FFD_2018,listw=data_plants.listw_2018,
               data=subset(data_plants,year==2018),
               alpha=0.1,verbose=T)
```

```
## eV[,4], I: 0.02947613 ZI: NA, pr(ZI): 0.21
```

```r
vector1_2017_FFD<-ME.FFD_2017$vectors[,1]
vector1_2018_FFD<-ME.FFD_2018$vectors[,1]

FFD_2017_ME<-lm(ffd~temp+vector1_2017_FFD,
                subset(data_plants,year==2017))
FFD_2018_ME<-lm(ffd~temp+vector1_2018_FFD,subset(data_plants,year==2018))
summ(FFD_2017_ME)
```

Observations	245
Dependent variable	ffd
Type	OLS linear regression

F(2,242)	18.245
R^2	0.131
Adj. R^2	0.124

```r
summ(FFD_2018_ME)
```

Tests of residual spatial autocorrelation

	Est.	S.E.	t val.	p
(Intercept)	180.866	0.988	182.973	0.000
temp	-0.349	0.061	-5.700	0.000
vector1_2017_FFD	19.614	7.548	2.599	0.010

Standard errors: OLS

Observations	104
Dependent variable	ffd
Type	OLS linear regression

F(2,101)	28.531
R^2	0.361
Adj. R^2	0.348

	Est.	S.E.	t val.	p
(Intercept)	193.920	2.200	88.129	0.000
temp	-0.974	0.155	-6.284	0.000
vector1_2018_FFD	28.781	11.368	2.532	0.013

Standard errors: OLS

```
res_FFD_2017_ME<-residuals(FFD_2017_ME)
res_FFD_2018_ME<-residuals(FFD_2018_ME)
```

```
correlog_FFD_2017_ME <- correlog(x=subset(data_plants,year==2017)$x,
                                 y=subset(data_plants,year==2017)$y,
                                 res_FFD_2017_ME,increment=1, resamp=100,quiet=F)
```

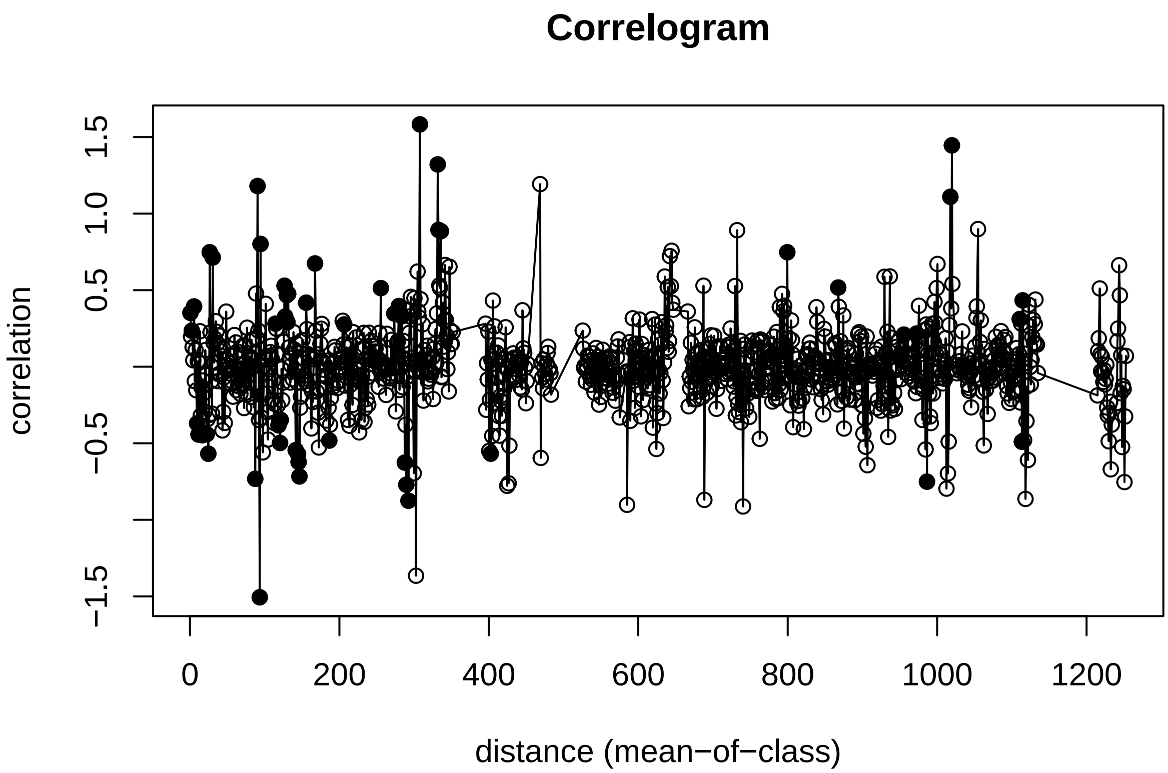
Spatial correlograms

```
## 10  of  100 20  of  100 30  of  100 40  of  100 50  of  100 60  of  100 70  of  100 80  of  100 90
```

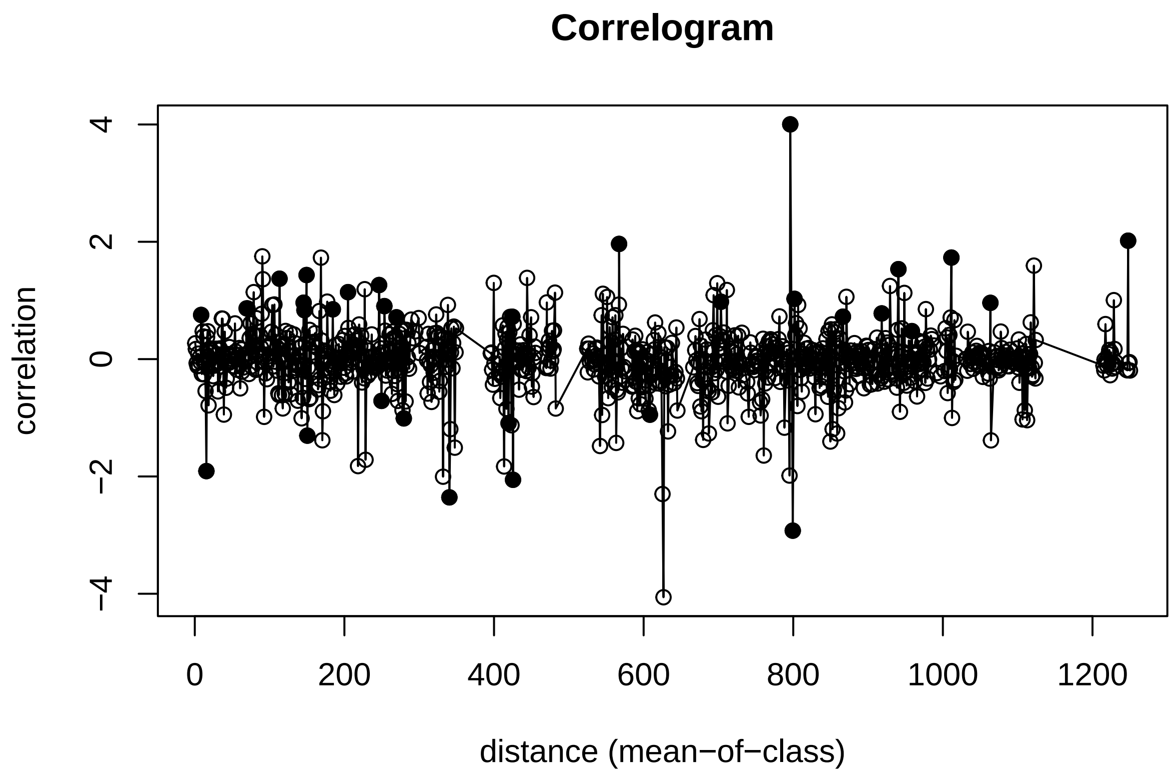
```
correlog_FFD_2018_ME <- correlog(x=subset(data_plants,year==2018)$x,
                                 y=subset(data_plants,year==2018)$y,
                                 res_FFD_2018_ME,increment=1, resamp=100,quiet=F)
```

```
## 10  of  100 20  of  100 30  of  100 40  of  100 50  of  100 60  of  100 70  of  100 80  of  100 90
```

```
plot(correlog_FFD_2017_ME)
```




```
plot(correlog_FFD_2018_ME)
```



Moran's I Calculation of global Moran's I with a permutation test (1000 random permutations), based on a connectivity matrix of pairwise Euclidean distances among the plants up to a distance of 30 m.

```
res_FFD_2017_ME<-residuals(FFD_2017_ME)
res_FFD_2018_ME<-residuals(FFD_2018_ME)
moran_FFD_2017_ME<- moran.mc(res_FFD_2017_ME, listw=data_plants.listw_2017,
                             nsim=999)
moran_FFD_2018_ME<- moran.mc(res_FFD_2018_ME, listw=data_plants.listw_2018,
                             nsim=999)
```

```
moran_FFD_2017_ME # No significant autocorrelation in the residuals
```

```
##
##  Monte-Carlo simulation of Moran I
##
## data:  res_FFD_2017_ME
## weights: data_plants.listw_2017
## number of simulations + 1: 1000
##
## statistic = 0.014631, observed rank = 831, p-value = 0.169
## alternative hypothesis: greater
```

```
moran_FFD_2018_ME # No significant autocorrelation in the residuals
```

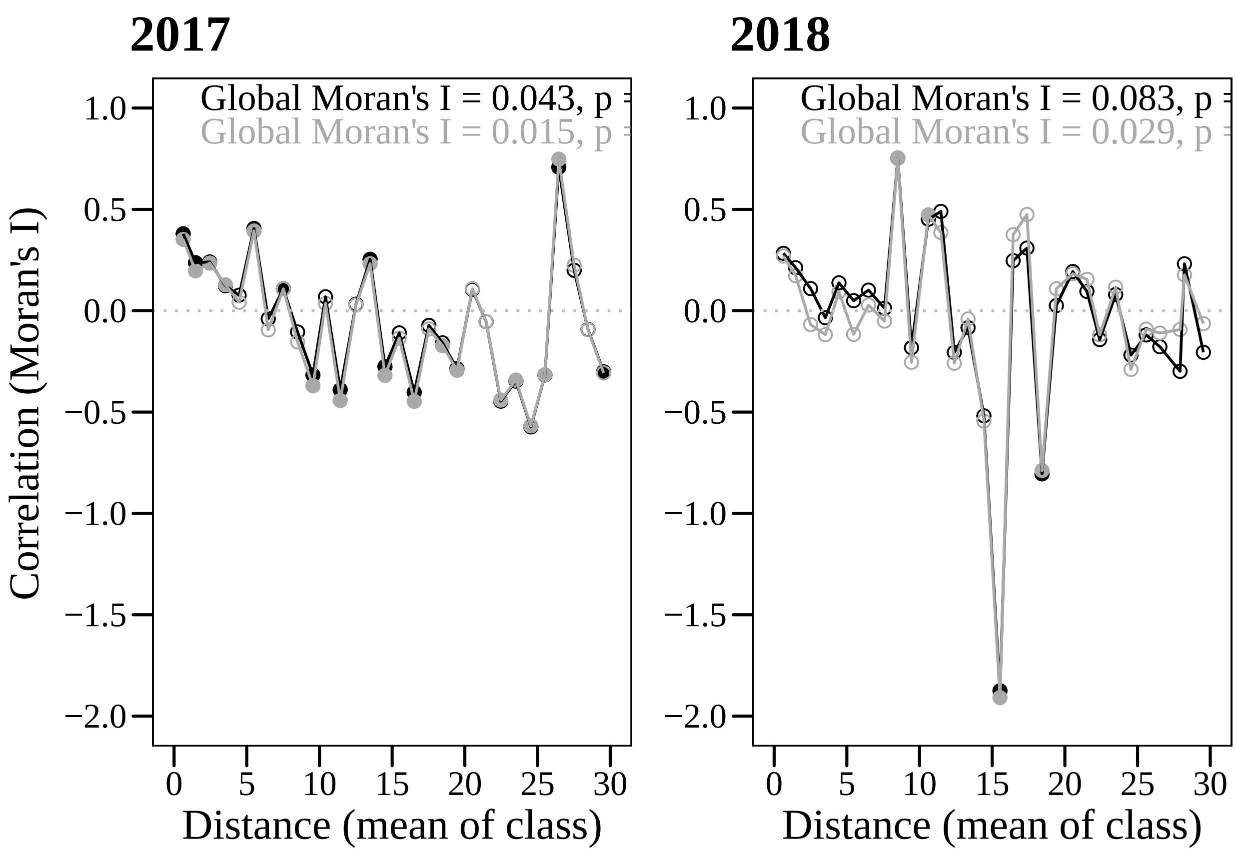
```
##
##  Monte-Carlo simulation of Moran I
##
## data:  res_FFD_2018_ME
## weights: data_plants.listw_2018
## number of simulations + 1: 1000
##
## statistic = 0.029476, observed rank = 795, p-value = 0.205
## alternative hypothesis: greater
```

AppS6 - Figure S1

