

Response of *Pinguicula vulgaris* to geothermal heating

Analyses

Alicia Valdés

Contents

1. Models for FFD	3
Model prediction	3
2. Models for absolute fitness	5
Model selection	5
Model diagnostics for best model	6
BCa intervals for averaged model	7
Model predictions from best model	7
3. Models for relative fitness (selection models)	13
Model selection	14
Model diagnostics for best model	15
Model predictions from best model	16
4. Models for binomial fitness (maybe not included in paper)	23
Model selection	23
Model diagnostics for best model	25
Model predictions from best model	25
5. Aster models	26
Test interactions	32
Test main effects without interactions	38
Table LRTs	40
Selection gradients	41
Reduced model without F	42
Test some interactions	43
Reduced model with terms included in averaged LMs (USE in main text)	44
Test interactions	45
Test main effects without interactions	46

Predictions	49
seeds:temp_c:P:	49
seeds:temp_c:FFD_c:	51
Figures 2-3	62
Figure 2	62
Without points	62
With points	63
Figure 3	65
Session info	66
Define two variables: P (1 if pollinated, 0 if not) and F (1 if fed, 0 if not).	
<pre>ping_20_data <- ping_20_data %>% mutate(P=as.factor(ifelse(treatment=="P" treatment=="PF",1,0)), F=as.factor(ifelse(treatment=="F" treatment=="PF",1,0)))</pre>	
<pre>nrow(subset(ping_20_data,P==0))/nrow(ping_20_data)</pre>	
## [1] 0.5156794	
<pre>nrow(subset(ping_20_data,F==0))/nrow(ping_20_data)</pre>	
## [1] 0.5121951	
Approximately half of the plants are pollinated and half are fed.	
Proportion of plants that produced 0 seeds:	
<pre>nrow(subset(ping_20_data,n_seeds==0))/nrow(ping_20_data)</pre>	
## [1] 0.5052265	
Around half of the plants produced no seeds.	
Create 0/1 variable for fitness:	
<pre>ping_20_data\$fitness_01<-with(ping_20_data,ifelse(n_seeds==0,0, ifelse(n_seeds>0,1,NA)))</pre>	
One case when n_seedpods>0 and n_seeds=0: convert to NA:	
<pre>subset(ping_20_data,n_seedpods>0&n_seeds==0)</pre>	

```
## # A tibble: 1 x 24
##   id      FFD_corr plot      x      y peak comments start_treatment treatment
##   <chr>    <dbl> <chr> <dbl> <dbl> <dbl> <chr>    <date>          <chr>
## 1 P-450    181. HC10    3.8  1.25    1 <NA>    2020-07-01      PF
##   temp date_peak    dm1    dm2 n_stems n_seedpods n_seeds
##   <dbl> <date>    <dbl> <dbl>   <dbl>    <dbl>   <dbl>
## 1  11.6 2020-07-01    2.6    2.3     1        1        0
##   comments_peak      comments_seeds max_h med_h ros_area P
##   <chr>              <chr>          <dbl> <dbl>   <dbl> <fct>
## 1 pod had opened before collectioni <NA>          4.4  4.4    18.8 1
##   F      fitness_01
##   <fct>    <dbl>
## 1 1      0
```

```
ping_20_data$n_seeds<-with(ping_20_data,ifelse(id=="P-450",NA,n_seeds))
```

1. Models for FFD

Effect of temperature on FFD

Using linear models, including quadratic effects of temp.

```
FFD_2020_1<-lm(FFD_corr~temp+I(temp^2),ping_20_data)
summary(FFD_2020_1)
```

```
##
## Call:
## lm(formula = FFD_corr ~ temp + I(temp^2), data = ping_20_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.3407  -2.5589   0.2415   2.8193  14.0155
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  216.81764    3.78519   57.281 < 2e-16 ***
## temp         -3.37575    0.51862   -6.509 3.41e-10 ***
## I(temp^2)     0.03967    0.01681    2.360  0.0189 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.435 on 284 degrees of freedom
## Multiple R-squared:  0.7598, Adjusted R-squared:  0.7581
## F-statistic: 449.2 on 2 and 284 DF, p-value: < 2.2e-16
```

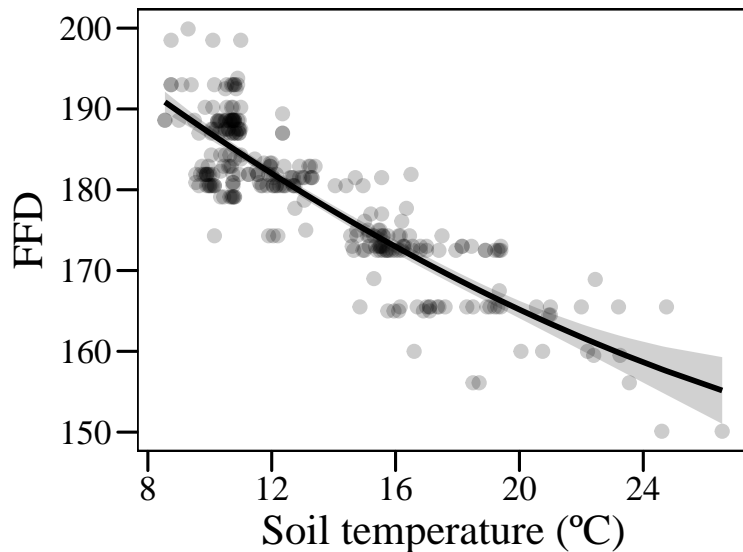
Model diagnostics (not shown) indicated that linear models are OK.

Model prediction

```

predict_FFD<-ggpredict(FFD_2020_1,terms="temp")
(fig1<-ggplot(ping_20_data,aes(x=temp,y=FFD_corr))+
  geom_ribbon(data=predict_FFD,
            aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),
            fill="grey",alpha=0.7)+
  geom_line(data=predict_FFD,aes(x=x,y=predicted),size=1,color="black")+
  geom_point(size=2,alpha=0.2)+
  xlab("Soil temperature (°C)")+ylab("FFD")+
  my_theme())

```



```

ggsave(filename="output/figures/fig1.tiff",device="tiff",
        width=9,height=8,units="cm",dpi=300,compression="lzw")

```

The effect looks almost linear.

Predictions of FFD_corr for minimum and maximum temperatures:

```
ggpredict(FFD_2020_1,terms="temp[minmax]")
```

```

## # Predicted values of FFD_corr
##
##  temp | Predicted |          95% CI
## -----
##  8.55 |    190.85 | [189.57, 192.14]
## 26.55 |    155.16 | [151.03, 159.28]

```

```
# 190.85-155.16=35.69 days earlier on warmer soils
```

2. Models for absolute fitness

Model selection

```
ping_20_data$ros_area_log<-log(ping_20_data$ros_area)
subset3<-subset(ping_20_data,!is.na(n_seeds)&!is.na(ros_area_log))
globmod_fitness_2020<-lm(n_seeds~temp*FFD_corr*P*F+ros_area_log,
                        subset3,na.action="na.fail")
clusterType <- if(length(find.package("snow", quiet = TRUE))) "SOCK" else "PSOCK"
clust1 <- try(makeCluster(getOption("cl.cores", 3), type = clusterType))
clusterExport(clust1, "subset3")
clusterEvalQ(clust1, library(stats))
```

```
## [[1]]
## [1] "stats"      "graphics"  "grDevices" "utils"      "datasets"  "methods"
## [7] "base"
##
## [[2]]
## [1] "stats"      "graphics"  "grDevices" "utils"      "datasets"  "methods"
## [7] "base"
##
## [[3]]
## [1] "stats"      "graphics"  "grDevices" "utils"      "datasets"  "methods"
## [7] "base"
```

```
modsel_fitness_2020<-pdredge(globmod_fitness_2020,fixed=c("ros_area_log"),
                             cluster=clust1)

summary(model.avg(modsel_fitness_2020,
                  subset=delta<2)) # Summary averaged model
```

```
##
## Call:
## model.avg(object = modsel_fitness_2020, subset = delta < 2)
##
## Component model call:
## lm(formula = n_seeds ~ <2 unique rhs>, data = subset3, na.action =
##     na.fail)
##
## Component models:
##           df logLik   AICc delta weight
## 23456789  10 -1710.3 3441.44  0.00   0.57
## 123456789 11 -1709.5 3442.01  0.57   0.43
##
## Term codes:
##           F           FFD_corr           P       ros_area_log           temp
##           1             2             3             4             5
##      FFD_corr:P      FFD_corr:temp      P:temp FFD_corr:P:temp
##           6             7             8             9
##
## Model-averaged coefficients:
```

```
## (full average)
##               Estimate Std. Error Adjusted SE z value Pr(>|z|)
## (Intercept)    2548.0586    819.9343    823.7147   3.093  0.00198 **
## FFD_corr       -13.2362     4.4415     4.4620   2.966  0.00301 **
## P1             1626.7779   1399.9698   1406.4276   1.157  0.24741
## temp          -116.9397    47.1370    47.3541   2.469  0.01353 *
## ros_area_log    14.9968    20.1601    20.2528   0.740  0.45901
## FFD_corr:P1     -11.4347     7.6640     7.6993   1.485  0.13750
## FFD_corr:temp    0.6085     0.2754     0.2767   2.199  0.02785 *
## P1:temp        -229.5088    88.6467    89.0556   2.577  0.00996 **
## FFD_corr:P1:temp  1.4747     0.5128     0.5152   2.862  0.00421 **
## F1             -8.4873    14.2828    14.3179   0.593  0.55333
##
## (conditional average)
##               Estimate Std. Error Adjusted SE z value Pr(>|z|)
## (Intercept)    2548.0586    819.9343    823.7147   3.093  0.00198 **
## FFD_corr       -13.2362     4.4415     4.4620   2.966  0.00301 **
## P1             1626.7779   1399.9698   1406.4276   1.157  0.24741
## temp          -116.9397    47.1370    47.3541   2.469  0.01353 *
## ros_area_log    14.9968    20.1601    20.2528   0.740  0.45901
## FFD_corr:P1     -11.4347     7.6640     7.6993   1.485  0.13750
## FFD_corr:temp    0.6085     0.2754     0.2767   2.199  0.02785 *
## P1:temp        -229.5088    88.6467    89.0556   2.577  0.00996 **
## FFD_corr:P1:temp  1.4747     0.5128     0.5152   2.862  0.00421 **
## F1            -19.7647    15.8796    15.9530   1.239  0.21537
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
importance(modsel_fitness_2020) # Variable importance
```

```
##               ros_area_log temp  FFD_corr FFD_corr:temp P      P:temp
## Sum of weights:           1           1           1           1      0.97  0.96
## N containing models:    167          148    148           84          148    84
##               FFD_corr:P FFD_corr:P:temp F      F:P      F:FFD_corr F:temp
## Sum of weights:           0.95          0.91           0.68  0.24    0.2      0.2
## N containing models:     84           20          148    84    84          84
##               F:P:temp F:FFD_corr:P F:FFD_corr:temp F:FFD_corr:P:temp
## Sum of weights:           0.02          0.02           0.02          <0.01
## N containing models:     20           20           20           1
```

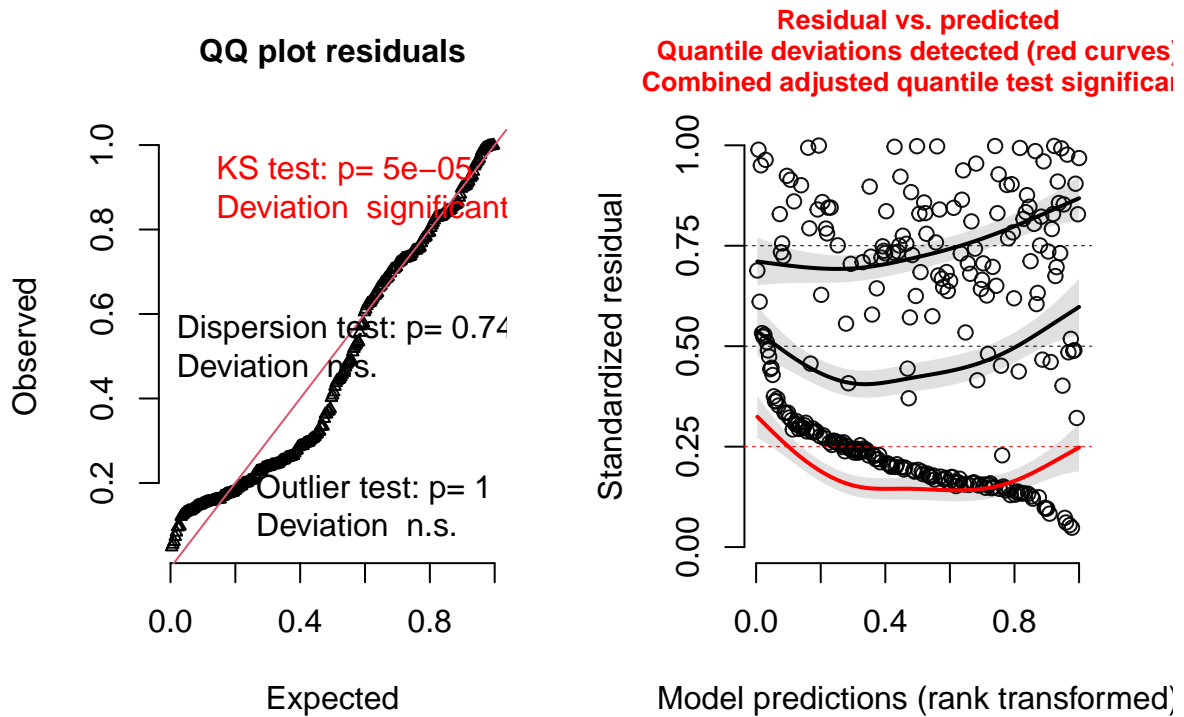
```
r.squaredGLMM(get.models(modsel_fitness_2020,subset=1)$"9102") #R square of best model
```

```
##               R2m      R2c
## [1,] 0.1221926 0.1221926
```

Model diagnostics for best model

```
plot(simulateResiduals(fittedModel=get.models(modsel_fitness_2020,
                                                subset=1)$"9102",n=5000))
```