```
# FFD_2017
corr_FFD_2017<-data.frame(cbind(distance=
                                as.vector(correlog_FFD_2017$mean.of.class[1:31]),
                      correlation=as.vector(correlog_FFD_2017$correlation[1:31]),
                      p=as.vector(correlog_FFD_2017$p[1:31])))
corr_FFD_2017_ME<-data.frame(cbind(distance=
                                   as.vector(
                                     correlog_FFD_2017_ME$mean.of.class[1:31]),
                                 correlation=
                                   as.vector(correlog_FFD_2017_ME$correlation[1:31]),
                                  p=as.vector(correlog_FFD_2017_ME$p[1:31])))
corr_FFD_2017$type<-"FFD_2017"
corr_FFD_2017_ME$type<-"FFD_2017_ME"
corr_FFD_2017<-rbind(corr_FFD_2017,corr_FFD_2017_ME)
corr_FFD_2017$sig<-as.factor(ifelse(corr_FFD_2017$p<0.05,1,0))

# FFD_2018
corr_FFD_2018<-data.frame(cbind(distance=
                                as.vector(correlog_FFD_2018$mean.of.class[1:31]),
                      correlation=as.vector(correlog_FFD_2018$correlation[1:31]),
                      p=as.vector(correlog_FFD_2018$p[1:31])))
corr_FFD_2018_ME<-data.frame(cbind(distance=
                                   as.vector(
                                     correlog_FFD_2018_ME$mean.of.class[1:31]),
                                 correlation=
                                   as.vector(correlog_FFD_2018_ME$correlation[1:31]),
                                  p=as.vector(correlog_FFD_2018_ME$p[1:31])))
corr_FFD_2018$type<-"FFD_2018"
corr_FFD_2018_ME$type<-"FFD_2018_ME"
corr_FFD_2018<-rbind(corr_FFD_2018,corr_FFD_2018_ME)
corr_FFD_2018$sig<-as.factor(ifelse(corr_FFD_2018$p<0.05,1,0))
```

```
AppS6_FigS1<-grid.arrange(
  ggplot(corr_FFD_2017,aes(x=distance, y=correlation)) +
    geom_point(aes(colour=type,shape=sig),size=2) +
    geom_line(aes(colour=type)) + ylab(NULL)+
    scale_x_continuous("Distance (mean of class)",limits=c(0,30),
                       breaks=c(0,5,10,15,20,25,30)) +
    scale_y_continuous(limits=c(-2,1),
                       breaks = seq(-2,1,0.5))+
```

```r
  scale_shape_manual(values=c(1,19))+
  scale_color_manual(values=c("black","darkgrey"))+
  geom_hline(aes(yintercept=0), colour="grey",linetype=3)+
  my_theme()+ggtitle("2017")+
  annotation_custom(grobTree(textGrob("Global Moran's I = 0.043, p = 0.025",
                                      x=0.1,y=0.97,hjust=0,
                                      gp=gpar(col="black",fontsize=14,
                                              fontfamily="serif"))))+
  annotation_custom(grobTree(textGrob("Global Moran's I = 0.015, p = 0.156",
                                      x=0.1,y=0.92,hjust=0,
                                      gp=gpar(col="darkgrey",fontsize=14,
                                              fontfamily="serif")))),
ggplot(corr_FFD_2018,aes(x=distance, y=correlation)) +
  geom_point(aes(colour=type,shape=sig),size=2) +
  geom_line(aes(colour=type)) + ylab(NULL)+
  scale_x_continuous("Distance (mean of class)",limits=c(0,30),
                     breaks=c(0,5,10,15,20,25,30)) +
  scale_y_continuous(limits=c(-2,1),
                     breaks = seq(-2,1,0.5))+
  scale_shape_manual(values=c(1,19))+
  scale_color_manual(values=c("black","darkgrey"))+
  geom_hline(aes(yintercept=0), colour="grey",linetype=3)+
  my_theme()+ggtitle("2018")+
  annotation_custom(grobTree(textGrob("Global Moran's I = 0.083, p = 0.043",
                                      x=0.1,y=0.97,hjust=0,
                                      gp=gpar(col="black",fontsize=14,
                                              fontfamily="serif"))))+
  annotation_custom(grobTree(textGrob("Global Moran's I = 0.029, p = 0.213",
                                      x=0.1,y=0.92,hjust=0,
                                      gp=gpar(col="darkgrey",fontsize=14,
                                              fontfamily="serif")))),
ncol=2,left=textGrob("Correlation (Moran's I)",just="center",
                    hjust=0.42,
                    gp=gpar(fontsize=16,fontfamily="serif"),
                    rot = 90))
```




```
ggsave(filename="output/figures/AppS6_FigS1.tiff",
       plot=AppS6_FigS1,device="tiff",width=28,height=12,units="cm",dpi=300,
       compression="lzw")
```

Hypothesis 2

```
fitness_2017<-glm.nb(n_seed_round~temp+log(nfl),
                      subset(data_plants,year==2017))
fitness_2018<-glm.nb(n_seed_round~temp+log(nfl),
                      subset(data_plants,year==2018))
# No quadratic terms of temperature because they were
# non-significant in both models
summ(fitness_2017,vif=T)
```

Observations	245
Dependent variable	n_seed_round
Type	Generalized linear model
Family	Negative Binomial(2.0993)
Link	log

χ^2()	0.722	0.088	3273.106	3287.112
Pseudo-R^2 (Cragg-Uhler)	0.722	0.088	3273.106	3287.112
Pseudo-R^2 (McFadden)	0.722	0.088	3273.106	3287.112
AIC	0.722	0.088	3273.106	3287.112
BIC	0.722	0.088	3273.106	3287.112

	Est.	S.E.	z val.	p	VIF
(Intercept)	4.079	0.113	36.029	0.000	NA
temp	-0.030	0.006	-5.117	0.000	1.094
log(nfl)	0.982	0.042	23.377	0.000	1.094

Standard errors: MLE

```
summ(fitness_2018,vif=T)
```

Observations	104
Dependent variable	n_seed_round
Type	Generalized linear model
Family	Negative Binomial(1.9313)
Link	log

χ^2()	0.684	0.091	1203.369	1213.947
Pseudo-R^2 (Cragg-Uhler)	0.684	0.091	1203.369	1213.947
Pseudo-R^2 (McFadden)	0.684	0.091	1203.369	1213.947
AIC	0.684	0.091	1203.369	1213.947
BIC	0.684	0.091	1203.369	1213.947

	Est.	S.E.	z val.	p	VIF
(Intercept)	3.543	0.188	18.878	0.000	NA
temp	-0.042	0.010	-4.166	0.000	1.003
log(nfl)	0.981	0.069	14.202	0.000	1.003

Standard errors: MLE

Spatial correlograms

```
res_fitness_2017<-residuals(fitness_2017)
res_fitness_2018<-residuals(fitness_2018)
```

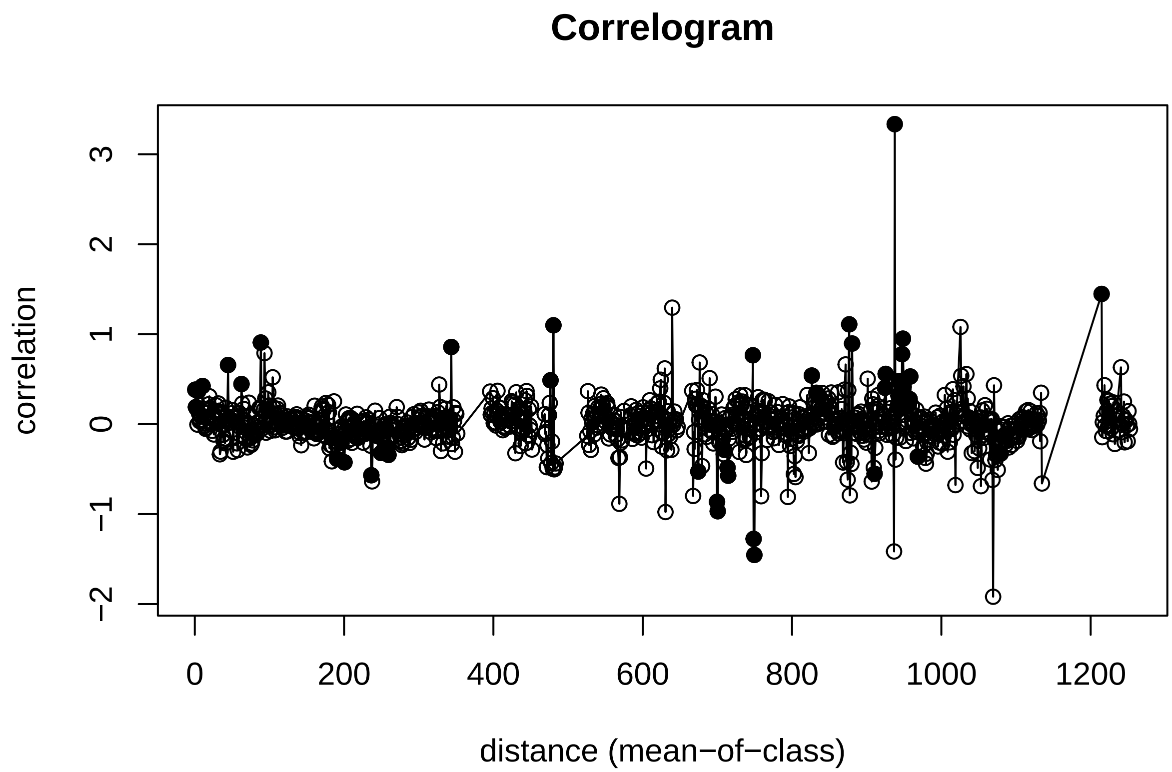
```
correlog_fitness_2017 <- correlog(x=subset(data_plants,year==2017)$x,
                        y=subset(data_plants,year==2017)$y,
                        res_fitness_2017,increment=1, resamp=100,quiet=F)
```