DHARMA residual diagnostics



BCa intervals for averaged model

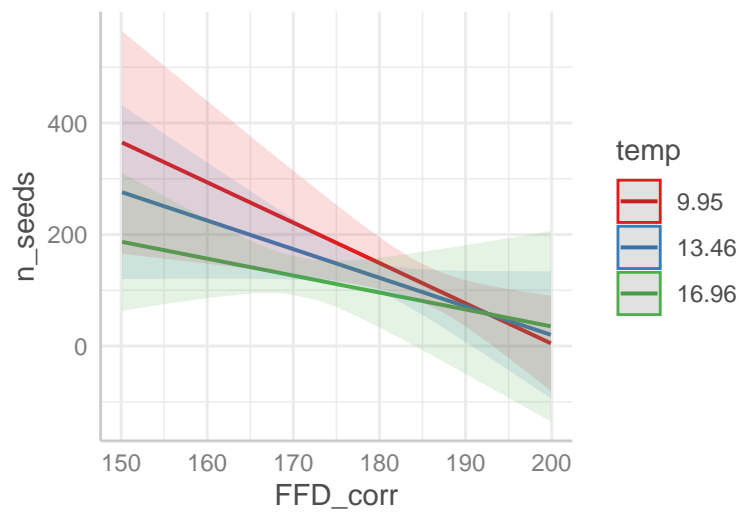
BCIs_av_fitness_2020

	lower	upper
## FFD_corr	-20.8240892	-6.437444
## P1	-1108.0783729	4426.193592
## temp	-195.3810415	-56.217322
## ros_area_log	-21.8377898	50.648481
## FFD_corr:P1	-26.5599454	3.623573
## FFD_corr:temp	0.2406201	1.074869
## P1:temp	-408.1403124	-41.821116
## FFD_corr:P1:temp	0.3976504	2.539033
## F1	-51.1850405	12.129912

Model predictions from best model

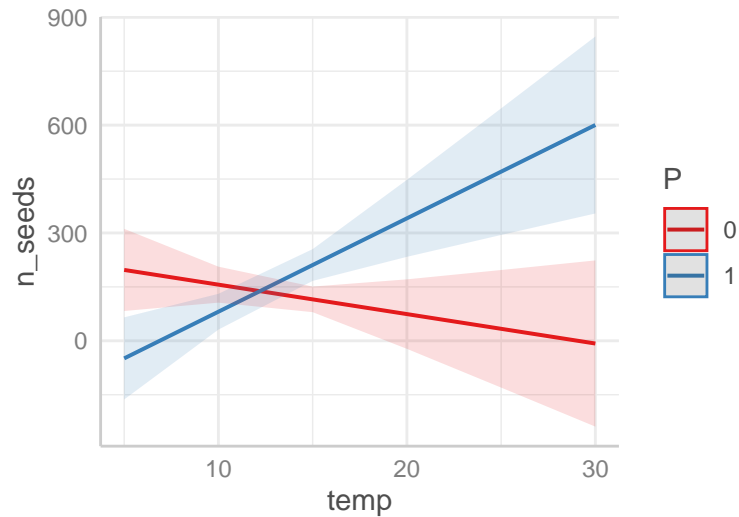
```
plot(ggpredict(get.models(modsel_fitness_2020, subset=1)$"9102",
  terms=c("FFD_corr[all]", "temp")))
```

Predicted values of n_seeds



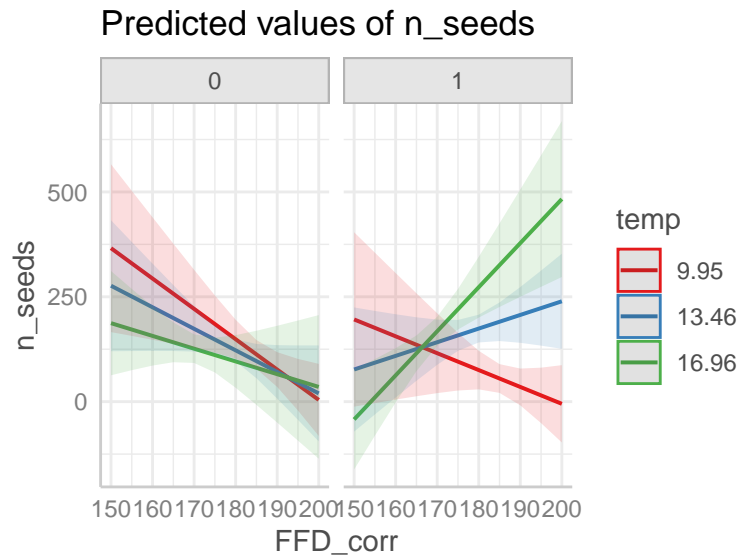
```
plot(ggpredict(get.models(modsel_fitness_2020,subset=1)$"9102",
  terms=c("temp","P")))
```

Predicted values of n_seeds



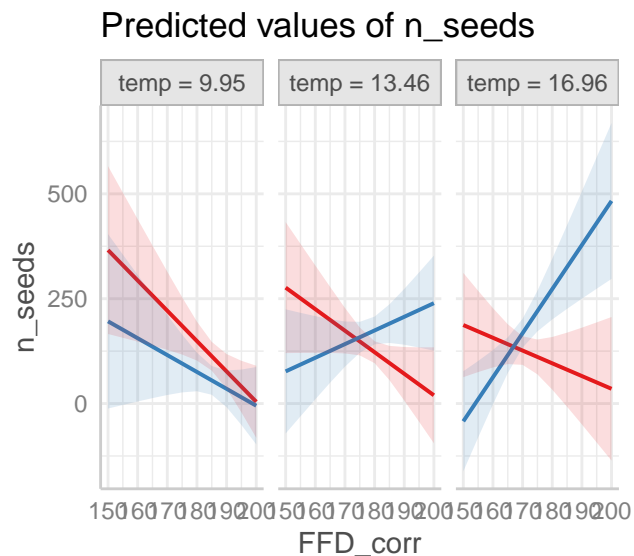
3-way interaction (two different representations):

```
plot(ggpredict(get.models(modsel_fitness_2020,subset=1)$"9102",
  terms=c("FFD_corr","temp","P")))
```

For non-pollinated plants, it is always better to flower early, irrespective of soil temperatures. For pollinated plants, it is better to flower early at cold soil temperatures, but late at medium and warm soil temperatures.

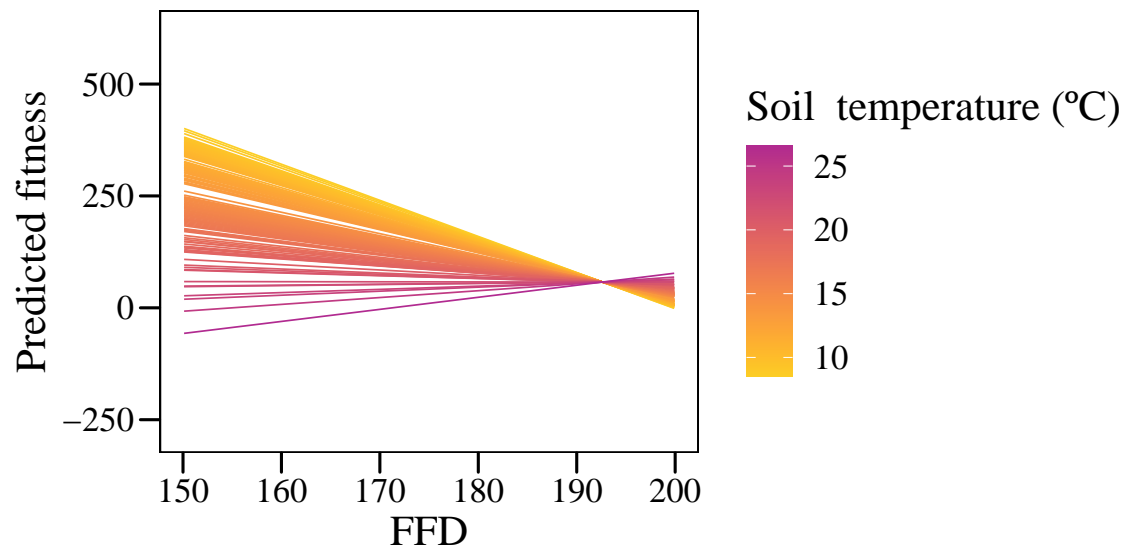
```
plot(ggpredict(get.models(modsel_fitness_2020,
  subset=1)$"9102",terms=c("FFD_corr","P","temp")))
```



At low temperatures (left panel above), it is always better to flower early, irrespective of if the plant was pollinated or not. At medium and high temperatures (center and right panel above), it is better to flower early for non-pollinated plants, but better to flower late for pollinated plants.

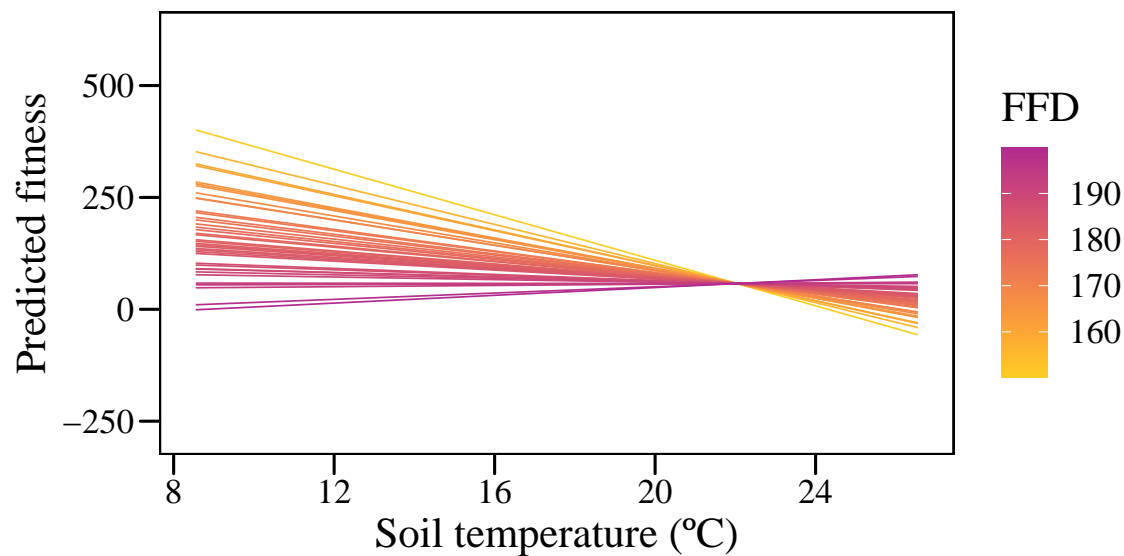
```
predict_fitness1<-ggpredict(get.models(modsel_fitness_2020,subset=1)$"9102",
  terms=c("FFD_corr[all]","temp[all]"))
ggplot(predict_fitness1,aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high,
  group=group))+
  geom_line(aes(color=as.numeric(as.character(group))),size=0.3)+
  xlab("FFD")+ylab("Predicted fitness")+
```

```
my_theme_legend()+
scale_colour_gradientn(colours = plasma(n=132,begin=0.4,end=0.9,direction=-1))+
labs(colour="Soil temperature (°C)")
```



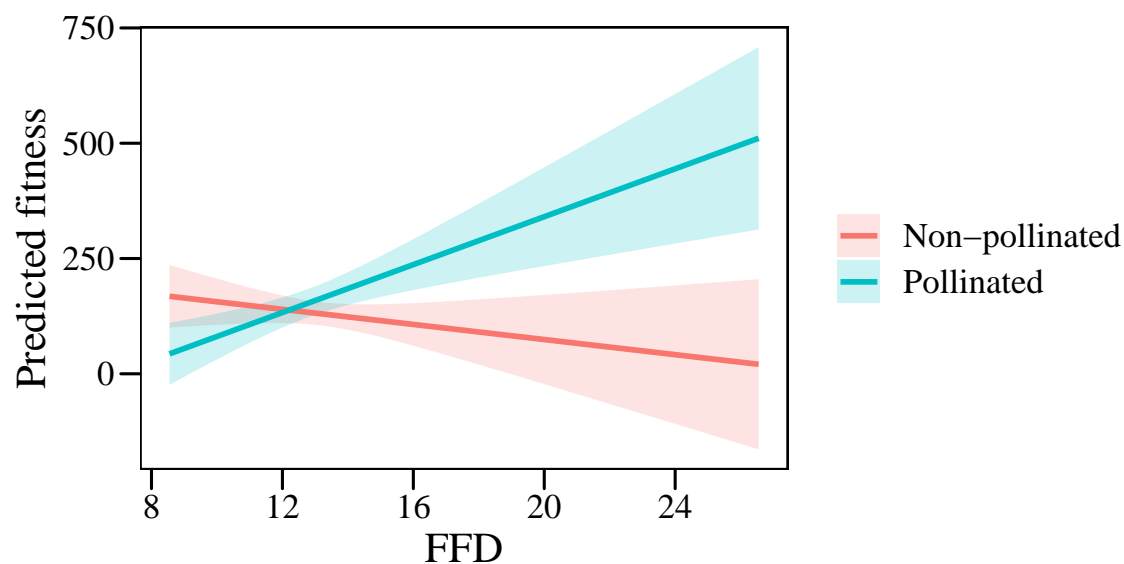
```
ggsave(filename="output/figures/fitness_1.tiff",device="tiff",
        width=15,height=8,units="cm",dpi=300,compression="lzw")

predict_fitness2<-ggpredict(get.models(modsel_fitness_2020,subset=1)$"9102",
                             terms=c("temp[all]", "FFD_corr[all]"))
ggplot(predict_fitness2,aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high,
                           group=group))+
  geom_line(aes(color=as.numeric(as.character(group))),size=0.3)+
  xlab("Soil temperature (°C)")+ylab("Predicted fitness")+
  my_theme_legend()+
  scale_colour_gradientn(colours = plasma(n=132,begin=0.4,end=0.9,direction=-1))+
  labs(colour="FFD")
```