```
## 10  of  100 20  of  100 30  of  100 40  of  100 50  of  100 60  of  100 70  of  100 80  of  100 90
```

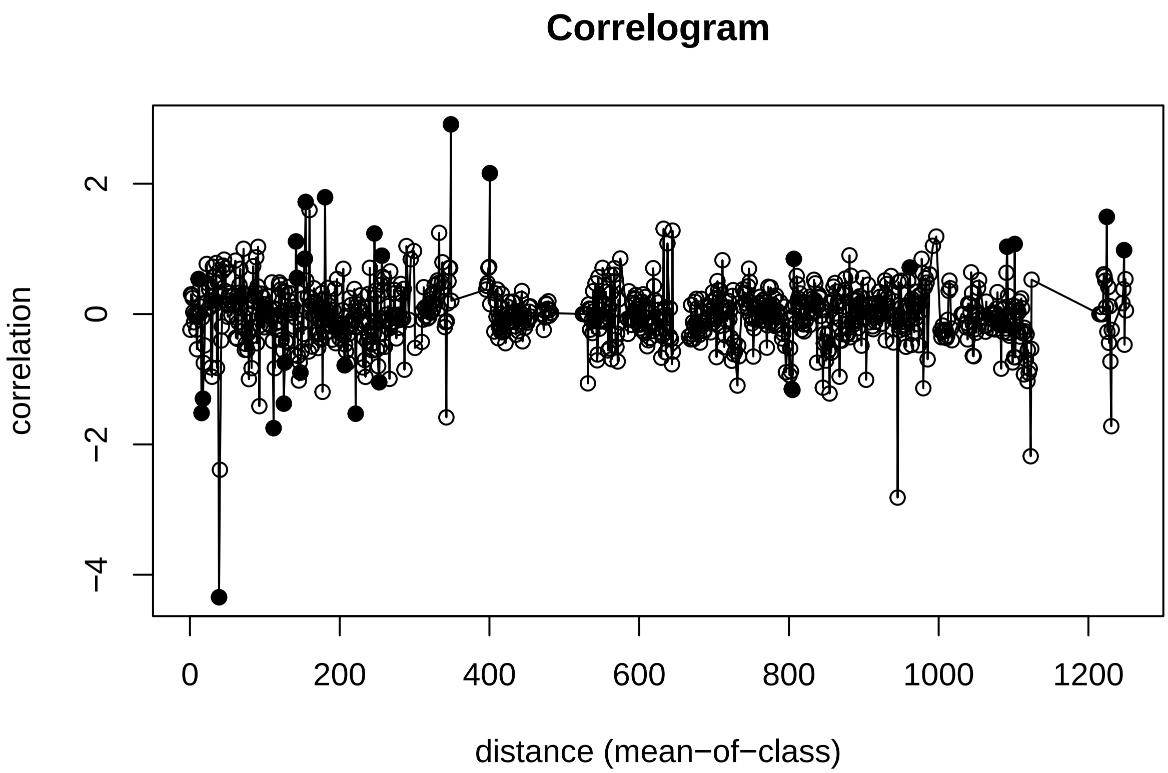
```
correlog_fitness_2018 <- correlog(x=subset(data_plants,year==2018)$x,
                             y=subset(data_plants,year==2018)$y,
                             res_fitness_2018,increment=1, resamp=100,quiet=F)
```

```
## 10  of  100 20  of  100 30  of  100 40  of  100 50  of  100 60  of  100 70  of  100 80  of  100 90  o
```

```
plot(correlog_fitness_2017)
```




```
plot(correlog_fitness_2018)
```



Moran's I

Calculation of global Moran's I with a permutation test (1000 random permutations), based on a connectivity matrix of pairwise Euclidean distances among the plants up to a distance of 30 m.

```
moran_fitness_2017<- moran.mc(res_fitness_2017,
                              listw=data_plants.listw_2017,nsim=999)
moran_fitness_2018<- moran.mc(res_fitness_2018,
                              listw=data_plants.listw_2018,nsim=999)
```

```
moran_fitness_2017 # Significant autocorrelation in the residuals
```

```
##
##  Monte-Carlo simulation of Moran I
##
## data:  res_fitness_2017
## weights: data_plants.listw_2017
## number of simulations + 1: 1000
##
## statistic = 0.12186, observed rank = 1000, p-value = 0.001
## alternative hypothesis: greater
```

```
moran_fitness_2018 # No significant autocorrelation in the residuals
```

```
##
##  Monte-Carlo simulation of Moran I
##
## data:  res_fitness_2018
## weights: data_plants.listw_2018
## number of simulations + 1: 1000
##
## statistic = 0.054454, observed rank = 888, p-value = 0.112
## alternative hypothesis: greater
```

Moran's eigenvector mapping

```
ME.fitness_2017 <-ME(fitness_2017,listw=data_plants.listw_2017,
               data=subset(data_plants,year==2017),
               alpha=0.1,verbose=T)
```

```
## eV[,3], I: 0.03874708 ZI: NA, pr(ZI): 0.06
## eV[,11], I: 0.01278019 ZI: NA, pr(ZI): 0.19
```

```
vector1_2017_fitness<-ME.fitness_2017$vectors[,1]
vector2_2017_fitness<-ME.fitness_2017$vectors[,2]

fitness_2017_ME<-glm.nb(n_seed_round~temp+log(nfl)+
                          vector1_2017_fitness+vector2_2017_fitness,
                        subset(data_plants,year==2017))
summ(fitness_2017_ME)
```

Observations	245
Dependent variable	n_seed_round
Type	Generalized linear model
Family	Negative Binomial(2.2191)
Link	log

$\chi^2()$	0.739	0.092	3261.922	3282.929
Pseudo-R^2 (Cragg-Uhler)	0.739	0.092	3261.922	3282.929
Pseudo-R^2 (McFadden)	0.739	0.092	3261.922	3282.929
AIC	0.739	0.092	3261.922	3282.929
BIC	0.739	0.092	3261.922	3282.929

Tests of residual spatial autocorrelation

```
res_fitness_2017_ME<-residuals(fitness_2017_ME)
```

	Est.	S.E.	z val.	p
(Intercept)	4.102	0.112	36.788	0.000
temp	-0.029	0.006	-5.027	0.000
log(nfl)	0.960	0.042	22.870	0.000
vector1_2017_fitness	-2.297	0.684	-3.360	0.001
vector2_2017_fitness	-1.418	0.693	-2.046	0.041

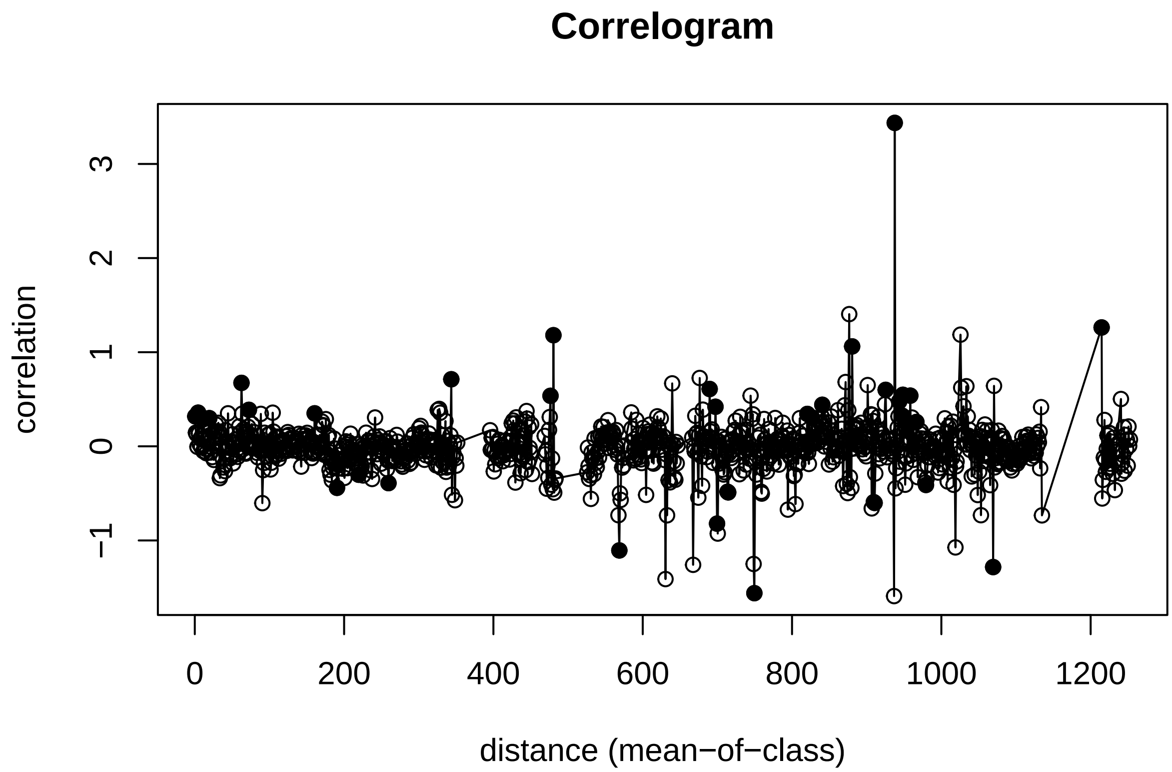
Standard errors: MLE

```
correlog_fitness_2017_ME <- correlog(x=subset(data_plants,year==2017)$x,
                                     y=subset(data_plants,year==2017)$y,
                                     res_fitness_2017_ME,increment=1, resamp=100,quiet=F)
```

Spatial correlograms

```
## 10  of  100 20  of  100 30  of  100 40  of  100 50  of  100 60  of  100 70  of  100 80  of  100 90
```

```
plot(correlog_fitness_2017_ME)
```



Moran's I Calculation of global Moran's I with a permutation test (1000 random permutations), based on a connectivity matrix of pairwise Euclidean distances among the plants up to a distance of 30 m.

```
moran_fitness_2017_ME<- moran.mc(res_fitness_2017_ME,
                                 listw=data_plants.listw_2017,nsim=999)
```

```
moran_fitness_2017_ME # STILL significant autocorrelation in the residuals
```

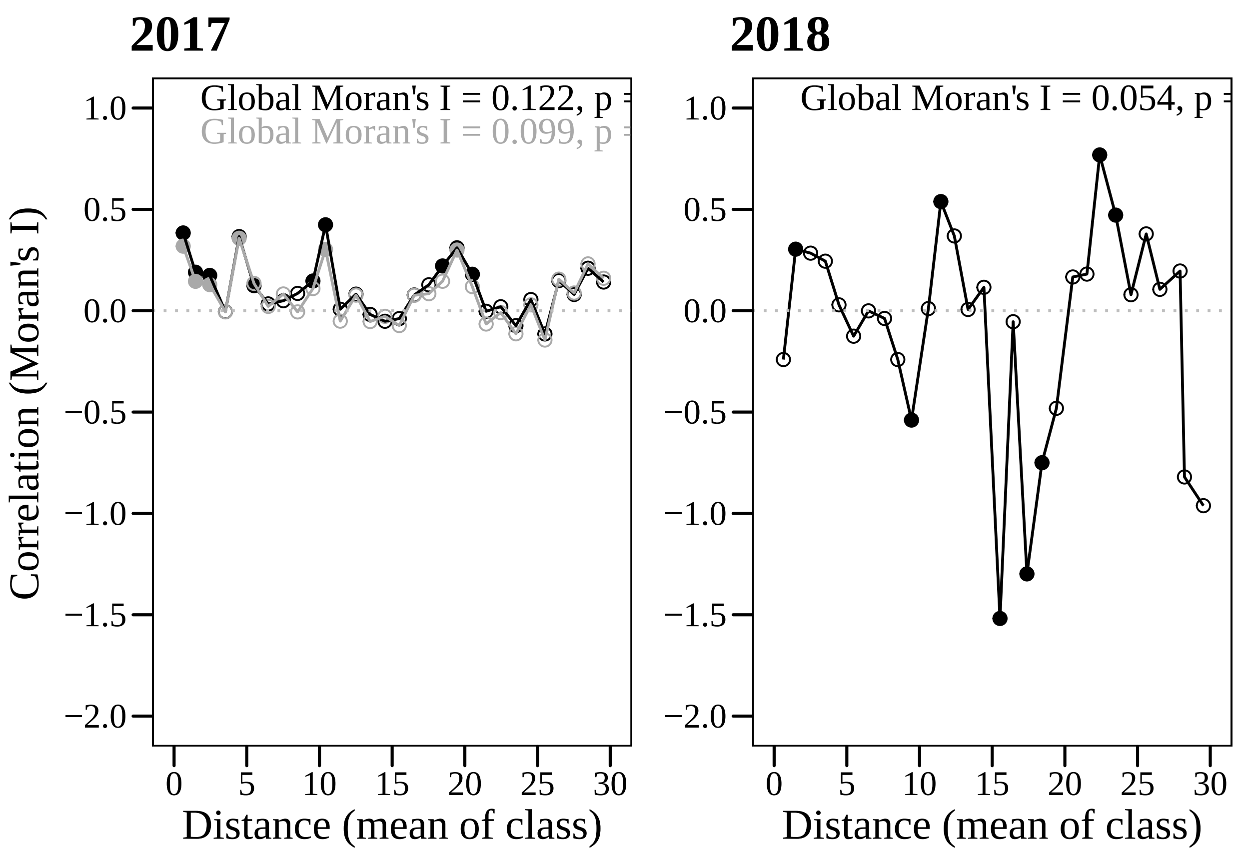
```
##
##  Monte-Carlo simulation of Moran I
##
## data:  res_fitness_2017_ME
## weights: data_plants.listw_2017
## number of simulations + 1: 1000
##
## statistic = 0.098683, observed rank = 1000, p-value = 0.001
## alternative hypothesis: greater
```

AppS6 - Figure S2