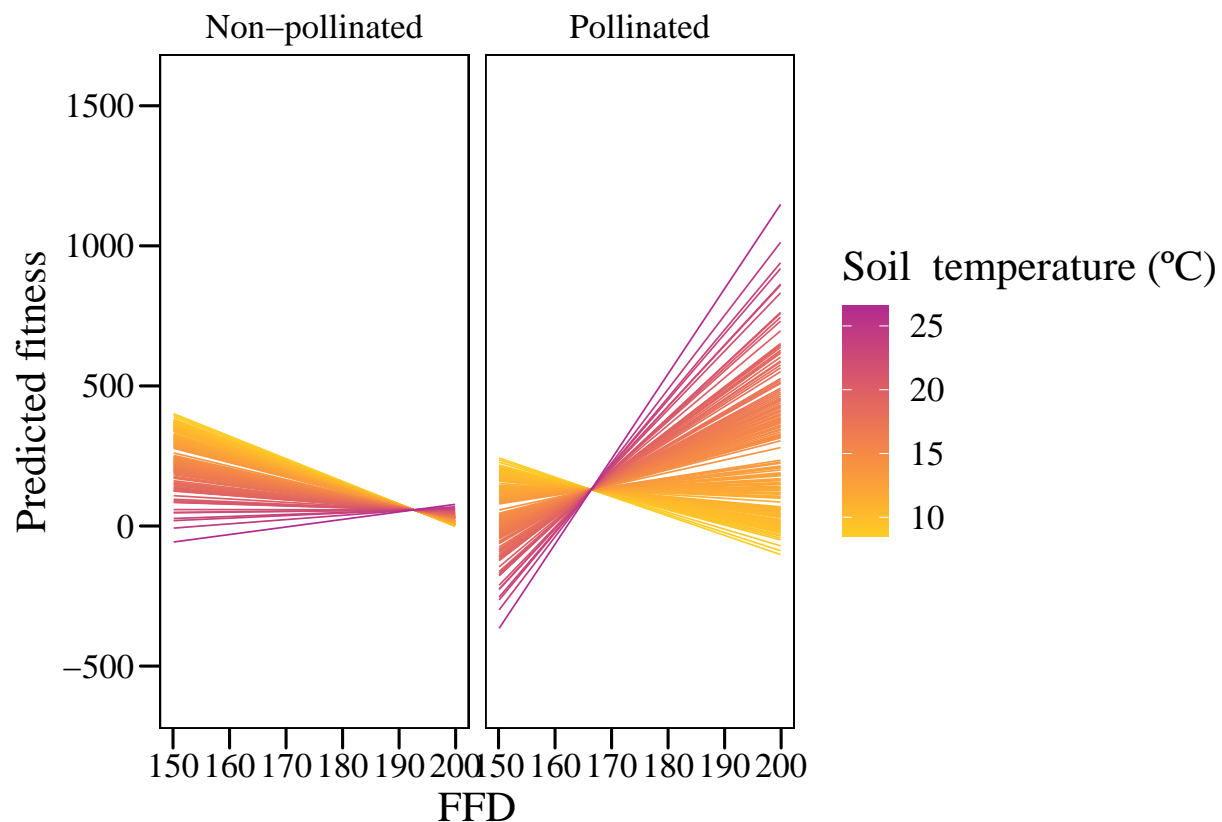
```
ggsave(filename="output/figures/fitness_2.tiff",device="tiff",
        width=11,height=8,units="cm",dpi=300,compression="lzw")
```

```
predict_fitness3<-ggpredict(get.models(modsel_fitness_2020,subset=1)$"9102",
                             terms=c("temp[all]", "P"))
ggplot(predict_fitness3,aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high,
                             group=group))+
  geom_ribbon(aes(fill=group),alpha=0.2)+
  geom_line(aes(color=group),size=1)+
  xlab("FFD")+ylab("Predicted fitness")+
  scale_fill_discrete(name=NULL,labels=c("Non-pollinated","Pollinated"))+
  scale_color_discrete(name=NULL,labels=c("Non-pollinated","Pollinated"))+
  my_theme_legend()
```



```
ggsave(filename="output/figures/fitness_3.tiff",device="tiff",
        width=14,height=8,units="cm",dpi=300,compression="lzw")
```

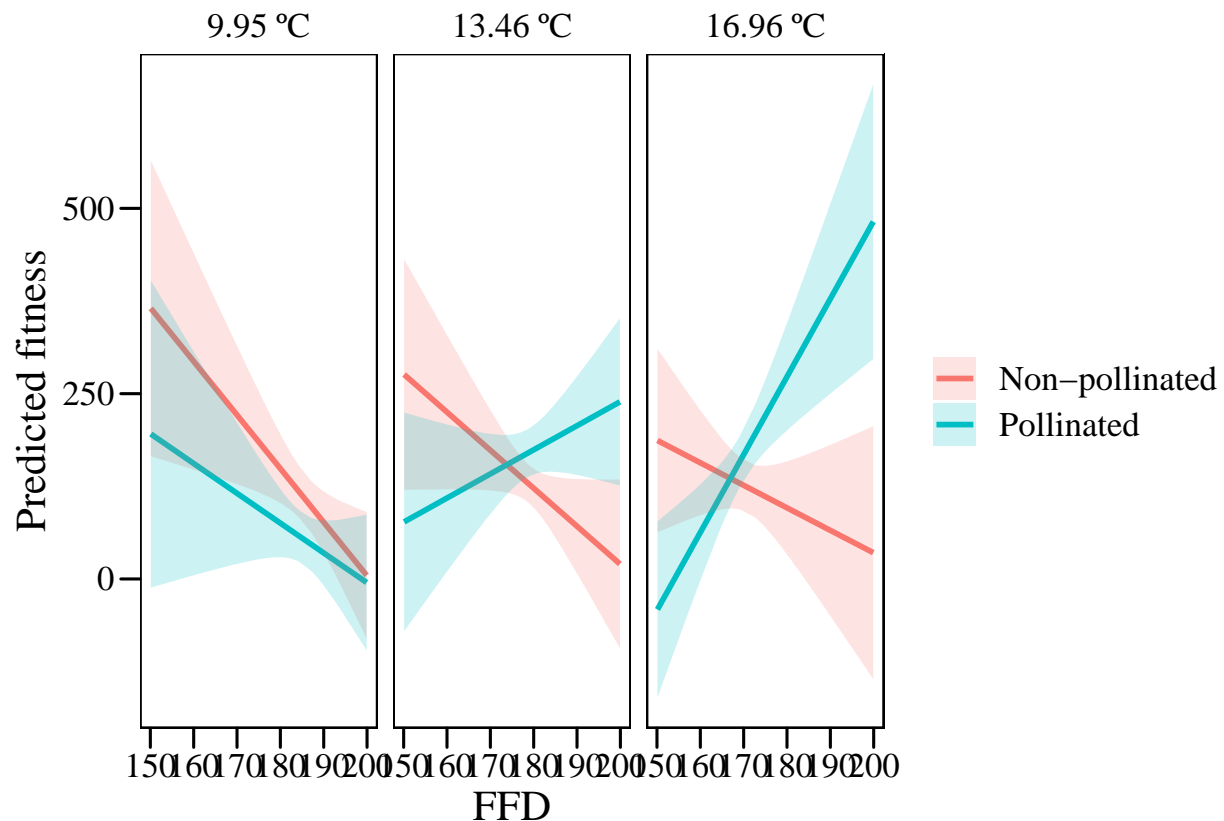
```
predict_fitness4<-ggpredict(get.models(modsel_fitness_2020,subset=1)$"9102",
                             terms=c("FFD_corr[all]","temp[all]","P"))
P.labs <- c("Non-pollinated","Pollinated")
names(P.labs) <- c(0,1)
ggplot(predict_fitness4,aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high,
                             group=group))+
  geom_line(aes(color=as.numeric(as.character(group))),size=0.3)+
  facet_wrap(~facet,labeller=labeller(facet=P.labs))+
  xlab("FFD")+ylab("Predicted fitness")+
  my_theme_legend()+
  scale_colour_gradientn(colours = plasma(n=132,begin=0.4,end=0.9,direction=-1))+
  labs(colour="Soil temperature (°C)")
```



```
ggsave(filename="output/figures/fitness_4.tiff",device="tiff",
        width=20,height=8,units="cm",dpi=300,compression="lzw")
```

```
predict_fitness5<-ggpredict(get.models(modsel_fitness_2020,
                                         subset=1)$"9102",terms=c("FFD_corr[all]","P","temp"))
temp.labs <- c("9.95 °C","13.46 °C","16.96 °C")
names(temp.labs) <- c(9.95,13.46,16.96)
ggplot(predict_fitness5,aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high,
                             group=group))+
```

```
geom_line(aes(color=group),size=1)+
geom_ribbon(aes(fill=group),alpha=0.2)+
facet_wrap(~facet,labeller=labeller(facet=temp.labs))+
xlab("FFD")+ylab("Predicted fitness")+
scale_fill_discrete(name=NULL,labels=c("Non-pollinated","Pollinated"))+
scale_color_discrete(name=NULL,labels=c("Non-pollinated","Pollinated"))+
my_theme_legend()+
labs(colour=NULL,fill=NULL)
```



```
ggsave(filename="output/figures/fitness_5.tiff",device="tiff",
width=20,height=8,units="cm",dpi=300,compression="lzw")
```

3. Models for relative fitness (selection models)

Standardize traits and relativize fitness:

```
# Standardize FFD
ping_20_data$FFD_corr_std<-as.numeric(scale(ping_20_data$FFD_corr))
# Standardize rosette area (first log, then scale)
ping_20_data$ros_area_std<-as.numeric(scale(log(ping_20_data$ros_area)))
# Relativize fitness
ping_20_data$n_seeds_rel<-with(ping_20_data,n_seeds/mean(n_seeds,na.rm=T))
```

Model selection

```
subset2<-subset(ping_20_data,!is.na(n_seeds_rel)&!is.na(ros_area_std))
globmod_selection_2020<-lm(n_seeds_rel~temp*FFD_corr_std*P*F+ros_area_std,
                           subset2,na.action="na.fail")
clusterType <- if(length(find.package("snow", quiet = TRUE))) "SOCK" else "PSOCK"
clust1 <- try(makeCluster(getOption("cl.cores", 3), type = clusterType))
clusterExport(clust1, "subset2")
clusterEvalQ(clust1, library(stats))
```

```
## [[1]]
## [1] "stats"      "graphics"   "grDevices"  "utils"      "datasets"   "methods"
## [7] "base"
##
## [[2]]
## [1] "stats"      "graphics"   "grDevices"  "utils"      "datasets"   "methods"
## [7] "base"
##
## [[3]]
## [1] "stats"      "graphics"   "grDevices"  "utils"      "datasets"   "methods"
## [7] "base"
```

```
modsel_selection_2020<-pdredge(globmod_selection_2020,fixed=c("ros_area_std"),
                               cluster=clust1)

summary(model.avg(modsel_selection_2020,
                  subset=delta<2)) # Summary averaged model
```

```
##
## Call:
## model.avg(object = modsel_selection_2020, subset = delta < 2)
##
## Component model call:
## lm(formula = n_seeds_rel ~ <2 unique rhs>, data = subset2, na.action =
##     na.fail)
##
## Component models:
##           df logLik   AICc delta weight
## 23456789  10 -424.46 869.76  0.00   0.57
## 123456789 11 -423.66 870.33  0.57   0.43
##
## Term codes:
##           F           FFD_corr_std           P           ros_area_std
##           1           2           3           4
##           temp       FFD_corr_std:P   FFD_corr_std:temp       P:temp
##           5           6           7           8
## FFD_corr_std:P:temp
##           9
##
## Model-averaged coefficients:
## (full average)
##           Estimate Std. Error Adjusted SE z value Pr(>|z|)
```

```
## (Intercept)          2.15149    0.83299    0.83683    2.571 0.010141 *
## FFD_corr_std        -1.07479    0.36065    0.36232    2.966 0.003013 **
## P1                   -3.78833    1.18381    1.18927    3.185 0.001445 **
## temp                -0.07189    0.06284    0.06313    1.139 0.254832
## ros_area_std         0.05636    0.07576    0.07611    0.740 0.459007
## FFD_corr_std:P1     -0.92851    0.62232    0.62519    1.485 0.137502
## FFD_corr_std:temp    0.04941    0.02236    0.02247    2.199 0.027852 *
## P1:temp              0.31113    0.09064    0.09105    3.417 0.000633 ***
## FFD_corr_std:P1:temp 0.11975    0.04164    0.04184    2.862 0.004205 **
## F1                   -0.07642    0.12861    0.12893    0.593 0.553328
##
## (conditional average)
## Estimate Std. Error Adjusted SE z value Pr(>|z|)
## (Intercept)          2.15149    0.83299    0.83683    2.571 0.010141 *
## FFD_corr_std        -1.07479    0.36065    0.36232    2.966 0.003013 **
## P1                   -3.78833    1.18381    1.18927    3.185 0.001445 **
## temp                -0.07189    0.06284    0.06313    1.139 0.254832
## ros_area_std         0.05636    0.07576    0.07611    0.740 0.459007
## FFD_corr_std:P1     -0.92851    0.62232    0.62519    1.485 0.137502
## FFD_corr_std:temp    0.04941    0.02236    0.02247    2.199 0.027852 *
## P1:temp              0.31113    0.09064    0.09105    3.417 0.000633 ***
## FFD_corr_std:P1:temp 0.11975    0.04164    0.04184    2.862 0.004205 **
## F1                   -0.17797    0.14299    0.14365    1.239 0.215371
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
importance(modsel_selection_2020) # Variable importance
```

```
##               ros_area_std temp  FFD_corr_std FFD_corr_std:temp P
## Sum of weights:           1           1           1           1           0.97
## N containing models:    167          148    148           84          148
##               P:temp FFD_corr_std:P FFD_corr_std:P:temp F      F:P
## Sum of weights:      0.96    0.95           0.91           0.68 0.24
## N containing models:    84     84           20           148  84
##               F:FFD_corr_std F:temp F:P:temp F:FFD_corr_std:P
## Sum of weights:      0.2           0.2    0.02           0.02
## N containing models:    84           84     20           20
##               F:FFD_corr_std:temp F:FFD_corr_std:P:temp
## Sum of weights:      0.02           <0.01
## N containing models:    20           1
```

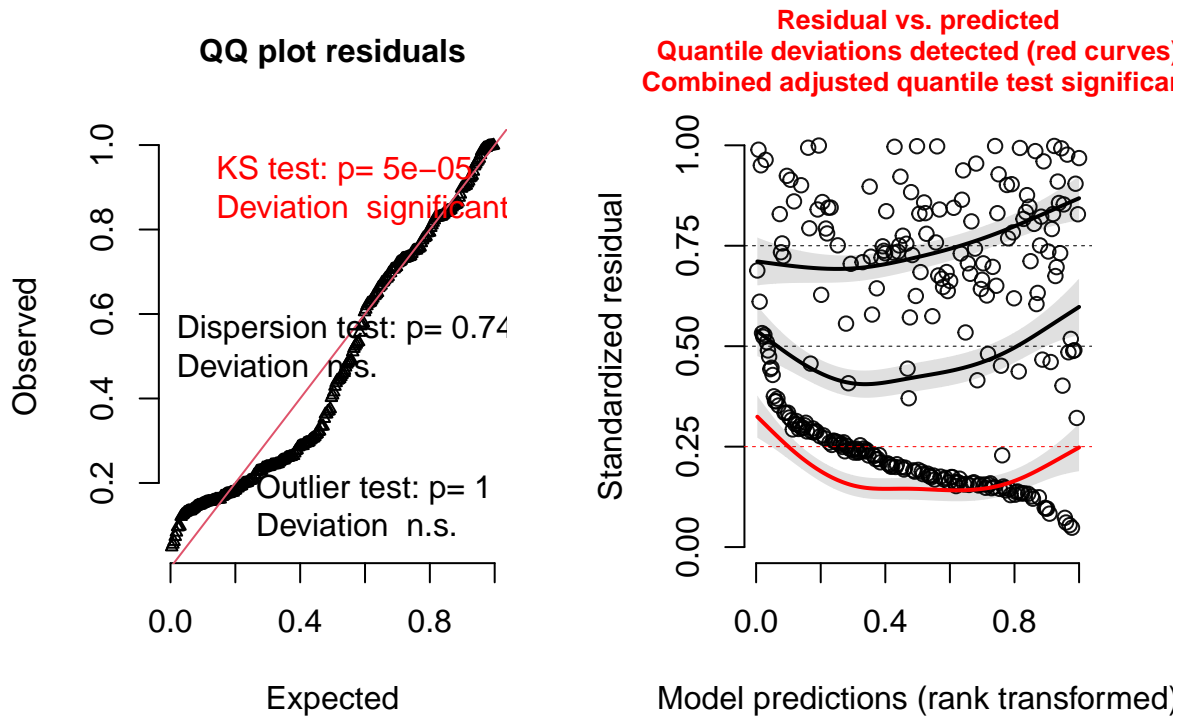
```
r.squaredGLMM(get.models(modsel_selection_2020,subset=1)$"9102") #R square of best model
```

```
##           R2m           R2c
## [1,] 0.1221926 0.1221926
```