```
# fitness_2017
corr_fitness_2017<-data.frame(cbind(distance=
                                as.vector(correlog_fitness_2017$mean.of.class[1:31]),
                      correlation=as.vector(correlog_fitness_2017$correlation[1:31]),
                      p=as.vector(correlog_fitness_2017$p[1:31])))
corr_fitness_2017_ME<-data.frame(cbind(distance=
                                   as.vector(
                                     correlog_fitness_2017_ME$mean.of.class[1:31]),
                                 correlation=
                                   as.vector(correlog_fitness_2017_ME$correlation[1:31]),
                                  p=as.vector(correlog_fitness_2017_ME$p[1:31])))
corr_fitness_2017$type<-"fitness_2017"
corr_fitness_2017_ME$type<-"fitness_2017_ME"
corr_fitness_2017<-rbind(corr_fitness_2017,corr_fitness_2017_ME)
corr_fitness_2017$sig<-as.factor(ifelse(corr_fitness_2017$p<0.05,1,0))

# fitness_2018
corr_fitness_2018<-data.frame(cbind(distance=
                                as.vector(correlog_fitness_2018$mean.of.class[1:31]),
                      correlation=as.vector(correlog_fitness_2018$correlation[1:31]),
                      p=as.vector(correlog_fitness_2018$p[1:31])))
corr_fitness_2018$sig<-as.factor(ifelse(corr_fitness_2018$p<0.05,1,0))
```

```
AppS6_FigS2<-grid.arrange(
  ggplot(corr_fitness_2017,aes(x=distance, y=correlation)) +
    geom_point(aes(colour=type,shape=sig),size=2) +
    geom_line(aes(colour=type)) + ylab(NULL)+
    scale_x_continuous("Distance (mean of class)",limits=c(0,30),
                       breaks=c(0,5,10,15,20,25,30)) +
    scale_y_continuous(limits=c(-2,1),
                       breaks = seq(-2,1,0.5))+
    scale_shape_manual(values=c(1,19))+
    scale_color_manual(values=c("black","darkgrey"))+
    geom_hline(aes(yintercept=0), colour="grey",linetype=3)+
```

```
  my_theme()+ggtitle("2017")+
  annotation_custom(grobTree(textGrob("Global Moran's I = 0.122, p = 0.001",
                                      x=0.1,y=0.97,hjust=0,
                                      gp=gpar(col="black",fontsize=14,
                                              fontfamily="serif"))))+
  annotation_custom(grobTree(textGrob("Global Moran's I = 0.099, p = 0.001",
                                      x=0.1,y=0.92,hjust=0,
                                      gp=gpar(col="darkgrey",fontsize=14,
                                              fontfamily="serif")))),
ggplot(corr_fitness_2018,aes(x=distance, y=correlation)) +
  geom_point(aes(shape=sig),size=2,color="black") +
  geom_line(color="black") + ylab(NULL)+
  scale_x_continuous("Distance (mean of class)",limits=c(0,30),
                     breaks=c(0,5,10,15,20,25,30)) +
  scale_y_continuous(limits=c(-2,1),
                     breaks = seq(-2,1,0.5))+
  scale_shape_manual(values=c(1,19))+
  geom_hline(aes(yintercept=0), colour="grey",linetype=3)+
  my_theme()+ggtitle("2018")+
  annotation_custom(grobTree(textGrob("Global Moran's I = 0.054, p = 0.109",
                                      x=0.1,y=0.97,hjust=0,
                                      gp=gpar(col="black",fontsize=14,
                                              fontfamily="serif")))),
ncol=2,left=textGrob("Correlation (Moran's I)",just="center",
                    hjust=0.42,
                    gp=gpar(fontsize=16,fontfamily="serif"),
                    rot = 90))
```




```
ggsave(filename="output/figures/AppS6_FigS2.tiff",
       plot=AppS6_FigS2,device="tiff",width=28,height=12,units="cm",dpi=300,
       compression="lzw")
```

Hypothesis 3

```
selection_2017<-lm(nseed_rel~ffd_std*temp+nfl_std,
                          subset(data_plants,year==2017))
selection_2018<-lm(nseed_rel~ffd_std*temp+nfl_std,
                          subset(data_plants,year==2018))
summ(selection_2017,vif=T)
```

Observations	245
Dependent variable	nseed_rel
Type	OLS linear regression

F(4,240)	51.549
R^2	0.462
Adj. R^2	0.453

	Est.	S.E.	t val.	p	VIF
(Intercept)	1.408	0.166	8.460	0.000	NA
ffd_std	0.331	0.206	1.602	0.110	7.311
temp	-0.030	0.010	-2.937	0.004	1.151
nfl_std	1.213	0.089	13.663	0.000	1.354
ffd_std:temp	-0.009	0.010	-0.848	0.397	6.682

Standard errors: OLS

```
summ(selection_2018,vif=T)
```

Observations	104
Dependent variable	nseed_rel
Type	OLS linear regression

F(4,99)	34.660
R^2	0.583
Adj. R^2	0.567

	Est.	S.E.	t val.	p	VIF
(Intercept)	1.374	0.201	6.825	0.000	NA
ffd_std	-0.293	0.205	-1.431	0.156	5.999
temp	-0.021	0.016	-1.368	0.174	1.805
nfl_std	0.976	0.101	9.679	0.000	1.453
ffd_std:temp	0.027	0.012	2.241	0.027	5.976

Standard errors: OLS

BCA intervals

Used for assessing significance.

```
# ffd
slp <- function(selection_2017) coef(selection_2017)[2]
b <- car::Boot(selection_2017,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
ffd_ci_17 <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# temp
slp <- function(selection_2017) coef(selection_2017)[3]
b <- car::Boot(selection_2017,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
temp_ci_17 <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# nfl
slp <- function(selection_2017) coef(selection_2017)[4]
b <- car::Boot(selection_2017,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
nfl_ci_17 <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# ffd:temp
slp <- function(selection_2017) coef(selection_2017)[5]
b <- car::Boot(selection_2017,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
ffd_temp_ci_17 <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# Save confidence intervals as a table
BCIs_selection_2017 <- cbind(
  rbind(ffd_ci_17[1,] ,temp_ci_17[1,], nfl_ci_17[1,],
        ffd_temp_ci_17[1,]),
  rbind(ffd_ci_17[2,] ,temp_ci_17[2,], nfl_ci_17[2,],
        ffd_temp_ci_17[2,])
)
colnames(BCIs_selection_2017)<-c("lower","upper")
rownames(BCIs_selection_2017) <- c("ffd","temp","nfl","ffd:temp")
save(BCIs_selection_2017,file="output/BCIs_selection_2017.RData")
```

```
BCIs_selection_2017
```

2017

```
##                lower        upper
## ffd       0.02385641  0.798458215
## temp     -0.05521903 -0.012722425
## nfl       0.90770421  1.694343153
## ffd:temp -0.02624749  0.005367757
```

```
# ffd
slp <- function(selection_2018) coef(selection_2018)[2]
b <- car::Boot(selection_2018,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
ffd_ci_17 <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```r
# temp
slp <- function(selection_2018) coef(selection_2018)[3]
b <- car::Boot(selection_2018,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
temp_ci_17 <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```r
# nfl
slp <- function(selection_2018) coef(selection_2018)[4]
b <- car::Boot(selection_2018,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
nfl_ci_17 <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```r
# ffd:temp
slp <- function(selection_2018) coef(selection_2018)[5]
b <- car::Boot(selection_2018,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
ffd_temp_ci_17 <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```r
# Save confidence intervals as a table
BCIs_selection_2018 <- cbind(
  rbind(ffd_ci_17[1,] ,temp_ci_17[1,], nfl_ci_17[1,],
        ffd_temp_ci_17[1,]),
  rbind(ffd_ci_17[2,] ,temp_ci_17[2,], nfl_ci_17[2,],
        ffd_temp_ci_17[2,])
)
colnames(BCIs_selection_2018)<-c("lower","upper")
rownames(BCIs_selection_2018) <- c("ffd","temp","nfl","ffd:temp")
save(BCIs_selection_2018,file="output/BCIs_selection_2018.RData")
```

```r
BCIs_selection_2018
```

2018

```
##                  lower        upper
## ffd      -0.746581836  0.225209947
## temp     -0.041135154 -0.001714171
## nfl       0.734937706  1.261393143
## ffd:temp  0.004130652  0.051821909
```

Spatial correlograms

```
res_selection_2017<-residuals(selection_2017)
res_selection_2018<-residuals(selection_2018)
```

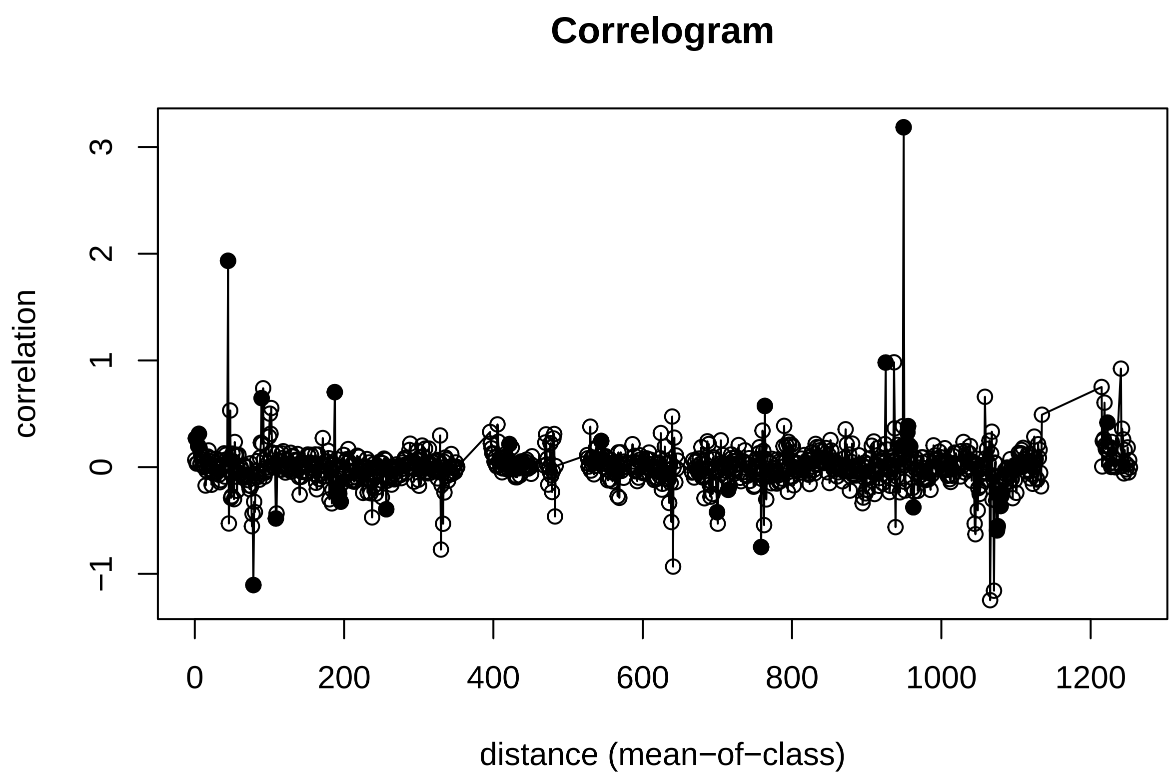
```
correlog_selection_2017 <- correlog(x=subset(data_plants,year==2017)$x,
                                     y=subset(data_plants,year==2017)$y,
                                     res_selection_2017,increment=1, resamp=100,quiet=F)
```