Model diagnostics for best model

```
plot(simulateResiduals(fittedModel=get.models(modsel_selection_2020,
                                                subset=1)$"9102",n=5000))
```

DHARMA residual diagnostics



BCa intervals for averaged model

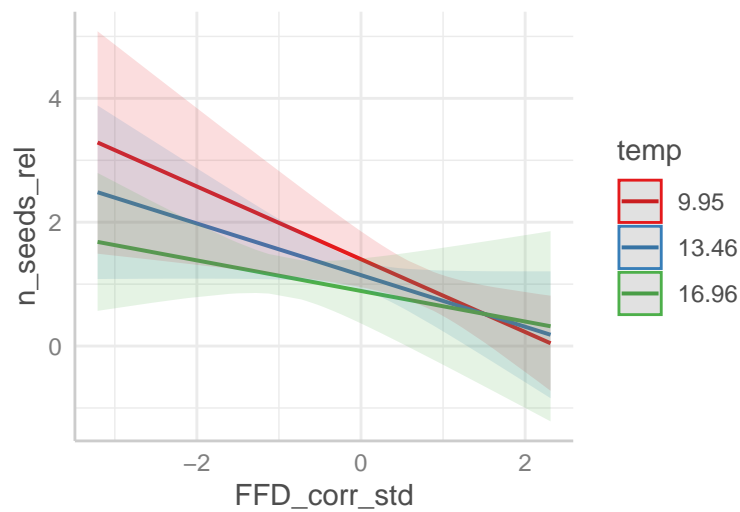
BCIs_av_selection_2020

##		lower	upper
##	FFD_corr_std	-1.70338017	-0.49505278
##	P1	-6.17686342	-1.28234156
##	temp	-0.18645778	0.04785985
##	ros_area	-0.07752336	0.19322921
##	FFD_corr_std:P1	-2.14811207	0.28185004
##	FFD_corr_std:temp	0.01953391	0.08653356
##	P1:temp	0.12511394	0.49313806
##	FFD_corr:P1:temp	0.03526116	0.20898297
##	F1	-0.45294620	0.11129898

Model predictions from best model

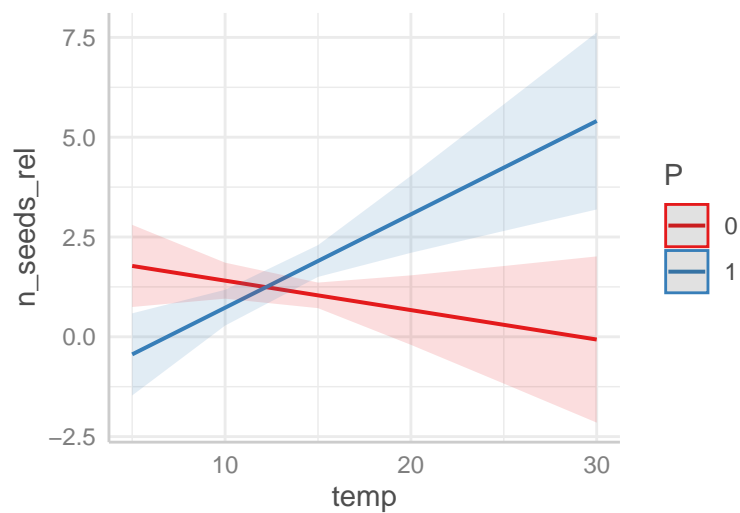
```
plot(ggpredict(get.models(modsel_selection_2020, subset=1)$"9102",
  terms=c("FFD_corr_std[all]", "temp")))
```


Predicted values of n_seeds_rel



```
plot(ggpredict(get.models(modsel_selection_2020,subset=1)$"9102",
  terms=c("temp","P")))
```

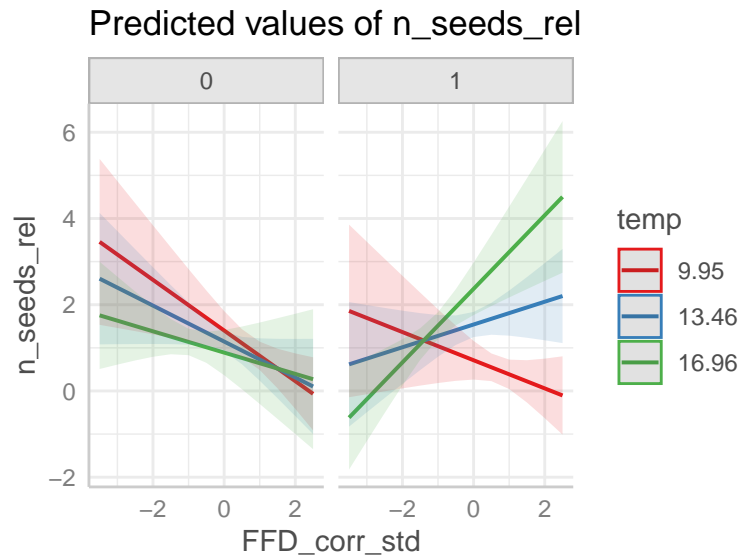
Predicted values of n_seeds_rel



tations):

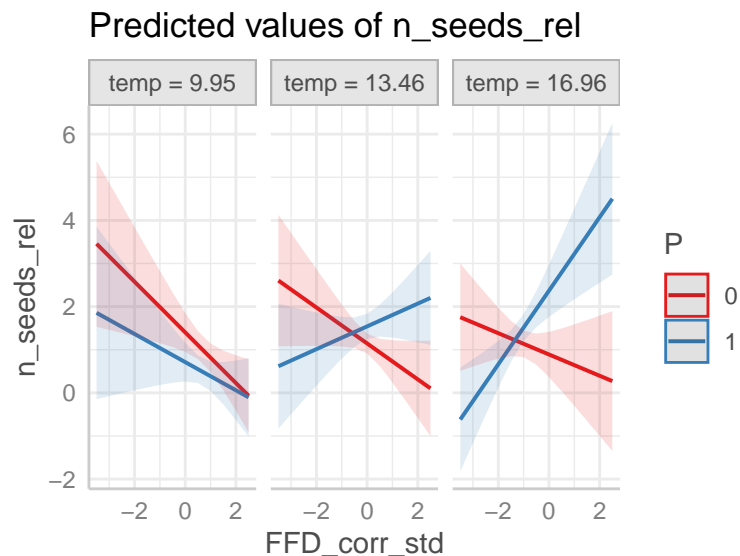
3-way interaction (two different represen-

```
plot(ggpredict(get.models(modsel_selection_2020,subset=1)$"9102",
  terms=c("FFD_corr_std","temp","P")))
```



For non-pollinated plants, it is always better to flower early, irrespective of soil temperatures. For pollinated plants, it is better to flower early at cold soil temperatures, but late at medium and warm soil temperatures.

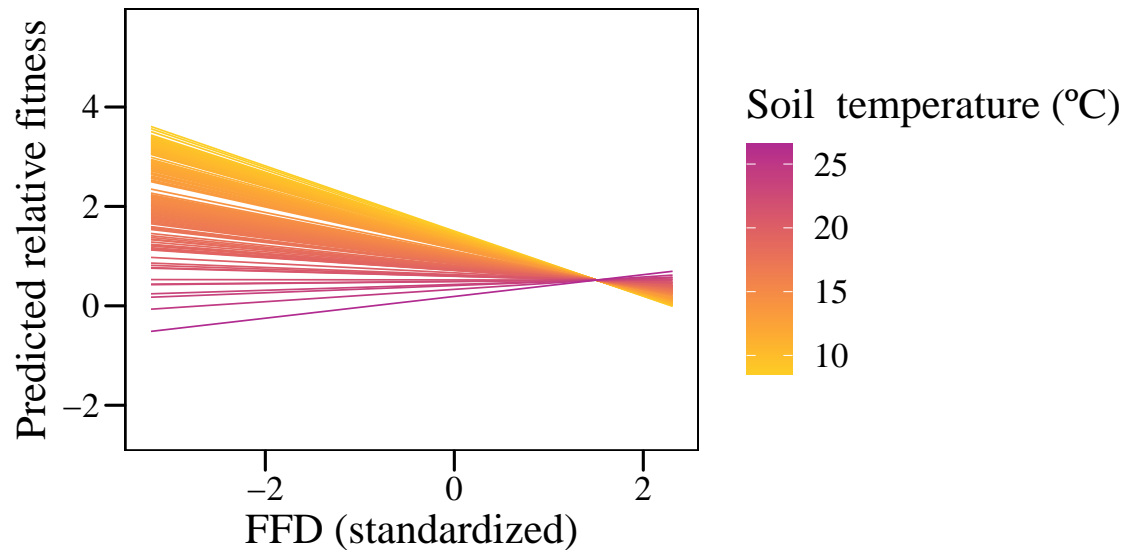
```
plot(ggpredict(get.models(modsel_selection_2020,
subset=1)$"9102",terms=c("FFD_corr_std","P","temp")))
```



At low temperatures (left panel above), it is always better to flower early, irrespective of if the plant was pollinated or not. At medium and high temperatures (center and right panel above), it is better to flower early for non-pollinated plants, but better to flower late for pollinated plants.

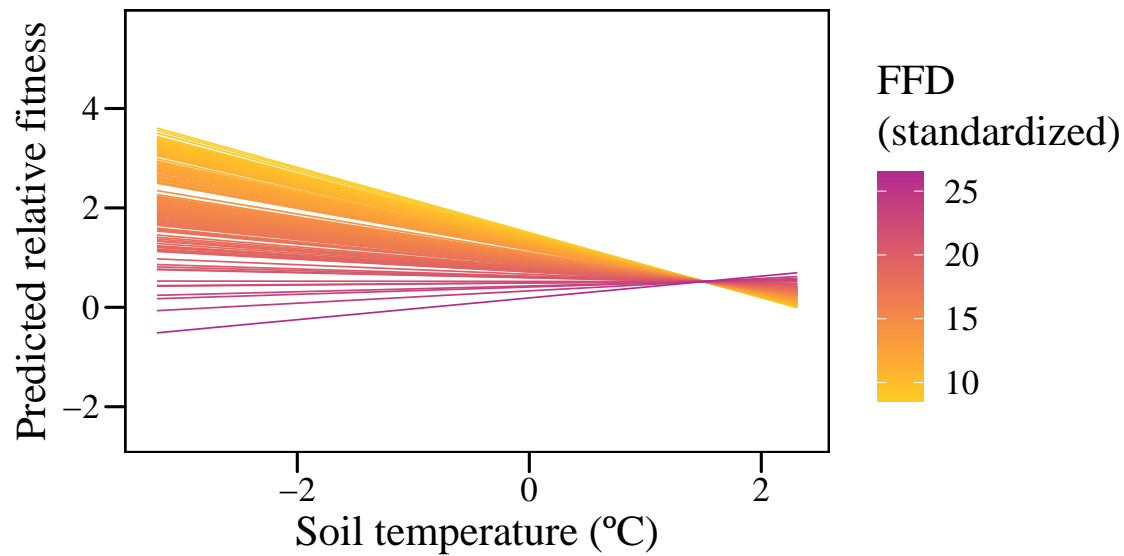
```
predict_relfitness1<-ggpredict(get.models(modsel_selection_2020,subset=1)$"9102",
terms=c("FFD_corr_std[all]","temp[all]"))
ggplot(predict_relfitness1,aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high,
group=group))+
geom_line(aes(color=as.numeric(as.character(group))),size=0.3)+
xlab("FFD (standardized)")+ylab("Predicted relative fitness")+
```

```
my_theme_legend()+
scale_colour_gradientn(colours = plasma(n=132,begin=0.4,end=0.9,direction=-1))+
labs(colour="Soil temperature (°C)")
```



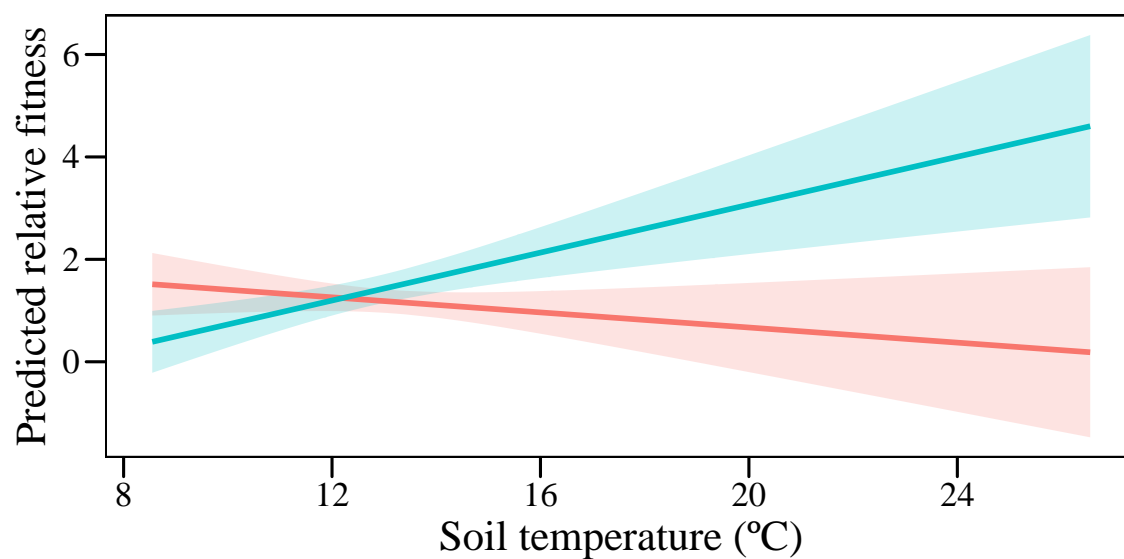
```
ggsave(filename="output/figures/relfitness_1.tiff",device="tiff",
        width=15,height=8,units="cm",dpi=300,compression="lzw")

predict_relfitness2<-ggpredict(get.models(modsel_selection_2020,subset=1)$"9102",
                              terms=c("temp[all]", "FFD_corr_std[all]"))
ggplot(predict_relfitness1,aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high,
                              group=group))+
  geom_line(aes(color=as.numeric(as.character(group))),size=0.3)+
  xlab("Soil temperature (°C)")+ylab("Predicted relative fitness")+
  my_theme_legend()+
  scale_colour_gradientn(colours = plasma(n=132,begin=0.4,end=0.9,direction=-1))+
  labs(colour="FFD\n(standardized)")
```