```
## 10  of  100 20  of  100 30  of  100 40  of  100 50  of  100 60  of  100 70  of  100 80  of  100 90  o
```

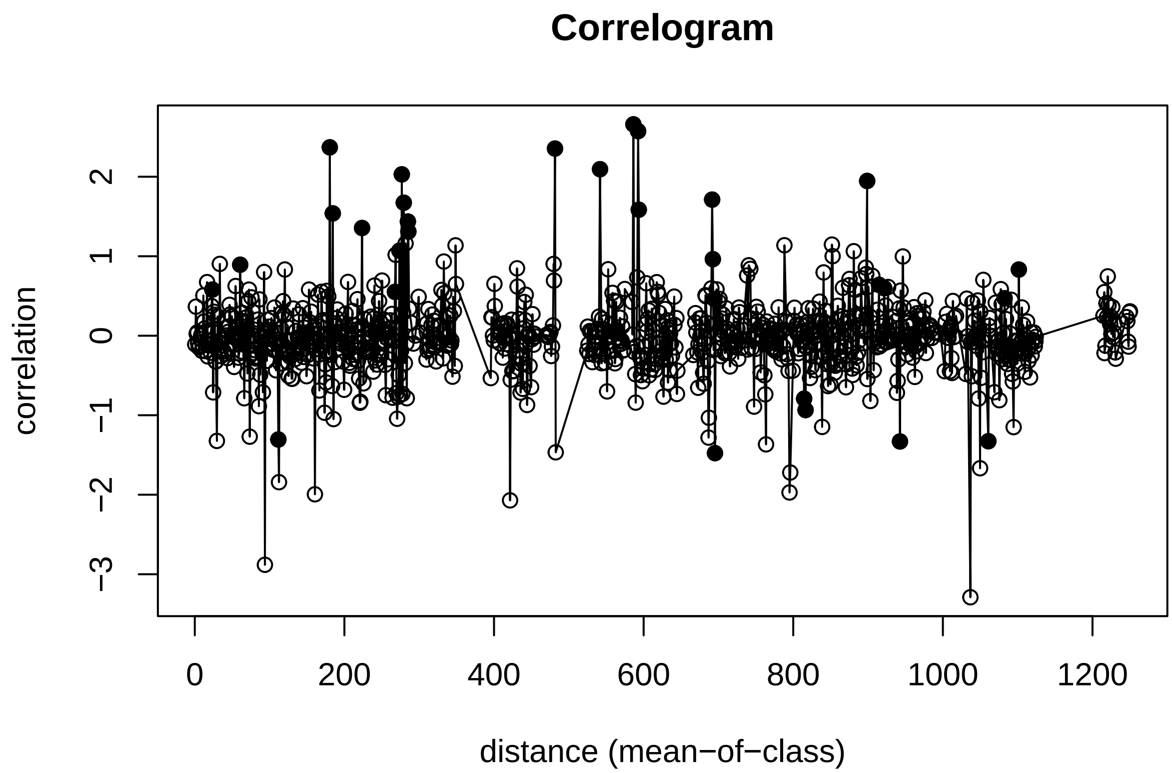
```
correlog_selection_2018 <- correlog(x=subset(data_plants,year==2018)$x,
                                     y=subset(data_plants,year==2018)$y,
                                     res_selection_2018,increment=1, resamp=100,quiet=F)
```

```
## 10  of  100 20  of  100 30  of  100 40  of  100 50  of  100 60  of  100 70  of  100 80  of  100 90  o
```

```
plot(correlog_selection_2017)
```




```
plot(correlog_selection_2018)
```



Moran's I

Calculation of global Moran's I with a permutation test (1000 random permutations), based on a connectivity matrix of pairwise Euclidean distances among the plants up to a distance of 30 m.

```
moran_selection_2017<- moran.mc(res_selection_2017,
                                 listw=data_plants.listw_2017,nsim=999)
moran_selection_2018<- moran.mc(res_selection_2018,
                                 listw=data_plants.listw_2018,nsim=999)
```

```
moran_selection_2017 # Significant autocorrelation in the residuals
```

```
##
##  Monte-Carlo simulation of Moran I
##
## data:  res_selection_2017
## weights: data_plants.listw_2017
## number of simulations + 1: 1000
##
## statistic = 0.097925, observed rank = 1000, p-value = 0.001
## alternative hypothesis: greater
```

```
moran_selection_2018 # No significant autocorrelation in the residuals
```

```
##
##  Monte-Carlo simulation of Moran I
##
## data:  res_selection_2018
## weights: data_plants.listw_2018
## number of simulations + 1: 1000
##
## statistic = 0.025308, observed rank = 792, p-value = 0.208
## alternative hypothesis: greater
```

Moran's eigenvector mapping

```
ME.selection_2017 <-ME(selection_2017,listw=data_plants.listw_2017,
               data=subset(data_plants,year==2017),
               alpha=0.05,verbose=T)
```

```
## eV[,3], I: 0.0498509 ZI: NA, pr(ZI): 0.04
## eV[,11], I: 0.01902798 ZI: NA, pr(ZI): 0.14
```

```
vector1_2017_selection<-ME.selection_2017$vectors[,1]
vector2_2017_selection<-ME.selection_2017$vectors[,2]

selection_2017_ME<-lm(nseed_rel~ffd_std*temp+nfl_std+
                        vector1_2017_selection+vector2_2017_selection,
                     subset(data_plants,year==2017))
summ(selection_2017_ME)
```

Observations	245
Dependent variable	nseed_rel
Type	OLS linear regression

F(6,238)	41.275
R^2	0.510
Adj. R^2	0.498

	Est.	S.E.	t val.	p
(Intercept)	1.377	0.163	8.456	0.000
ffd_std	0.422	0.200	2.113	0.036
temp	-0.029	0.010	-2.848	0.005
nfl_std	1.150	0.086	13.326	0.000
vector1_2017_selection	-4.244	1.158	-3.666	0.000
vector2_2017_selection	-3.725	1.188	-3.135	0.002
ffd_std:temp	-0.012	0.010	-1.284	0.200

Standard errors: OLS

BCA intervals Used for assessing significance.

```
# ffd
slp <- function(selection_2017_ME) coef(selection_2017_ME)[2]
b <- car::Boot(selection_2017_ME,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
ffd_ci_17 <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# temp
slp <- function(selection_2017_ME) coef(selection_2017_ME)[3]
b <- car::Boot(selection_2017_ME,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
temp_ci_17 <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# nfl
slp <- function(selection_2017_ME) coef(selection_2017_ME)[4]
b <- car::Boot(selection_2017_ME,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
nfl_ci_17 <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# vector1
slp <- function(selection_2017_ME) coef(selection_2017_ME)[5]
b <- car::Boot(selection_2017_ME,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
vector1_ci_17 <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# vector2
slp <- function(selection_2017_ME) coef(selection_2017_ME)[6]
b <- car::Boot(selection_2017_ME,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
vector2_ci_17 <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# ffd:temp
slp <- function(selection_2017_ME) coef(selection_2017_ME)[7]
b <- car::Boot(selection_2017_ME,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
ffd_temp_ci_17 <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```r
# Save confidence intervals as a table
BCIs_selection_2017_ME <- cbind(
  rbind(ffd_ci_17[1,] ,temp_ci_17[1,], nfl_ci_17[1,],vector1_ci_17[1,],
        vector2_ci_17[1,],ffd_temp_ci_17[1,]),
  rbind(ffd_ci_17[2,] ,temp_ci_17[2,], nfl_ci_17[2,],vector1_ci_17[2,],
        vector2_ci_17[2,],ffd_temp_ci_17[2,])
)
colnames(BCIs_selection_2017_ME)<-c("lower","upper")
rownames(BCIs_selection_2017_ME) <- c("ffd","temp","nfl",
                                      "vector1","vector2","ffd:temp")
save(BCIs_selection_2017_ME,file="output/BCIs_selection_2017_ME.RData")
```

```r
BCIs_selection_2017_ME
```

2017

```
##                lower        upper
## ffd       0.08341493  0.924404464
## temp     -0.05060868 -0.012215293
## nfl       0.88013003  1.512195683
## vector1  -6.95502029 -2.348393244
## vector2  -9.17287152 -0.337324697
## ffd:temp -0.03097270  0.001702003
```

Tests of residual spatial autocorrelation

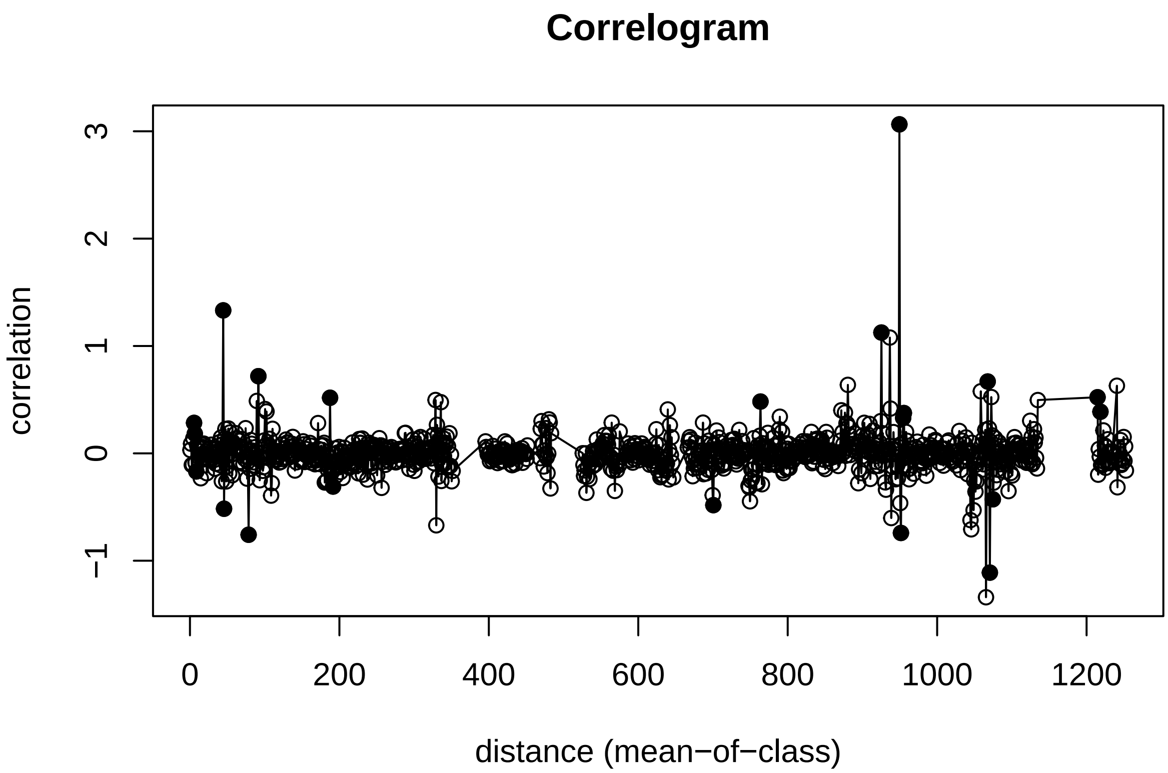
```r
res_selection_2017_ME<-residuals(selection_2017_ME)
```

```r
correlog_selection_2017_ME <- correlog(x=subset(data_plants,year==2017)$x,
                            y=subset(data_plants,year==2017)$y,
                            res_selection_2017_ME,increment=1, resamp=100,quiet=F)
```

Spatial correlograms

```
## 10  of  100 20  of  100 30  of  100 40  of  100 50  of  100 60  of  100 70  of  100 80  of  100 90  o
```

```r
plot(correlog_selection_2017_ME)
```



Moran's I Calculation of global Moran's I with a permutation test (1000 random permutations), based on a connectivity matrix of pairwise Euclidean distances among the plants up to a distance of 30 m.

```
res_selection_2017_ME<-residuals(selection_2017_ME)
moran_selection_2017_ME<- moran.mc(res_selection_2017_ME,
                                   listw=data_plants.listw_2017,nsim=999)
```

```
moran_selection_2017_ME # No significant autocorrelation in the residuals
```

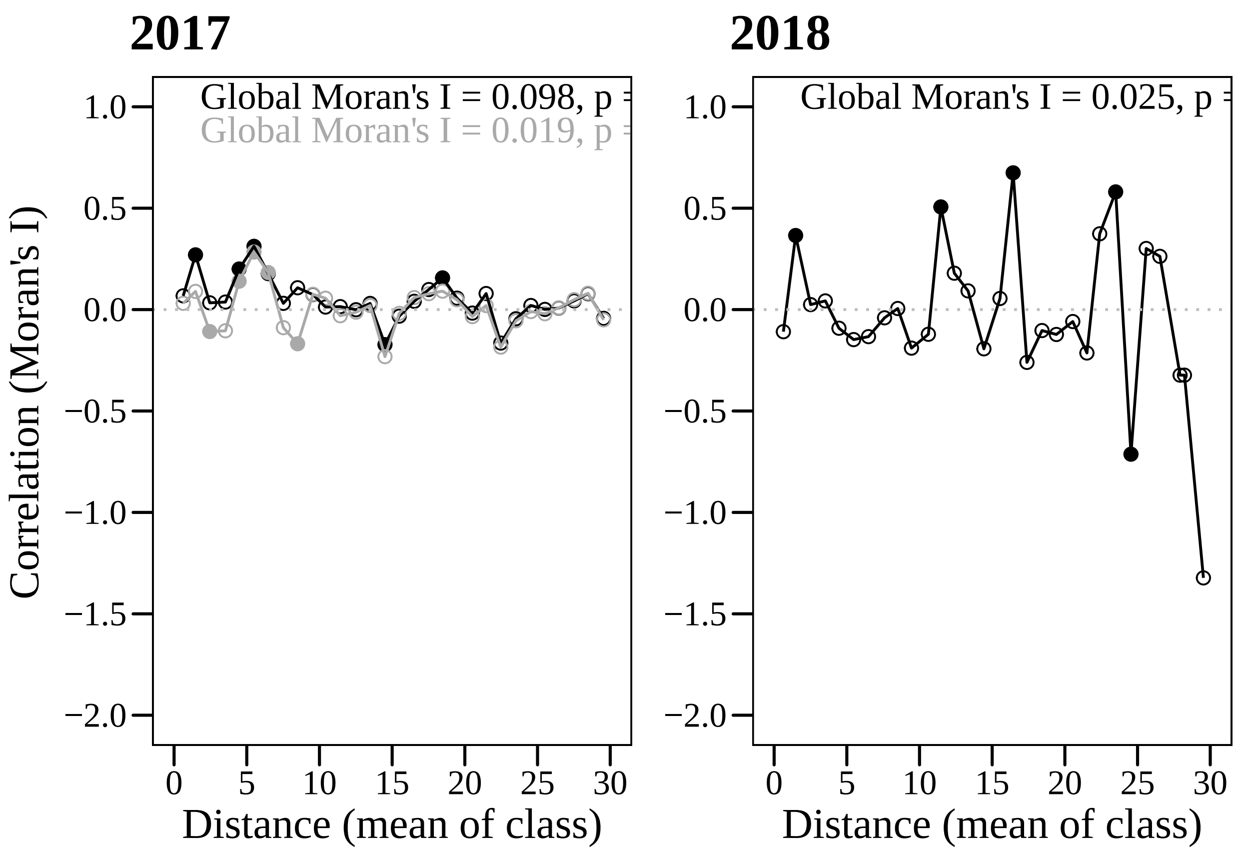
```
##
##  Monte-Carlo simulation of Moran I
##
## data:  res_selection_2017_ME
## weights: data_plants.listw_2017
## number of simulations + 1: 1000
##
## statistic = 0.019028, observed rank = 883, p-value = 0.117
## alternative hypothesis: greater
```

App S6 - Figure S3

```r
# selection_2017
corr_selection_2017<-data.frame(cbind(distance=
                                as.vector(correlog_selection_2017$mean.of.class[1:31]),
                      correlation=as.vector(correlog_selection_2017$correlation[1:31]),
                      p=as.vector(correlog_selection_2017$p[1:31])))
corr_selection_2017_ME<-data.frame(cbind(distance=
                                   as.vector(
                                     correlog_selection_2017_ME$mean.of.class[1:31]),
                                 correlation=
                                   as.vector(correlog_selection_2017_ME$correlation[1:31]),
                                  p=as.vector(correlog_selection_2017_ME$p[1:31])))
corr_selection_2017$type<-"selection_2017"
corr_selection_2017_ME$type<-"selection_2017_ME"
corr_selection_2017<-rbind(corr_selection_2017,corr_selection_2017_ME)
corr_selection_2017$sig<-as.factor(ifelse(corr_selection_2017$p<0.05,1,0))

# selection_2018
corr_selection_2018<-data.frame(cbind(distance=
                                as.vector(correlog_selection_2018$mean.of.class[1:31]),
                      correlation=as.vector(correlog_selection_2018$correlation[1:31]),
                      p=as.vector(correlog_selection_2018$p[1:31])))
corr_selection_2018$sig<-as.factor(ifelse(corr_selection_2018$p<0.05,1,0))
```

```r
AppS6_FigS3<-grid.arrange(
  ggplot(corr_selection_2017,aes(x=distance, y=correlation)) +
    geom_point(aes(colour=type,shape=sig),size=2) +
    geom_line(aes(colour=type)) + ylab(NULL)+
    scale_x_continuous("Distance (mean of class)",limits=c(0,30),
                       breaks=c(0,5,10,15,20,25,30)) +
    scale_y_continuous(limits=c(-2,1),
                       breaks = seq(-2,1,0.5))+
    scale_shape_manual(values=c(1,19))+
    scale_color_manual(values=c("black","darkgrey"))+
    geom_hline(aes(yintercept=0), colour="grey",linetype=3)+
    my_theme()+ggtitle("2017")+
    annotation_custom(grobTree(textGrob("Global Moran's I = 0.098, p = 0.001",
                                        x=0.1,y=0.97,hjust=0,
                                        gp=gpar(col="black",fontsize=14,
                                                fontfamily="serif"))))+
    annotation_custom(grobTree(textGrob("Global Moran's I = 0.019, p = 0.124",
                                        x=0.1,y=0.92,hjust=0,
                                        gp=gpar(col="darkgrey",fontsize=14,
                                                fontfamily="serif")))),
  ggplot(corr_selection_2018,aes(x=distance, y=correlation)) +
    geom_point(aes(shape=sig),size=2,color="black") +
    geom_line(color="black") + ylab(NULL)+
    scale_x_continuous("Distance (mean of class)",limits=c(0,30),
                       breaks=c(0,5,10,15,20,25,30)) +
    scale_y_continuous(limits=c(-2,1),
                       breaks = seq(-2,1,0.5))+
    scale_shape_manual(values=c(1,19))+
    geom_hline(aes(yintercept=0), colour="grey",linetype=3)+
    my_theme()+ggtitle("2018")+
```

```
  annotation_custom(grobTree(textGrob("Global Moran's I = 0.025, p = 0.228",
                                    x=0.1,y=0.97,hjust=0,
                                    gp=gpar(col="black",fontsize=14,
                                            fontfamily="serif")))),
 ncol=2,left=textGrob("Correlation (Moran's I)",just="center",
                     hjust=0.42,
                     gp=gpar(fontsize=16,fontfamily="serif"),
                     rot = 90))
```




```
ggsave(filename="output/figures/AppS6_FigS3.tiff",
       plot=AppS6_FigS3,device="tiff",width=28,height=12,units="cm",dpi=300,
       compression="lzw")
```

Effect of temperature on the relationship absolute fitness-FFD

Spatial correlograms

```
res_selectionabs_2017<-residuals(selectionabs_2017_2)
res_selectionabs_2018<-residuals(selectionabs_2018_2)
```

```
correlog_selectionabs_2017 <- correlog(x=subset(data_plants,year==2017)$x,
                                       y=subset(data_plants,year==2017)$y,
```

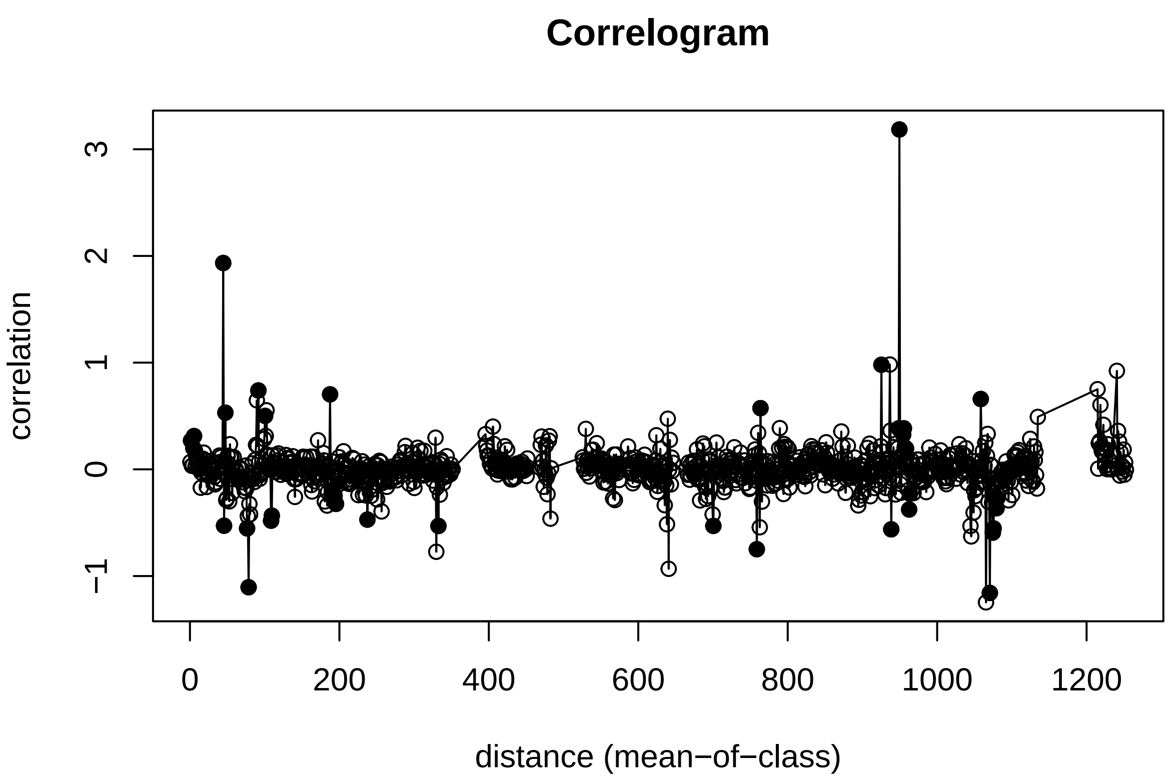
```r
                                      res_selectionabs_2017,increment=1, resamp=100,
                                      quiet=F)
```