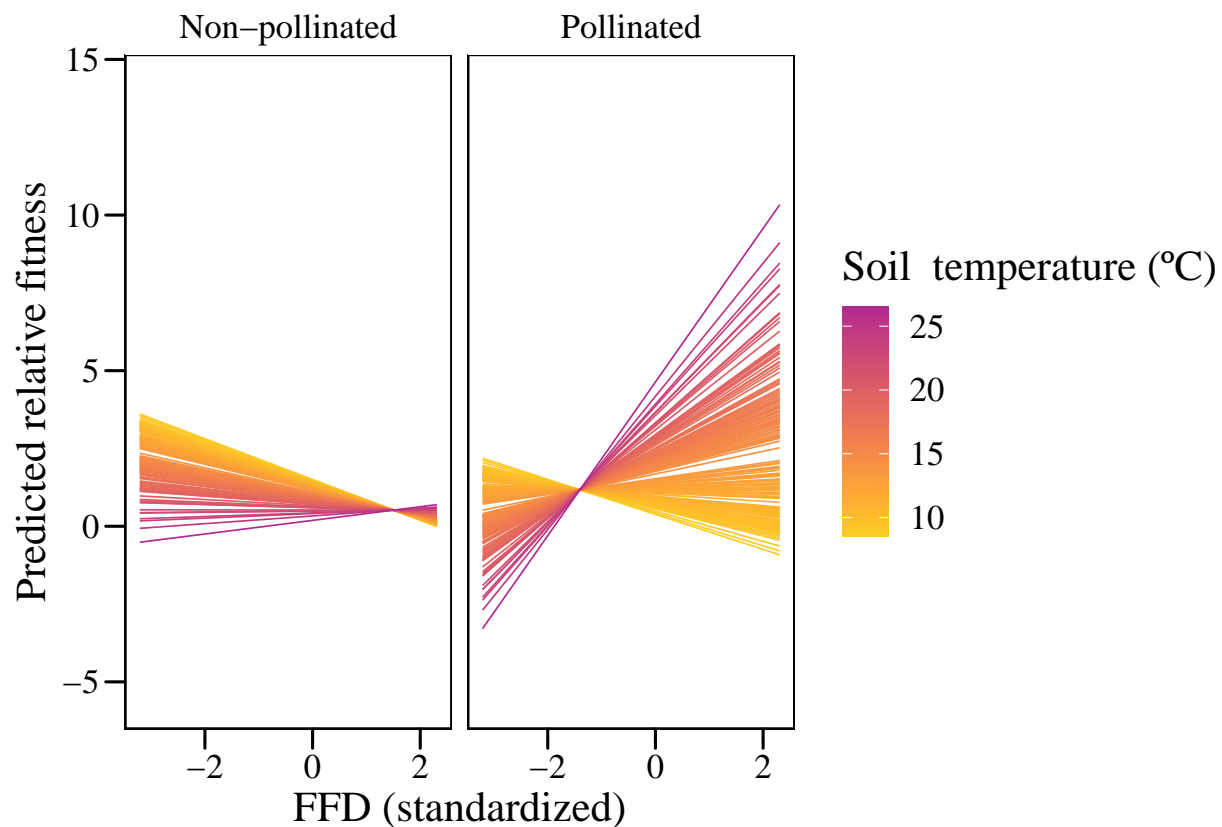
```
ggsave(filename="output/figures/relfitness_2.tiff",device="tiff",
        width=13,height=8,units="cm",dpi=300,compression="lzw")
```

```
predict_relfitness3<-ggpredict(get.models(modsel_selection_2020,subset=1)$"9102",
                               terms=c("temp[all]", "P"))
ggplot(predict_relfitness3,aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high,
                               group=group))+
  geom_ribbon(aes(fill=group),alpha=0.2)+
  geom_line(aes(color=group),size=1)+
  xlab("Soil temperature (°C)")+ylab("Predicted relative fitness")+
  scale_fill_discrete(name=NULL,labels=c("Non-pollinated","Pollinated"))+
  scale_color_discrete(name=NULL,labels=c("Non-pollinated","Pollinated"))+
  my_theme()
```



```
ggsave(filename="output/figures/relfitness_3.tiff",device="tiff",
        width=10,height=8,units="cm",dpi=300,compression="lzw")
```

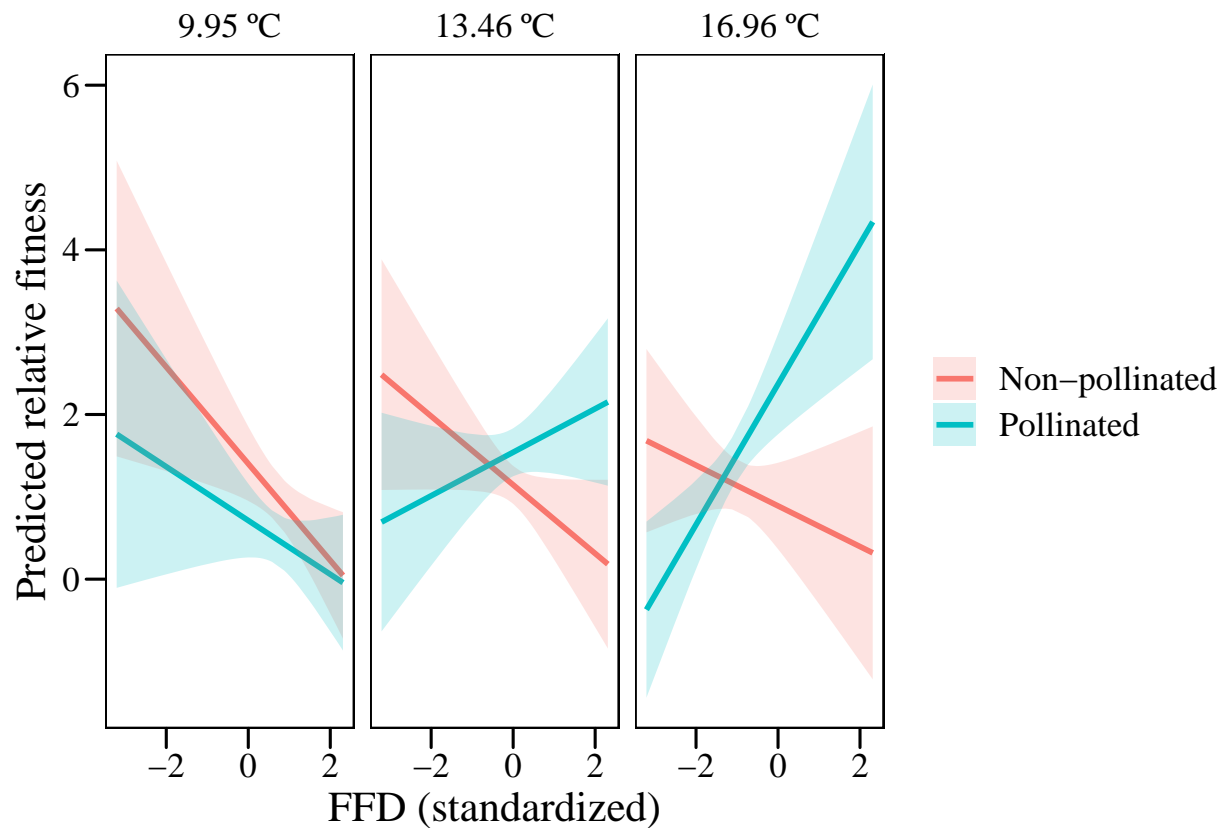
```
predict_relfitness4<-ggpredict(get.models(modsel_selection_2020,subset=1)$"9102",
                              terms=c("FFD_corr_std[all]","temp[all]","P"))
ggplot(predict_relfitness4,aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high,
                              group=group))+
  geom_line(aes(color=as.numeric(as.character(group))),size=0.3)+
  facet_wrap(~facet,labeller=labeller(facet=P.labs))+
  xlab("FFD (standardized)")+ylab("Predicted relative fitness")+
  my_theme_legend()+
  scale_colour_gradientn(colours = plasma(n=132,begin=0.4,end=0.9,direction=-1))+
  labs(colour="Soil temperature (°C)")
```



```
ggsave(filename="output/figures/relfitness_4.tiff",device="tiff",
        width=20,height=8,units="cm",dpi=300,compression="lzw")
```

```
predict_relfitness5<-ggpredict(get.models(modsel_selection_2020,
                                          subset=1)$"9102",terms=c("FFD_corr_std[all]","P","temp"))
ggplot(predict_relfitness5,aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high,
                              group=group))+
  geom_line(aes(color=group),size=1)+
  geom_ribbon(aes(fill=group),alpha=0.2)+
  facet_wrap(~facet,labeller=labeller(facet=temp.labs))+
  xlab("FFD (standardized)")+ylab("Predicted relative fitness")+
```

```
scale_fill_discrete(name=NULL,labels=c("Non-pollinated","Pollinated"))+
scale_color_discrete(name=NULL,labels=c("Non-pollinated","Pollinated"))+
my_theme_legend()+
labs(colour=NULL,fill=NULL)
```



```
ggsave(filename="output/figures/relfitness_5.tiff",device="tiff",
width=20,height=8,units="cm",dpi=300,compression="lzw")
```

Predictions for minimum and maximum temperatures:

```
ggpredict(get.models(modsel_selection_2020,subset=1)$"9102",
terms=c("temp[minmax]","P"))
```

```
## # Predicted values of n_seeds_rel
##
## # P = 0
##
## temp | Predicted |          95% CI
## -----
## 8.55 |          1.51 | [ 0.90, 2.12]
## 26.55 |          0.19 | [-1.48, 1.85]
##
## # P = 1
##
```

```
## temp | Predicted |          95% CI
## -----
## 8.55 |      0.39 | [-0.22, 0.99]
## 26.55 |      4.60 | [ 2.82, 6.38]
##
## Adjusted for:
## * FFD_corr_std = -0.01
## * ros_area_std = 0.01
```

```
# 190.85-155.16=35.69 days earlier on warmer soils
```

4. Models for binomial fitness (maybe not included in paper)

Model selection

```
subset1<-subset(ping_20_data,!is.na(fitness_01)&!is.na(ros_area_std))
globmod_selection_2020_01<-glm(fitness_01~temp*FFD_corr_std*P*F+ros_area_std,
                             subset1,family="binomial",na.action="na.fail")
clusterType <- if(length(find.package("snow", quiet = TRUE))) "SOCK" else "PSOCK"
clust1 <- try(makeCluster(getOption("cl.cores", 3), type = clusterType))
clusterExport(clust1, "subset1")
clusterEvalQ(clust1, library(stats))
```

```
## [[1]]
## [1] "stats"      "graphics"  "grDevices" "utils"      "datasets"  "methods"
## [7] "base"
##
## [[2]]
## [1] "stats"      "graphics"  "grDevices" "utils"      "datasets"  "methods"
## [7] "base"
##
## [[3]]
## [1] "stats"      "graphics"  "grDevices" "utils"      "datasets"  "methods"
## [7] "base"
```

```
modsel_selection_2020_01<-pdredge(globmod_selection_2020_01,
                                fixed=c("ros_area_std"),
                                cluster=clust1)

summary(model.avg(modsel_selection_2020_01,
                  subset=delta<2)) # Summary averaged model
```

```
##
## Call:
## model.avg(object = modsel_selection_2020_01, subset = delta <
##      2)
##
## Component model call:
## glm(formula = fitness_01 ~ <4 unique rhs>, family = binomial, data =
```

```
## subset1, na.action = na.fail)
##
## Component models:
##      df logLik AICc delta weight
## 2457  5 -176.74 363.70 0.00 0.42
## 23457  6 -176.42 365.16 1.46 0.20
## 12457  6 -176.43 365.18 1.48 0.20
## 124567 7 -175.53 365.48 1.79 0.17
##
## Term codes:
##      F      FFD_corr_std      P      ros_area_std
##      1      2      3      4
##      temp      F:temp FFD_corr_std:temp
##      5      6      7
##
## Model-averaged coefficients:
## (full average)
##      Estimate Std. Error Adjusted SE z value Pr(>|z|)
## (Intercept) -1.77044 1.13408 1.13890 1.555 0.120061
## FFD_corr_std -2.57780 0.70568 0.70887 3.636 0.000276 ***
## temp 0.16804 0.08712 0.08749 1.921 0.054777 .
## ros_area_std -0.12601 0.13418 0.13479 0.935 0.349869
## FFD_corr_std:temp 0.19370 0.05029 0.05052 3.834 0.000126 ***
## P1 -0.04112 0.14106 0.14148 0.291 0.771335
## F1 -0.31941 0.75782 0.75913 0.421 0.673930
## F1:temp 0.01836 0.05197 0.05206 0.353 0.724426
##
## (conditional average)
##      Estimate Std. Error Adjusted SE z value Pr(>|z|)
## (Intercept) -1.77044 1.13408 1.13890 1.555 0.120061
## FFD_corr_std -2.57780 0.70568 0.70887 3.636 0.000276 ***
## temp 0.16804 0.08712 0.08749 1.921 0.054777 .
## ros_area_std -0.12601 0.13418 0.13479 0.935 0.349869
## FFD_corr_std:temp 0.19370 0.05029 0.05052 3.834 0.000126 ***
## P1 -0.20241 0.25553 0.25669 0.789 0.430382
## F1 -0.85266 1.03845 1.04100 0.819 0.412744
## F1:temp 0.10612 0.07938 0.07974 1.331 0.183224
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
importance(modsel_selection_2020_01) # Variable importance
```

```
##      ros_area_std temp FFD_corr_std FFD_corr_std:temp P
## Sum of weights:      1      1      1      1      0.64
## N containing models: 167      148      148      84      148
##      F      P:temp F:temp FFD_corr_std:P F:FFD_corr_std F:P
## Sum of weights:      0.64 0.36 0.29 0.28      0.24      0.12
## N containing models: 148      84      84      84      84      84
##      FFD_corr_std:P:temp F:FFD_corr_std:temp F:P:temp
## Sum of weights:      0.07      0.03      <0.01
## N containing models: 20      20      20
##      F:FFD_corr_std:P F:FFD_corr_std:P:temp
## Sum of weights:      <0.01      <0.01
## N containing models: 20      1
```