```
## 10  of  100 20  of  100 30  of  100 40  of  100 50  of  100 60  of  100 70  of  100 80  of  100 90
```

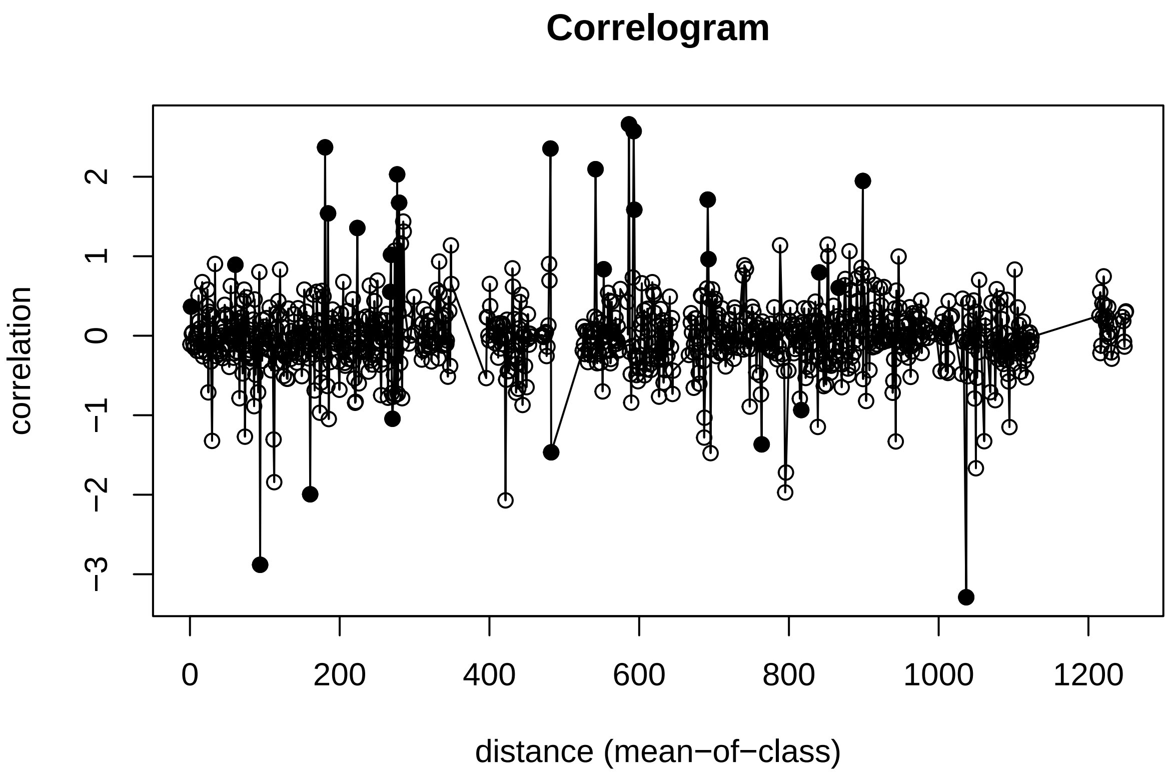
```r
correlog_selectionabs_2018 <- correlog(x=subset(data_plants,year==2018)$x,
                                      y=subset(data_plants,year==2018)$y,
                                      res_selectionabs_2018,increment=1, resamp=100,quiet=F)
```

```
## 10  of  100 20  of  100 30  of  100 40  of  100 50  of  100 60  of  100 70  of  100 80  of  100 90
```

```r
plot(correlog_selectionabs_2017)
```




```r
plot(correlog_selectionabs_2018)
```



Moran's I

Calculation of global Moran's I with a permutation test (1000 random permutations), based on a connectivity matrix of pairwise Euclidean distances among the plants up to a distance of 30 m.

```
moran_selectionabs_2017<- moran.mc(res_selectionabs_2017,
                               listw=data_plants.listw_2017,nsim=999)
moran_selectionabs_2018<- moran.mc(res_selectionabs_2018,
                               listw=data_plants.listw_2018,nsim=999)
```

```
moran_selectionabs_2017 # Significant autocorrelation in the residuals
```

```
##
##  Monte-Carlo simulation of Moran I
##
## data:  res_selectionabs_2017
## weights: data_plants.listw_2017
## number of simulations + 1: 1000
##
## statistic = 0.097925, observed rank = 1000, p-value = 0.001
## alternative hypothesis: greater
```

```
moran_selectionabs_2018 # No significant autocorrelation in the residuals
```

```
##
##  Monte-Carlo simulation of Moran I
##
## data:  res_selectionabs_2018
## weights: data_plants.listw_2018
## number of simulations + 1: 1000
##
## statistic = 0.025308, observed rank = 798, p-value = 0.202
## alternative hypothesis: greater
```

Moran's eigenvector mapping

```
ME.selectionabs_2017 <-ME(selectionabs_2017_2,listw=data_plants.listw_2017,
               data=subset(data_plants,year==2017),
               alpha=0.05,verbose=T)
```

```
## eV[,3], I: 0.0498509 ZI: NA, pr(ZI): 0.01
## eV[,11], I: 0.01902798 ZI: NA, pr(ZI): 0.1
```

```
vector1_2017_selectionabs<-ME.selectionabs_2017$vectors[,1]
vector2_2017_selectionabs<-ME.selectionabs_2017$vectors[,2]

selectionabs_2017_ME<-lm(nseed~ffd*temp+nfl+
                        vector1_2017_selectionabs+vector2_2017_selectionabs,
                     subset(data_plants,year==2017))
summ(selectionabs_2017_ME)
```

Observations	245
Dependent variable	nseed
Type	OLS linear regression

F(6,238)	78.796
R^2	0.665
Adj. R^2	0.657

	Est.	S.E.	t val.	p
(Intercept)	2075.105	1998.064	1.039	0.300
ffd	-9.859	11.230	-0.878	0.381
temp	-137.335	99.447	-1.381	0.169
nfl	28.109	1.461	19.241	0.000
vector1_2017_selectionabs	-1924.663	542.057	-3.551	0.000
vector2_2017_selectionabs	-965.923	563.339	-1.715	0.088
ffd:temp	0.683	0.564	1.210	0.227

Standard errors: OLS

BCA intervals Used for assessing significance.

```
# ffd
slp <- function(selectionabs_2017_ME) coef(selectionabs_2017_ME)[2]
b <- car::Boot(selectionabs_2017_ME,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
ffd_ci_17 <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# temp
slp <- function(selectionabs_2017_ME) coef(selectionabs_2017_ME)[3]
b <- car::Boot(selectionabs_2017_ME,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
temp_ci_17 <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# nfl
slp <- function(selectionabs_2017_ME) coef(selectionabs_2017_ME)[4]
b <- car::Boot(selectionabs_2017_ME,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
nfl_ci_17 <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# vector1
slp <- function(selectionabs_2017_ME) coef(selectionabs_2017_ME)[5]
b <- car::Boot(selectionabs_2017_ME,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
vector1_ci_17 <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# vector2
slp <- function(selectionabs_2017_ME) coef(selectionabs_2017_ME)[6]
b <- car::Boot(selectionabs_2017_ME,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
vector2_ci_17 <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```
# ffd:temp
slp <- function(selectionabs_2017_ME) coef(selectionabs_2017_ME)[7]
b <- car::Boot(selectionabs_2017_ME,slp, R=10000) # note the capital B
b1 <- boot::boot.ci(b,type="bca")
ffd_temp_ci_17 <- as.data.frame(b1$bca[1,4:5])
rm(slp, b, b1)
```

```r
# Save confidence intervals as a table
BCIs_selectionabs_2017_ME <- cbind(
  rbind(ffd_ci_17[1,] ,temp_ci_17[1,], nfl_ci_17[1,],vector1_ci_17[1,],
        vector2_ci_17[1,],ffd_temp_ci_17[1,]),
  rbind(ffd_ci_17[2,] ,temp_ci_17[2,], nfl_ci_17[2,],vector1_ci_17[2,],
        vector2_ci_17[2,],ffd_temp_ci_17[2,])
)
colnames(BCIs_selectionabs_2017_ME)<-c("lower","upper")
rownames(BCIs_selectionabs_2017_ME) <- c("ffd","temp","nfl",
                                          "vector1","vector2","ffd:temp")
save(BCIs_selectionabs_2017_ME,file="output/BCIs_selectionabs_2017_ME.RData")
```

```r
BCIs_selectionabs_2017_ME
```

2017

```
##                  lower        upper
## ffd        -24.9035001    13.791561
## temp      -286.6793309    29.314151
## nfl         20.7376721    34.045376
## vector1  -3210.7847577 -1060.425022
## vector2  -2256.1513349   213.254205
## ffd:temp    -0.3372933     1.471582
```

Tests of residual spatial autocorrelation

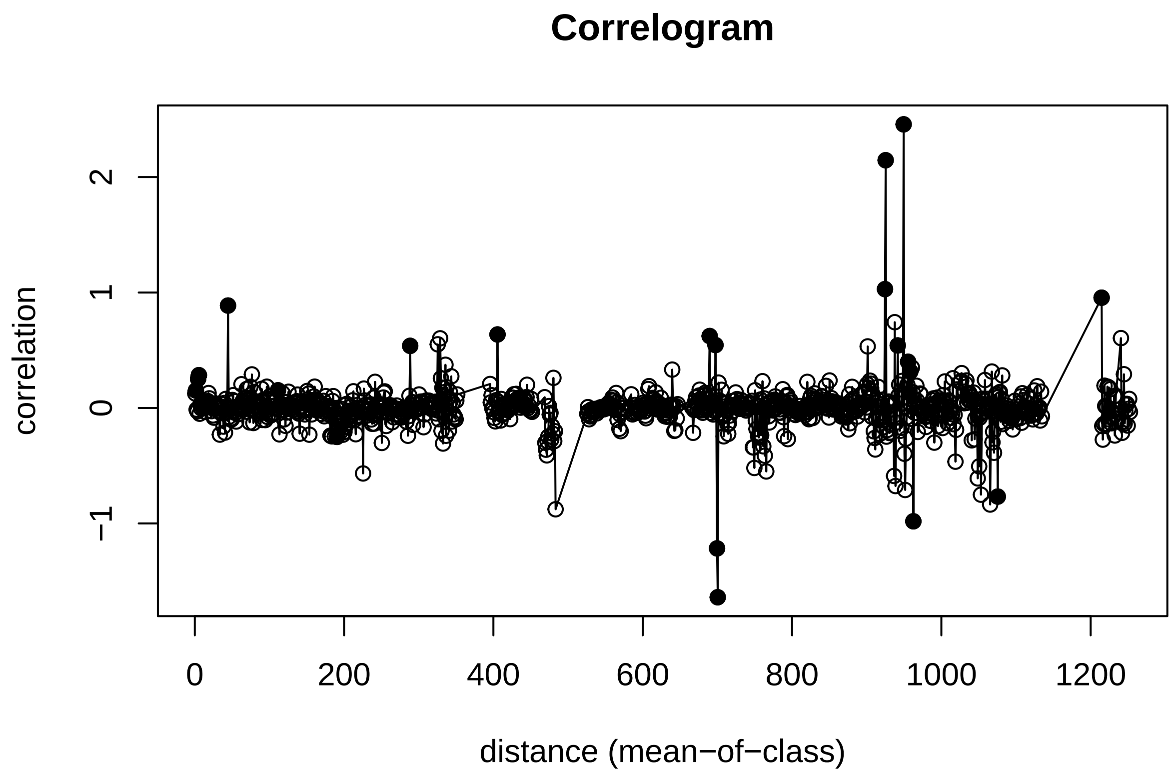
```r
res_selectionabs_2017_ME<-residuals(selectionabs_2017_ME)
```

```r
correlog_selectionabs_2017_ME <- correlog(x=subset(data_plants,year==2017)$x,
                            y=subset(data_plants,year==2017)$y,
                            res_selectionabs_2017_ME,increment=1, resamp=100,quiet=F)
```

Spatial correlograms

```
## 10  of  100 20  of  100 30  of  100 40  of  100 50  of  100 60  of  100 70  of  100 80  of  100 90
```

```r
plot(correlog_selectionabs_2017_ME)
```



Moran's I Calculation of global Moran's I with a permutation test (1000 random permutations), based on a connectivity matrix of pairwise Euclidean distances among the plants up to a distance of 30 m.