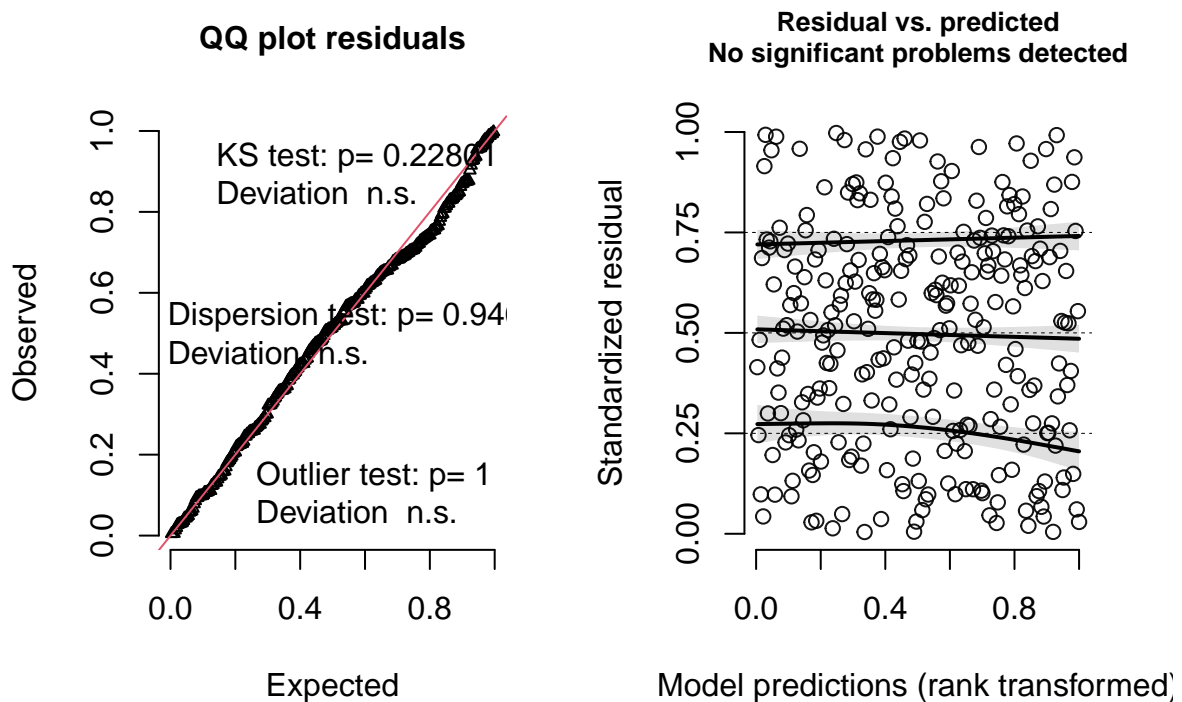
```
r.squaredGLMM(get.models(modsel_selection_2020_01,subset=1)$"266") #R square of best model
```

```
##               R2m       R2c
## theoretical 0.1490344 0.1490344
## delta      0.1256198 0.1256198
```

Model diagnostics for best model

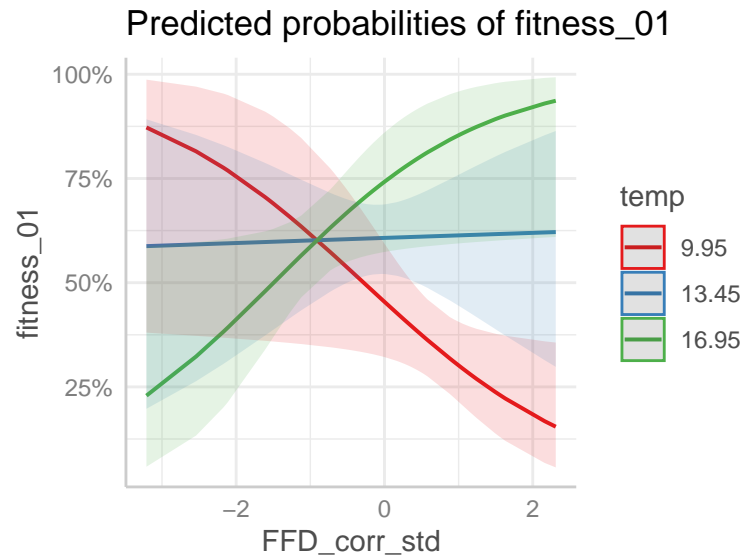
```
plot(simulateResiduals(fittedModel=get.models(modsel_selection_2020_01,
subset=1)$"266",n=5000))
```

DHARMA residual diagnostics



Model predictions from best model

```
plot(ggpredict(get.models(modsel_selection_2020_01,subset=1)$"266",
terms=c("FFD_corr_std[all]", "temp")))
```



We could use the method described by Janzen and Stern (1998) to transform the logit coefficients to linear scale, but not sure that is needed. Some code for that here: https://github.com/mabarbour/complexity_selection/blob/master/analyses/reproduce_analyses.Rmd

5. Aster models

Any Fruits (seedpods_01) → Fruits (n_seedpods) → Seeds (n_seeds)

Any fruits (did the plant produce fruits) is Bernoulli Number of Fruits is zero-truncated negative binomial Number of seeds is Poisson

Reshape data:

```
ping_20_data$n_seedpods<-as.integer(ping_20_data$n_seedpods)
ping_20_data$n_seeds<-as.integer(ping_20_data$n_seeds)
ping_20_data$seedpods_01<-with(ping_20_data,ifelse(n_seedpods>0,1,0))
vars <- c("seedpods_01", "n_seedpods","n_seeds") # Our fitness variables
redata <- reshape(data.frame(subset(ping_20_data,
                                     !is.na(n_seeds)&!is.na(ros_area_std))
                                     [c(1:2,10,15:16,21:29)]),
                  varying = list(vars),
                  direction = "long",
                  timevar = "varb", # reshape to long format
                  times = as.factor(vars), v.names = "resp")
redata <- data.frame(redata, root = 1) # This adds a variable root to the df
# and makes all its values one (including for non-root nodes, but those values
# are ignored by all aster package functions).

# resp contains all of the data in the variables indicated by the string
# vars packed into a single vector
# varb indicates which original variable the corresponding element of
# resp came from
```

```
names(data.frame(subset(ping_20_data,
                        !is.na(n_seeds)&!is.na(ros_area_std))
                [c(1:2,10,15:16,21:29)]))
```

```
## [1] "id"          "FFD_corr"      "temp"          "n_seedpods"    "n_seeds"
## [6] "ros_area"      "P"             "F"             "fitness_01"    "ros_area_log"
## [11] "FFD_corr_std" "ros_area_std" "n_seeds_rel"   "seedpods_01"
```

```
names(redata)
```

```
## [1] "id"          "FFD_corr"      "temp"          "ros_area"      "P"
## [6] "F"           "fitness_01"    "ros_area_log"  "FFD_corr_std"  "ros_area_std"
## [11] "n_seeds_rel"  "varb"          "resp"          "root"
```

```
nrow(data.frame(subset(ping_20_data,
                        !is.na(n_seeds)&!is.na(ros_area_std))
                [c(1:2,10,15:16,21:29)]))
```

```
## [1] 273
```

```
nrow(redata)
```

```
## [1] 819
```

```
nrow(data.frame(subset(ping_20_data,
                        !is.na(n_seeds)&!is.na(ros_area_std))
                [c(1:2,10,15:16,21:29)])) * length(vars)
```

```
## [1] 819
```

```
sapply(redata, class)
```

```
##          id      FFD_corr      temp      ros_area      P      F
## "character" "numeric"    "numeric" "numeric"    "factor" "factor"
## fitness_01 ros_area_log FFD_corr_std ros_area_std n_seeds_rel varb
## "numeric"   "numeric"    "numeric" "numeric"    "numeric" "factor"
##      resp      root
## "numeric"    "numeric"
```

```
levels(redata$varb)
```

```
## [1] "n_seedpods" "n_seeds"     "seedpods_01"
```

```
length(unique(redata$id))
```

```
## [1] 273
```

Show the graphical model:

```
pred <- c(0,1,2) # specifies the predecessor structure of the graph
```

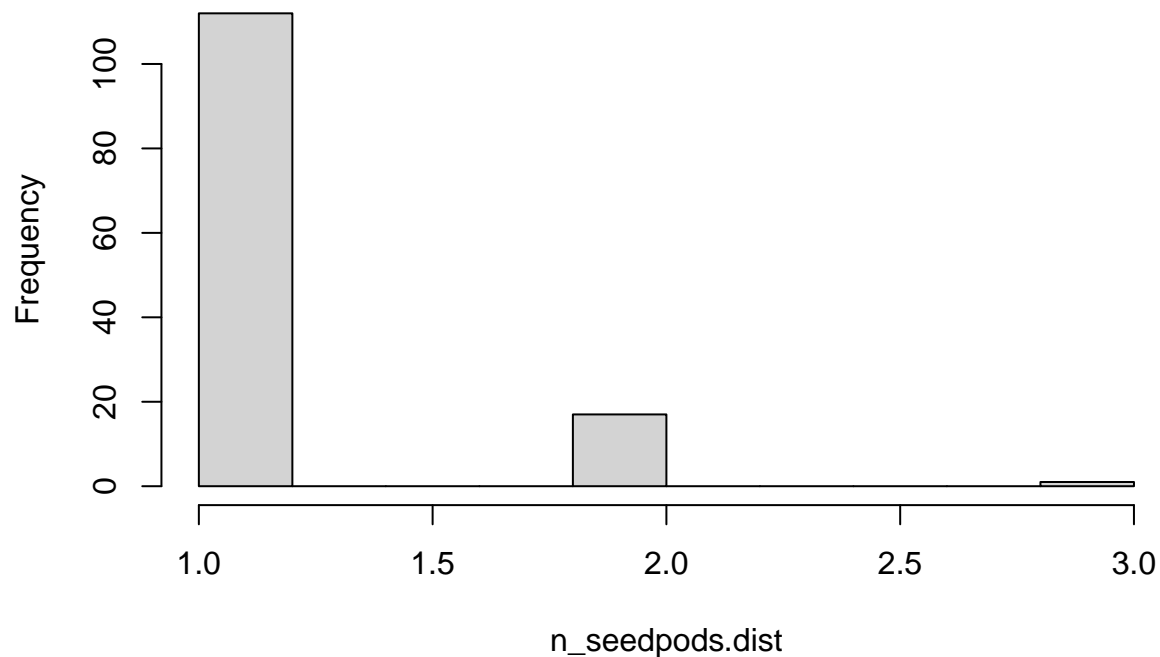
```
foo <- c("root", vars)
pvars <- foo[pred + 1]
bar <- cbind(pvars, vars)
colnames(bar) <- c("pred", "succ")
bar
```

```
##      pred      succ
## [1,] "root"      "seedpods_01"
## [2,] "seedpods_01" "n_seedpods"
## [3,] "n_seedpods" "n_seeds"
```

Look at Poisson distributions for seedpods and seeds:

```
#For n_seedpods
n_seedpods.dist <- subset(ping_20_data,
                          !is.na(n_seeds)&
                          !is.na(ros_area_std))$n_seedpods[subset(ping_20_data,
                          !is.na(n_seeds)&!is.na(ros_area_std))$seedpods_01 == 1]
hist(n_seedpods.dist)
```

Histogram of n_seedpods.dist



```
length(n_seedpods.dist) #130
```

```
## [1] 130
```

```
sum(n_seedpods.dist == 0) #0
```

```
## [1] 0
```

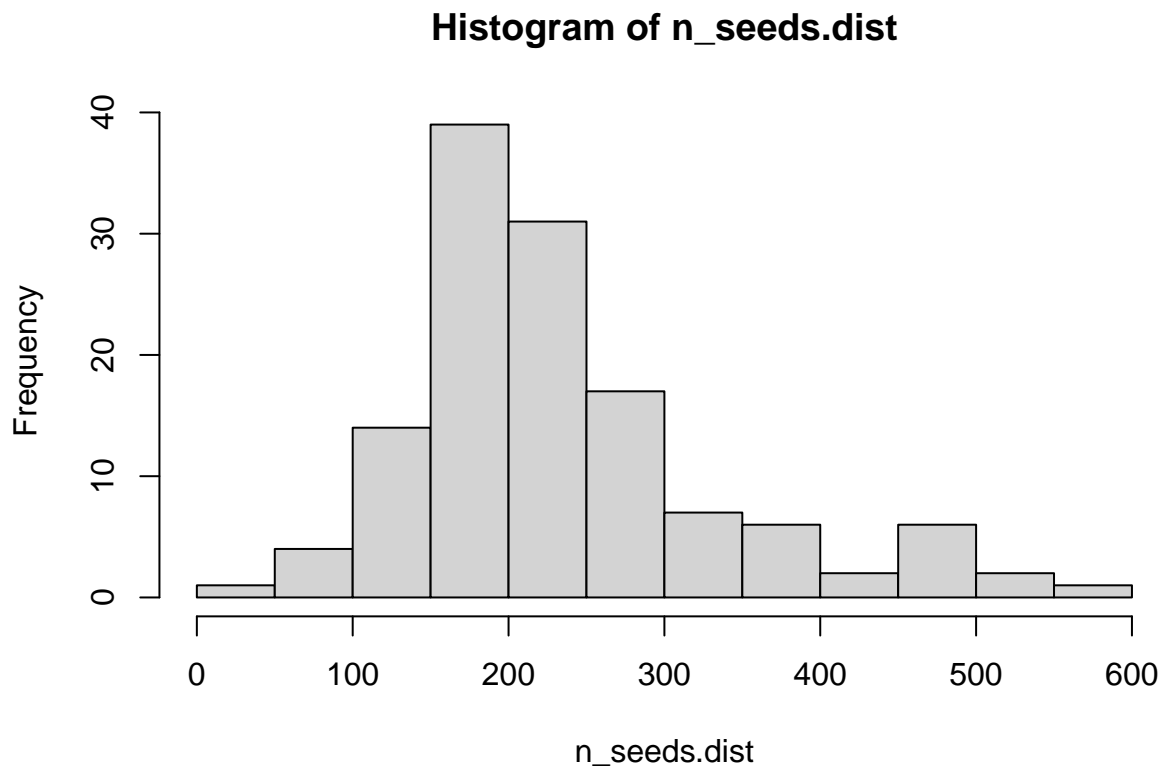
```
n_seedpods.parms <- fitdistr(n_seedpods.dist, "poisson")
```

```
n_seedpods.parms #lambda = 1.14615385
```

```
##      lambda  
## 1.14615385  
## (0.09389658)
```

```
#For n_seeds
```

```
n_seeds.dist <- subset(ping_20_data,  
  !is.na(n_seeds)&  
  !is.na(ros_area_std))$n_seeds[subset(ping_20_data,  
  !is.na(n_seeds)&  
  !is.na(ros_area_std))$n_seedpods > 0]  
hist(n_seeds.dist)
```



```
length(n_seeds.dist) #130
```

```
## [1] 130
```

```
sum(n_seeds.dist == 0) #0
```

```
## [1] 0
```

```
n_seeds.parms <- fitdistr(n_seeds.dist, "poisson")  
n_seeds.parms # lambda = 234.069231
```

```
##      lambda  
## 234.069231  
## ( 1.341839)
```

Look at negative binomial distributions for seedpods and seeds:

```
#For n_seedpods  
n_seedpods.parms2 <- fitdistr(n_seedpods.dist, "negative binomial")  
n_seedpods.parms2 #size = 4.876975e+02
```

```
##      size      mu  
## 4.876975e+02 1.146154e+00  
## (1.101478e+03) (9.400683e-02)
```

```
#For n_seeds  
n_seeds.parms2 <- fitdistr(n_seeds.dist, "negative binomial")  
n_seeds.parms2 # size = 5.8823340
```

```
##      size      mu  
## 5.8823340 234.0692308  
## ( 0.7302999) ( 8.5701348)
```

Specifying families:

```
fam <- c(1,2,3) # specifies the one-parameter exponential families for the nodes  
famlist <- list(fam.bernoulli(),  
               fam.truncated.negative.binomial(size=4.876975e+02,  
                                               truncation = 0),  
               fam.poisson())  
# Here: Bernoulli, truncated.negative.binomial, poisson  
vars[fam == 1]
```

```
## [1] "seedpods_01"
```

```
vars[fam == 2]
```

```
## [1] "n_seedpods"
```

```
vars[fam == 3]
```

```
## [1] "n_seeds"
```

```
# creating the fitness variable of interest
seeds <- grep("n_seeds", as.character(redata$varb))
seeds <- is.element(seq(along = redata$varb), seeds)
redata <- data.frame(redata, seeds = as.integer(seeds))
names(redata)
```

```
## [1] "id"          "FFD_corr"    "temp"        "ros_area"    "P"
## [6] "F"           "fitness_01"  "ros_area_log" "FFD_corr_std" "ros_area_std"
## [11] "n_seeds_rel" "varb"        "resp"        "root"        "seeds"
```

Center the predictors (I think we do not need to use standardized predictors in Aster models, but centering the predictors helps the model to converge):

```
redata$temp_c<-redata$temp-mean(redata$temp)
redata$FFD_c<-redata$FFD_corr-mean(redata$FFD_corr)
redata$ros_area_c<-redata$ros_area-mean(redata$ros_area)
```

Fitting the full Aster model:

```
# Full Aster model with seeds as the interaction for all fixed variables
aster1 <- aster(resp ~ varb+seeds:(temp_c*FFD_c*P*F+ros_area_c),
               pred, fam, famlist=famlist, varb, id, root, data = redata)
# Include varb in the formula to get a different "intercept"
# for each node in the graph (i.e. one for fitness_01 and one for n_seeds)
# The categorical variable varb gets turned into as many dummy
# variables as there are nodes in the graph, one is dropped, and the
# "intercept" dummy variable (all components = 1) is added; the
# effect is to provide a different intercept for each node.
# Include interaction seeds: with all other variables
# Seehttps://groups.google.com/g/aster-analysis-user-group/c/xwanCX8VOWQ/m/AcTnF3E3BAAJ
# and code there.
# According to https://www.stat.umn.edu/geyer/8931aster/slides/s1.pdf#page=55
# (slides 55-71), we do this to have separate coefficients for the "layer"
# of the graph consisting of terminal nodes (as in aster models, regression
# coefficients "for" a node of the graph also influence all "earlier" nodes
# To estimate the effects of each predictor on lifetime fitness, each predictor
# was fit at the level of total seed set in the model (Shaw et al. 2008)
```

```
summary(aster1,
       show.graph = TRUE, # Table about the graph structure in the printout
       info.tol = 1e-11) # To help the model to converge
```

```
##
## Call:
## aster.formula(formula = resp ~ varb + seeds:(temp_c * FFD_c *
## P * F + ros_area_c), pred = pred, fam = fam, varvar = varb,
## idvar = id, root = root, data = redata, famlist = famlist)
##
##
## Graphical Model:
## variable predecessor
## seedpods_01 root
```

```
## n_seedpods seedpods_01
## n_seeds n_seedpods
## family
## bernoulli
## truncated.negative.binomial(size = 487.6975, truncation = 0)
## poisson
##
## Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.099e+02 1.195e+00 -175.618 <2e-16 ***
## varbn_seeds 2.152e+02 1.201e+00 179.229 <2e-16 ***
## varbseedpods_01 4.351e+02 1.290e+00 337.400 <2e-16 ***
## seeds:temp_c -3.739e-04 6.470e-04 -0.578 0.5634
## seeds:FFD_c -3.164e-04 2.685e-04 -1.179 0.2385
## seeds:P1 3.044e-03 1.915e-03 1.590 0.1119
## seeds:F1 -1.006e-03 2.098e-03 -0.479 0.6316
## seeds:ros_area_c 6.635e-06 2.015e-05 0.329 0.7419
## seeds:temp_c:FFD_c 9.811e-05 5.105e-05 1.922 0.0546 .
## seeds:temp_c:P1 2.014e-03 9.076e-04 2.218 0.0265 *
## seeds:FFD_c:P1 4.876e-04 3.579e-04 1.362 0.1731
## seeds:temp_c:F1 -1.529e-04 9.775e-04 -0.156 0.8757
## seeds:FFD_c:F1 7.844e-06 3.985e-04 0.020 0.9843
## seeds:P1:F1 -2.164e-03 2.875e-03 -0.753 0.4517
## seeds:temp_c:FFD_c:P1 7.828e-05 7.045e-05 1.111 0.2666
## seeds:temp_c:FFD_c:F1 -3.024e-05 8.934e-05 -0.338 0.7350
## seeds:temp_c:P1:F1 4.427e-04 1.315e-03 0.337 0.7365
## seeds:FFD_c:P1:F1 7.098e-05 5.279e-04 0.134 0.8930
## seeds:temp_c:FFD_c:P1:F1 -5.301e-06 1.124e-04 -0.047 0.9624
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(see <https://groups.google.com/g/aster-analysis-user-group/c/xXXR0ez8aAA/m/BrNE6whiIEQJ>)

Test interactions

The Aster manual advises to not look at significances in the summary, but instead compare models with a likelihood ratio test to evaluate significance of each term. The comparison between two or more models . . . will only be valid if they are (1) fitted to the same dataset, (2) models are nested, (3) models are of the same type (all conditional or all unconditional), (4) have the same dependence graph and exponential families. None of this is currently checked.

Determine the significance of fixed effects and interactions by comparing submodels without the term of interest to the full model using LRTs.

Use LRTs comparing submodels to fuller models to test each predictor of interest.

Full model with different syntax, making it easier to remove terms:

```
aster1 <- aster(resp ~ varb+seeds:temp_c+seeds:FFD_c+seeds:P+seeds:F+
  seeds:ros_area_c+seeds:temp_c:FFD_c+seeds:temp_c:P+
  seeds:FFD_c:P+seeds:temp_c:F+seeds:FFD_c:F+seeds:P:F+
  seeds:temp_c:FFD_c:P+seeds:temp_c:FFD_c:F+
  seeds:temp_c:P:F+seeds:FFD_c:P:F+seeds:temp_c:FFD_c:P:F,
  pred, fam, famlist=famlist, varb, id, root, data = redata)
```


Submodels:

Testing the significance of seeds:temp_c:FFD_c:P:F

```
aster1_1 <- aster(resp ~ varb+seeds:(temp_c*FFD_c*P*F+ros_area_c)-
  seeds:temp_c:FFD_c:P:F,
  pred, fam, famlist=famlist, varb, id, root, data = redata)
```

```
anova(aster1_1,aster1)
```

```
## Analysis of Deviance Table
```

```
##
```

```
## Model 1: resp ~ varb + seeds:(temp_c * FFD_c * P * F + ros_area_c) - seeds:temp_c:FFD_c:P:F
```

```
## Model 2: c("resp ~ varb + seeds:temp_c + seeds:FFD_c + seeds:P + seeds:F + ", "      seeds:ros_area_c
```

```
##   Model Df Model Dev Df  Deviance P(>|Chi|)
```

```
## 1      18    260597
```

```
## 2      19    260597  1 0.0022239    0.9624
```

Testing the significance of seeds:FFD_c:P:F

```
aster1_2 <- aster(resp ~ varb+seeds:(temp_c*FFD_c*P*F+ros_area_c)-
  seeds:temp_c:FFD_c:P:F-
  seeds:FFD_c:P:F,
  pred, fam, famlist=famlist, varb, id, root, data = redata)
```

```
anova(aster1_2,aster1_1)
```

```
## Analysis of Deviance Table
```

```
##
```

```
## Model 1: c("resp ~ varb + seeds:(temp_c * FFD_c * P * F + ros_area_c) - seeds:temp_c:FFD_c:P:F - ",
```

```
## Model 2: resp ~ varb + seeds:(temp_c * FFD_c * P * F + ros_area_c) - seeds:temp_c:FFD_c:P:F
```

```
##   Model Df Model Dev Df Deviance P(>|Chi|)
```

```
## 1      17    260597
```

```
## 2      18    260597  1 0.021013    0.8847
```

Testing the significance of seeds:temp_c:P:F

```
aster1_3 <- aster(resp ~ varb+seeds:(temp_c*FFD_c*P*F+ros_area_c)-
  seeds:temp_c:FFD_c:P:F-
  seeds:FFD_c:P:F-
  seeds:temp_c:P:F,
  pred, fam, famlist=famlist, varb, id, root, data = redata)
```

```
anova(aster1_3,aster1_2)
```

```
## Analysis of Deviance Table
```

```
##
```

```
## Model 1: resp ~ varb + seeds:(temp_c * FFD_c * P * F + ros_area_c) - seeds:temp_c:FFD_c:P:F -
```

```
## Model 2:      seeds:FFD_c:P:F - seeds:temp_c:P:F
```

```
## Model 1: resp ~ varb + seeds:(temp_c * FFD_c * P * F + ros_area_c) - seeds:temp_c:FFD_c:P:F -
```

```
## Model 2:      seeds:FFD_c:P:F
```

```
##   Model Df Model Dev Df Deviance P(>|Chi|)
```

```
## 1      16    260597
```

```
## 2      17    260597  1 0.22244    0.6372
```

Testing the significance of seeds:temp_c:FFD_c:F

```
aster1_4 <- aster(resp ~ varb+seeds:(temp_c*FFD_c*P*F+ros_area_c)-
  seeds:temp_c:FFD_c:P:F-
  seeds:FFD_c:P:F-
  seeds:temp_c:P:F-
  seeds:temp_c:FFD_c:F,
  pred, fam, famlist=famlist, varb, id, root, data = redata)
```

```
anova(aster1_4,aster1_3)
```

Analysis of Deviance Table

##

Model 1: resp ~ varb + seeds:(temp_c * FFD_c * P * F + ros_area_c) - seeds:temp_c:FFD_c:P:F -

Model 2: seeds:FFD_c:P:F - seeds:temp_c:P:F - seeds:temp_c:FFD_c:F

Model 1: resp ~ varb + seeds:(temp_c * FFD_c * P * F + ros_area_c) - seeds:temp_c:FFD_c:P:F -

Model 2: seeds:FFD_c:P:F - seeds:temp_c:P:F

Model Df Model Dev Df Deviance P(>|Chi|)

1 15 260596

2 16 260597 1 0.49404 0.4821

Testing the significance of seeds:temp_c:FFD_c:P

```
aster1_5 <- aster(resp ~ varb+seeds:(temp_c*FFD_c*P*F+ros_area_c)-
  seeds:temp_c:FFD_c:P:F-
  seeds:FFD_c:P:F-
  seeds:temp_c:P:F-
  seeds:temp_c:FFD_c:F-
  seeds:temp_c:FFD_c:P,
  pred, fam, famlist=famlist, varb, id, root, data = redata)
```

```
anova(aster1_5,aster1_4)
```

Analysis of Deviance Table

##

Model 1: c("resp ~ varb + seeds:(temp_c * FFD_c * P * F + ros_area_c) - seeds:temp_c:FFD_c:P:F - ",

Model 2: c("resp ~ varb + seeds:(temp_c * FFD_c * P * F + ros_area_c) - seeds:temp_c:FFD_c:P:F - ",

Model Df Model Dev Df Deviance P(>|Chi|)

1 14 260595

2 15 260596 1 1.7131 0.1906

Testing the significance of seeds:P:F

```
aster1_6 <- aster(resp ~ varb+seeds:(temp_c*FFD_c*P*F+ros_area_c)-
  seeds:temp_c:FFD_c:P:F-
  seeds:FFD_c:P:F-
  seeds:temp_c:P:F-
  seeds:temp_c:FFD_c:F-
  seeds:temp_c:FFD_c:P-
  seeds:P:F,
  pred, fam, famlist=famlist, varb, id, root, data = redata)
```

```
anova(aster1_6,aster1_5)
```

```
## Analysis of Deviance Table
##
## Model 1: resp ~ varb + seeds:(temp_c * FFD_c * P * F + ros_area_c) - seeds:temp_c:FFD_c:P:F -
## Model 2:      seeds:FFD_c:P:F - seeds:temp_c:P:F - seeds:temp_c:FFD_c:F -
## Model 1:      seeds:temp_c:FFD_c:P - seeds:P:F
## Model 2: resp ~ varb + seeds:(temp_c * FFD_c * P * F + ros_area_c) - seeds:temp_c:FFD_c:P:F -
## Model 1:      seeds:FFD_c:P:F - seeds:temp_c:P:F - seeds:temp_c:FFD_c:F -
## Model 2:      seeds:temp_c:FFD_c:P
##   Model Df Model Dev Df Deviance P(>|Chi|)
## 1      13    260594
## 2      14    260595  1  0.87288    0.3502
```

Testing the significance of seeds:FFD_c:F

```
aster1_7 <- aster(resp ~ varb+seeds:(temp_c*FFD_c*P*F+ros_area_c)-
  seeds:temp_c:FFD_c:P:F-
  seeds:FFD_c:P:F-
  seeds:temp_c:P:F-
  seeds:temp_c:FFD_c:F-
  seeds:temp_c:FFD_c:P-
  seeds:P:F-
  seeds:FFD_c:F,
  pred, fam, famlist=famlist, varb, id, root, data = redata)
```