```
moran_selectionabs_2017_ME<- moran.mc(res_selectionabs_2017_ME,
                                      listw=data_plants.listw_2017,nsim=999)
```

```
moran_selectionabs_2017_ME # Still significant autocorrelation in the residuals
```

```
##
##  Monte-Carlo simulation of Moran I
##
## data:  res_selectionabs_2017_ME
## weights: data_plants.listw_2017
## number of simulations + 1: 1000
##
## statistic = 0.065462, observed rank = 996, p-value = 0.004
## alternative hypothesis: greater
```

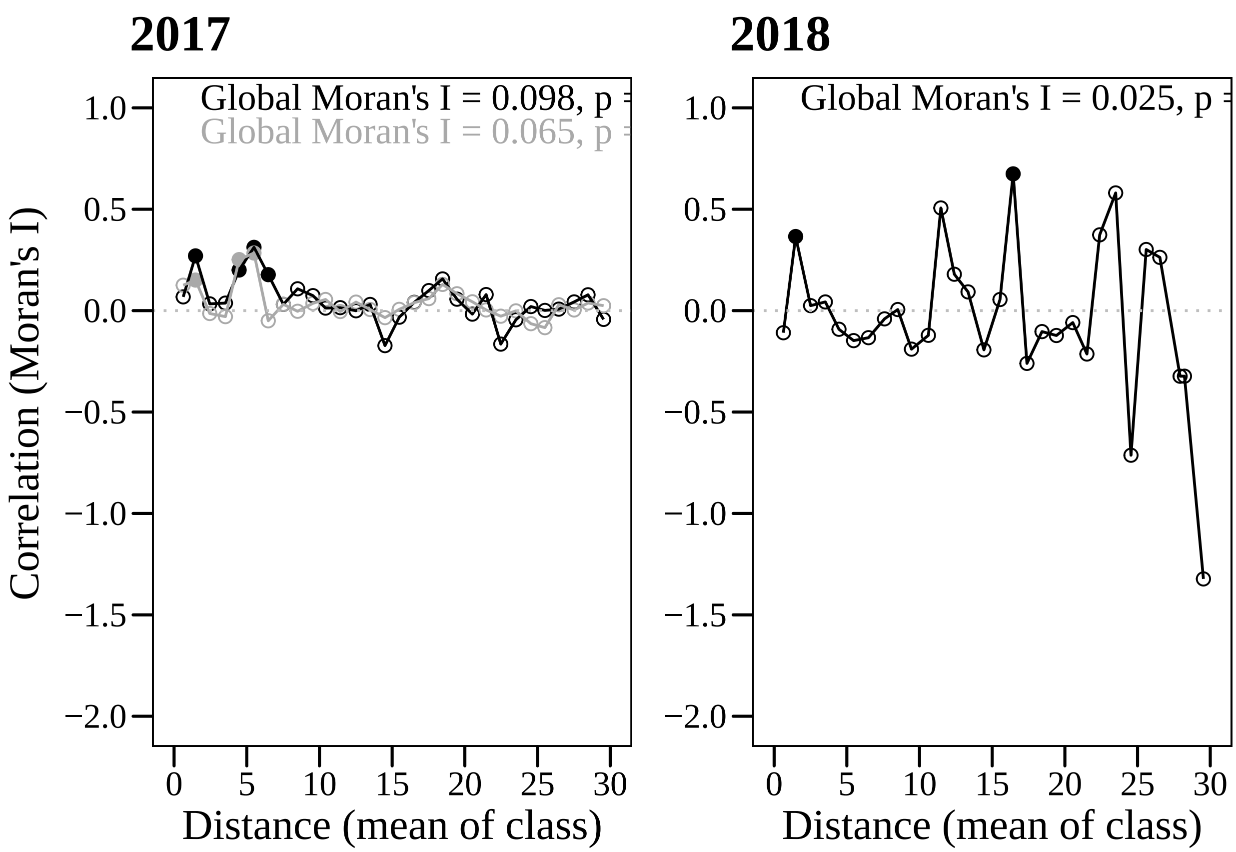
AppS6 - Figure S4

```
# selectionabs_2017
corr_selectionabs_2017<-data.frame(cbind(distance=
```

```
                                as.vector(correlog_selectionabs_2017$mean.of.class[1:31]),
                      correlation=as.vector(correlog_selectionabs_2017$correlation[1:31]),
                      p=as.vector(correlog_selectionabs_2017$p[1:31])))
corr_selectionabs_2017_ME<-data.frame(cbind(distance=
                                   as.vector(
                                     correlog_selectionabs_2017_ME$mean.of.class[1:31]),
                                 correlation=
                                   as.vector(correlog_selectionabs_2017_ME$correlation[1:31]),
                                  p=as.vector(correlog_selectionabs_2017_ME$p[1:31])))
corr_selectionabs_2017$type<-"selectionabs_2017"
corr_selectionabs_2017_ME$type<-"selectionabs_2017_ME"
corr_selectionabs_2017<-rbind(corr_selectionabs_2017,corr_selectionabs_2017_ME)
corr_selectionabs_2017$sig<-as.factor(ifelse(corr_selectionabs_2017$p<0.05,1,0))

# selectionabs_2018
corr_selectionabs_2018<-data.frame(cbind(distance=
                                as.vector(correlog_selectionabs_2018$mean.of.class[1:31]),
                      correlation=as.vector(correlog_selectionabs_2018$correlation[1:31]),
                      p=as.vector(correlog_selectionabs_2018$p[1:31])))
corr_selectionabs_2018$sig<-as.factor(ifelse(corr_selectionabs_2018$p<0.05,1,0))
```

```
AppS6_FigS4<-grid.arrange(
  ggplot(corr_selectionabs_2017,aes(x=distance, y=correlation)) +
    geom_point(aes(colour=type,shape=sig),size=2) +
    geom_line(aes(colour=type)) + ylab(NULL)+
    scale_x_continuous("Distance (mean of class)",limits=c(0,30),
                       breaks=c(0,5,10,15,20,25,30)) +
    scale_y_continuous(limits=c(-2,1),
                       breaks = seq(-2,1,0.5))+
    scale_shape_manual(values=c(1,19))+
    scale_color_manual(values=c("black","darkgrey"))+
    geom_hline(aes(yintercept=0), colour="grey",linetype=3)+
    my_theme()+ggtitle("2017")+
    annotation_custom(grobTree(textGrob("Global Moran's I = 0.098, p = 0.001",
                                        x=0.1,y=0.97,hjust=0,
                                        gp=gpar(col="black",fontsize=14,
                                                fontfamily="serif"))))+
    annotation_custom(grobTree(textGrob("Global Moran's I = 0.065, p = 0.001",
                                        x=0.1,y=0.92,hjust=0,
                                        gp=gpar(col="darkgrey",fontsize=14,
                                                fontfamily="serif")))),
  ggplot(corr_selectionabs_2018,aes(x=distance, y=correlation)) +
    geom_point(aes(shape=sig),size=2,color="black") +
    geom_line(color="black") + ylab(NULL)+
    scale_x_continuous("Distance (mean of class)",limits=c(0,30),
                       breaks=c(0,5,10,15,20,25,30)) +
    scale_y_continuous(limits=c(-2,1),
                       breaks = seq(-2,1,0.5))+
    scale_shape_manual(values=c(1,19))+
    geom_hline(aes(yintercept=0), colour="grey",linetype=3)+
    my_theme()+ggtitle("2018")+
    annotation_custom(grobTree(textGrob("Global Moran's I = 0.025, p = 0.228",
                                        x=0.1,y=0.97,hjust=0,
```

```
                                              gp=gpar(col="black",fontsize=14,
                                                      fontfamily="serif")))),
  ncol=2,left=textGrob("Correlation (Moran's I)",just="center",
                       hjust=0.42,
                       gp=gpar(fontsize=16,fontfamily="serif"),
                       rot = 90))
```




```
ggsave(filename="output/figures/AppS6_FigS4.tiff",
       plot=AppS6_FigS4,device="tiff",width=28,height=12,units="cm",dpi=300,
       compression="lzw")
```

R Session Info

```
sessionInfo()
```

```
## R version 4.1.2 (2021-11-01)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19043)
##
## Matrix products: default
##
## locale:
```

```
## [1] LC_COLLATE=English_Sweden.1252  LC_CTYPE=English_Sweden.1252
## [3] LC_MONETARY=English_Sweden.1252 LC_NUMERIC=C
## [5] LC_TIME=English_Sweden.1252
##
## attached base packages:
## [1] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] spatialreg_1.2-1   Matrix_1.4-0       spdep_1.2-1        sf_1.0-5
##  [5] spData_2.0.1       ncf_1.2-9          sp_1.4-6           foreign_0.8-82
##  [9] ggrepel_0.9.1      viridis_0.6.2      viridisLite_0.4.0  car_3.0-12
## [13] lmtest_0.9-39      zoo_1.8-9          ggforce_0.3.3      lubridate_1.8.0
## [17] effects_4.2-1      carData_3.0-5      segmented_1.3-4    MASS_7.3-54
## [21] MuMIn_1.43.17      ggeffects_1.1.1    kableExtra_1.3.4   jtools_2.1.4
## [25] ggpubr_0.4.0       broom_0.7.11       RColorBrewer_1.1-2 DHARMa_0.4.5
## [29] gridExtra_2.3      knitr_1.37         ggthemes_4.2.4     forcats_0.5.1
## [33] stringr_1.4.0      dplyr_1.0.6        purrr_0.3.4        readr_2.1.1
## [37] tidyr_1.1.4        tibble_3.1.2       ggplot2_3.3.5      tidyverse_1.3.1
##
## loaded via a namespace (and not attached):
##  [1] readxl_1.3.1       backports_1.4.1    systemfonts_1.0.3  splines_4.1.2
##  [5] digest_0.6.27      htmltools_0.5.2    gdata_2.18.0       fansi_0.4.2
##  [9] magrittr_2.0.1     tzdb_0.2.0         modelr_0.1.8       gmodels_2.18.1
## [13] vroom_1.5.7        svglite_2.0.0      colorspace_2.0-1   rvest_1.0.2
## [17] mitools_2.4        haven_2.4.3        xfun_0.29          rgdal_1.5-28
## [21] crayon_1.4.2       jsonlite_1.7.3     lme4_1.1-27.1      survival_3.2-13
## [25] glue_1.4.2         polyclip_1.10-0    gtable_0.3.0       webshot_0.5.2
## [29] abind_1.4-5        scales_1.1.1       DBI_1.1.2          rstatix_0.7.0
## [33] Rcpp_1.0.8         units_0.7-2        bit_4.0.4          proxy_0.4-26
## [37] stats4_4.1.2       survey_4.1-1       httr_1.4.2         wk_0.6.0
## [41] ellipsis_0.3.2     pkgconfig_2.0.3    farver_2.1.0       nnet_7.3-17
## [45] dbplyr_2.1.1       deldir_1.0-6       utf8_1.2.1         labeling_0.4.2
## [49] tidyselect_1.1.1   rlang_0.4.10       munsell_0.5.0      cellranger_1.1.0
## [53] tools_4.1.2        cli_3.1.1          generics_0.1.1     sjlabelled_1.1.8
## [57] evaluate_0.14      fastmap_1.1.0      yaml_2.2.2         bit64_4.0.5
## [61] fs_1.5.2           pander_0.6.4       s2_1.0.7           nlme_3.1-155
## [65] xml2_1.3.3         compiler_4.1.2     rstudioapi_0.13    e1071_1.7-9
## [69] ggsignif_0.6.3     reprex_2.0.1       tweenr_1.0.2       stringi_1.7.6
## [73] highr_0.9          lattice_0.20-45    classInt_0.4-3     nloptr_1.2.2.3
## [77] vctrs_0.3.8        pillar_1.6.1       LearnBayes_2.15.1  lifecycle_1.0.1
## [81] cowplot_1.1.1      insight_0.15.0     raster_3.4-10      R6_2.5.1
## [85] KernSmooth_2.23-20 codetools_0.2-18   boot_1.3-28        gtools_3.9.2
## [89] assertthat_0.2.1   withr_2.4.3        mgcv_1.8-38        expm_0.999-6
## [93] parallel_4.1.2     hms_1.1.1          coda_0.19-4        class_7.3-20
## [97] minqa_1.2.4        rmarkdown_2.11
```