```
anova(aster1_7,aster1_6)
```

```
## Analysis of Deviance Table
##
## Model 1: resp ~ varb + seeds:(temp_c * FFD_c * P * F + ros_area_c) - seeds:temp_c:FFD_c:P:F -
## Model 2:      seeds:FFD_c:P:F - seeds:temp_c:P:F - seeds:temp_c:FFD_c:F -
## Model 1:      seeds:temp_c:FFD_c:P - seeds:P:F - seeds:FFD_c:F
## Model 2: resp ~ varb + seeds:(temp_c * FFD_c * P * F + ros_area_c) - seeds:temp_c:FFD_c:P:F -
## Model 1:      seeds:FFD_c:P:F - seeds:temp_c:P:F - seeds:temp_c:FFD_c:F -
## Model 2:      seeds:temp_c:FFD_c:P - seeds:P:F
##   Model Df Model Dev Df Deviance P(>|Chi|)
## 1      12    260594
## 2      13    260594  1 0.093605    0.7596
```

Testing the significance of seeds:temp_c:F

```
aster1_8 <- aster(resp ~ varb+seeds:(temp_c*FFD_c*P*F+ros_area_c)-
  seeds:temp_c:FFD_c:P:F-
  seeds:FFD_c:P:F-
  seeds:temp_c:P:F-
  seeds:temp_c:FFD_c:F-
  seeds:temp_c:FFD_c:P-
  seeds:P:F-
  seeds:FFD_c:F-
  seeds:temp_c:F,
  pred, fam, famlist=famlist, varb, id, root, data = redata)
```

```
anova(aster1_8,aster1_7)
```

```
## Analysis of Deviance Table
##
## Model 1: resp ~ varb + seeds:(temp_c * FFD_c * P * F + ros_area_c) - seeds:temp_c:FFD_c:P:F -
## Model 2:      seeds:FFD_c:P:F - seeds:temp_c:P:F - seeds:temp_c:FFD_c:F -
## Model 1:      seeds:temp_c:FFD_c:P - seeds:P:F - seeds:FFD_c:F - seeds:temp_c:F
## Model 2: resp ~ varb + seeds:(temp_c * FFD_c * P * F + ros_area_c) - seeds:temp_c:FFD_c:P:F -
## Model 1:      seeds:FFD_c:P:F - seeds:temp_c:P:F - seeds:temp_c:FFD_c:F -
## Model 2:      seeds:temp_c:FFD_c:P - seeds:P:F - seeds:FFD_c:F
##   Model Df Model Dev Df Deviance P(>|Chi|)
## 1         11      260594
## 2         12      260594  1 0.046979    0.8284
```

Testing the significance of seeds:FFD_c:P

```
aster1_9 <- aster(resp ~ varb+seeds:(temp_c*FFD_c*P*F+ros_area_c)-
  seeds:temp_c:FFD_c:P:F-
  seeds:FFD_c:P:F-
  seeds:temp_c:P:F-
  seeds:temp_c:FFD_c:F-
  seeds:temp_c:FFD_c:P-
  seeds:P:F-
  seeds:FFD_c:F-
  seeds:temp_c:F-
  seeds:FFD_c:P,
  pred, fam, famlist=famlist, varb, id, root, data = redata)
```

```
anova(aster1_9,aster1_8)
```

```
## Analysis of Deviance Table
##
## Model 1: c("resp ~ varb + seeds:(temp_c * FFD_c * P * F + ros_area_c) - seeds:temp_c:FFD_c:P:F - ",
## Model 2: c("resp ~ varb + seeds:(temp_c * FFD_c * P * F + ros_area_c) - seeds:temp_c:FFD_c:P:F - ",
##   Model Df Model Dev Df Deviance P(>|Chi|)
## 1         10      260591
## 2         11      260594  1  2.9436    0.08622 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Testing the significance of seeds:temp_c:P

```
aster1_10 <- aster(resp ~ varb+seeds:(temp_c*FFD_c*P*F+ros_area_c)-
  seeds:temp_c:FFD_c:P:F-
  seeds:FFD_c:P:F-
  seeds:temp_c:P:F-
  seeds:temp_c:FFD_c:F-
  seeds:temp_c:FFD_c:P-
  seeds:P:F-
  seeds:FFD_c:F-
  seeds:temp_c:F-
```

```

seeds:FFD_c:P-
seeds:temp_c:P,
pred, fam, famlist=famlist, varb, id, root, data = redata)

```

```
anova(aster1_10,aster1_9)
```

```

## Analysis of Deviance Table
##
## Model 1: resp ~ varb + seeds:(temp_c * FFD_c * P * F + ros_area_c) - seeds:temp_c:FFD_c:P:F -
## Model 2:      seeds:FFD_c:P:F - seeds:temp_c:P:F - seeds:temp_c:FFD_c:F -
## Model 1:      seeds:temp_c:FFD_c:P - seeds:P:F - seeds:FFD_c:F - seeds:temp_c:F -
## Model 2:      seeds:FFD_c:P - seeds:temp_c:P
## Model 1: resp ~ varb + seeds:(temp_c * FFD_c * P * F + ros_area_c) - seeds:temp_c:FFD_c:P:F -
## Model 2:      seeds:FFD_c:P:F - seeds:temp_c:P:F - seeds:temp_c:FFD_c:F -
## Model 1:      seeds:temp_c:FFD_c:P - seeds:P:F - seeds:FFD_c:F - seeds:temp_c:F -
## Model 2:      seeds:FFD_c:P
##   Model Df Model Dev Df Deviance P(>|Chi|)
## 1         9    260583
## 2        10    260591  1     7.768  0.005318 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Testing the significance of seeds:temp_c:FFD_c

```

aster1_11 <- aster(resp ~ varb+seeds:(temp_c*FFD_c*P*F+ros_area_c)-
  seeds:temp_c:FFD_c:P:F-
  seeds:FFD_c:P:F-
  seeds:temp_c:P:F-
  seeds:temp_c:FFD_c:F-
  seeds:temp_c:FFD_c:P-
  seeds:P:F-
  seeds:FFD_c:F-
  seeds:temp_c:F-
  seeds:FFD_c:P-
  seeds:temp_c:P-
  seeds:temp_c:FFD_c,
  pred, fam, famlist=famlist, varb, id, root, data = redata)

```

```
anova(aster1_11,aster1_10)
```

```

## Analysis of Deviance Table
##
## Model 1: resp ~ varb + seeds:(temp_c * FFD_c * P * F + ros_area_c) - seeds:temp_c:FFD_c:P:F -
## Model 2:      seeds:FFD_c:P:F - seeds:temp_c:P:F - seeds:temp_c:FFD_c:F -
## Model 1:      seeds:temp_c:FFD_c:P - seeds:P:F - seeds:FFD_c:F - seeds:temp_c:F -
## Model 2:      seeds:FFD_c:P - seeds:temp_c:P - seeds:temp_c:FFD_c
## Model 1: resp ~ varb + seeds:(temp_c * FFD_c * P * F + ros_area_c) - seeds:temp_c:FFD_c:P:F -
## Model 2:      seeds:FFD_c:P:F - seeds:temp_c:P:F - seeds:temp_c:FFD_c:F -
## Model 1:      seeds:temp_c:FFD_c:P - seeds:P:F - seeds:FFD_c:F - seeds:temp_c:F -
## Model 2:      seeds:FFD_c:P - seeds:temp_c:P
##   Model Df Model Dev Df Deviance P(>|Chi|)
## 1         8    260552

```

```
## 2          9      260583  1    31.236 2.285e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Test main effects without interactions

```
aster1_mains <- aster(resp ~ varb+seeds:(temp_c+FFD_c+P+F+ros_area_c),
                      pred, fam, famlist=famlist, varb, id, root, data = redata)
```

```
summary(aster1_mains,
        show.graph = TRUE, # Table about the graph structure in the printout
        info.tol = 1e-11) # To help the model to converge
```

```
##
## Call:
## aster.formula(formula = resp ~ varb + seeds:(temp_c + FFD_c +
##      P + F + ros_area_c), pred = pred, fam = fam, varvar = varb,
##      idvar = id, root = root, data = redata, famlist = famlist)
##
##
## Graphical Model:
## variable      predecessor
## seedpods_01 root
## n_seedpods    seedpods_01
## n_seeds       n_seedpods
## family
## bernoulli
## truncated.negative.binomial(size = 487.6975, truncation = 0)
## poisson
##
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -2.097e+02  1.192e+00 -175.895  <2e-16 ***
## varbn_seeds    2.150e+02  1.198e+00  179.509  <2e-16 ***
## varbseedpods_01 4.344e+02  1.272e+00 341.447  <2e-16 ***
## seeds:temp_c   -2.973e-05  2.773e-04  -0.107    0.915
## seeds:FFD_c    -6.018e-05  1.134e-04  -0.530    0.596
## seeds:P1       9.212e-05  9.437e-04   0.098    0.922
## seeds:F1      -7.296e-04  9.509e-04  -0.767    0.443
## seeds:ros_area_c -9.431e-07  1.707e-05  -0.055    0.956
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Submodels:

Testing the significance of seeds:ros_area_c

```
aster1_12 <- aster(resp ~ varb+seeds:(temp_c+FFD_c+P+F),
                   pred, fam, famlist=famlist, varb, id, root, data = redata)
```

```
anova(aster1_12,aster1_mains)
```

```
## Analysis of Deviance Table
##
## Model 1: resp ~ varb + seeds:(temp_c + FFD_c + P + F)
## Model 2: resp ~ varb + seeds:(temp_c + FFD_c + P + F + ros_area_c)
##   Model Df Model Dev Df Deviance P(>|Chi|)
## 1         7    260552
## 2         8    260552  1 0.003057    0.9559
```

Testing the significance of seeds:F

```
aster1_13 <- aster(resp ~ varb+seeds:(temp_c+FFD_c+P+ros_area_c),
                  pred, fam, famlist=famlist, varb, id, root, data = redata)
```

```
anova(aster1_13,aster1_mains)
```

```
## Analysis of Deviance Table
##
## Model 1: resp ~ varb + seeds:(temp_c + FFD_c + P + ros_area_c)
## Model 2: resp ~ varb + seeds:(temp_c + FFD_c + P + F + ros_area_c)
##   Model Df Model Dev Df Deviance P(>|Chi|)
## 1         7    260551
## 2         8    260552  1 0.59114    0.442
```

Testing the significance of seeds:P

```
aster1_14 <- aster(resp ~ varb+seeds:(temp_c+FFD_c+F+ros_area_c),
                  pred, fam, famlist=famlist, varb, id, root, data = redata)
```

```
anova(aster1_14,aster1_mains)
```

```
## Analysis of Deviance Table
##
## Model 1: resp ~ varb + seeds:(temp_c + FFD_c + F + ros_area_c)
## Model 2: resp ~ varb + seeds:(temp_c + FFD_c + P + F + ros_area_c)
##   Model Df Model Dev Df Deviance P(>|Chi|)
## 1         7    260552
## 2         8    260552  1 0.0095281    0.9222
```

Testing the significance of seeds:FFD_c

```
aster1_15 <- aster(resp ~ varb+seeds:(temp_c+P+F+ros_area_c),
                  pred, fam, famlist=famlist, varb, id, root, data = redata)
```

```
anova(aster1_15,aster1_mains)
```

```
## Analysis of Deviance Table
##
```

```
## Model 1: resp ~ varb + seeds:(temp_c + P + F + ros_area_c)
## Model 2: resp ~ varb + seeds:(temp_c + FFD_c + P + F + ros_area_c)
##   Model Df Model Dev Df Deviance P(>|Chi|)
## 1       7   260551
## 2       8   260552  1  0.28187   0.5955
```

Testing the significance of seeds:temp_c

```
aster1_16 <- aster(resp ~ varb+seeds:(FFD_c+P+F+ros_area_c),
  pred, fam, famlist=famlist, varb, id, root, data = redata)
```

```
anova(aster1_16,aster1_mains)
```

```
## Analysis of Deviance Table
##
## Model 1: resp ~ varb + seeds:(FFD_c + P + F + ros_area_c)
## Model 2: resp ~ varb + seeds:(temp_c + FFD_c + P + F + ros_area_c)
##   Model Df Model Dev Df Deviance P(>|Chi|)
## 1       7   260552
## 2       8   260552  1 0.011492   0.9146
```

Table LRTs

```
table_LRTs<-bind_rows(
  tidy(anova(aster1_16,aster1_mains))%>%mutate(term="seeds:temp_c"),
  tidy(anova(aster1_15,aster1_mains))%>%mutate(term="seeds:FFD_c"),
  tidy(anova(aster1_14,aster1_mains))%>%mutate(term="seeds:P"),
  tidy(anova(aster1_13,aster1_mains))%>%mutate(term="seeds:F"),
  tidy(anova(aster1_12,aster1_mains))%>%mutate(term="seeds:ros_area_c"),
  tidy(anova(aster1_11,aster1_10))%>%mutate(term="seeds:temp_c:FFD_c"),
  tidy(anova(aster1_10,aster1_9))%>%mutate(term="seeds:temp_c:P"),
  tidy(anova(aster1_9,aster1_8))%>%mutate(term="seeds:FFD_c:P"),
  tidy(anova(aster1_8,aster1_7))%>%mutate(term="seeds:temp_c:F"),
  tidy(anova(aster1_7,aster1_6))%>%mutate(term="seeds:FFD_c:F"),
  tidy(anova(aster1_6,aster1_5))%>%mutate(term="seeds:P:F"),
  tidy(anova(aster1_5,aster1_4))%>%mutate(term="seeds:temp_c:FFD_c:P"),
  tidy(anova(aster1_4,aster1_3))%>%mutate(term="seeds:temp_c:FFD_c:F"),
  tidy(anova(aster1_3,aster1_2))%>%mutate(term="seeds:temp_c:P:F"),
  tidy(anova(aster1_2,aster1_1))%>%mutate(term="seeds:FFD_c:P:F"),
  tidy(anova(aster1_1,aster1))%>%mutate(term="seeds:temp_c:FFD_c:P:F")
)
kable(table_LRTs)
```

Model.Df	Model.Dev	df	Deviance	p.value	term
7	260551.7	NA	NA	NA	seeds:temp_c
8	260551.8	1	0.0114923	0.9146286	seeds:temp_c
7	260551.5	NA	NA	NA	seeds:FFD_c
8	260551.8	1	0.2818665	0.5954805	seeds:FFD_c
7	260551.7	NA	NA	NA	seeds:P
8	260551.8	1	0.0095281	0.9222404	seeds:P

Model.Df	Model.Dev	df	Deviance	p.value	term
7	260551.2	NA	NA	NA	seeds:F
8	260551.8	1	0.5911401	0.4419786	seeds:F
7	260551.8	NA	NA	NA	seeds:ros_area_c
8	260551.8	1	0.0030570	0.9559072	seeds:ros_area_c
8	260551.8	NA	NA	NA	seeds:temp_c:FFD_c
9	260583.0	1	31.2363032	0.0000000	seeds:temp_c:FFD_c
9	260583.0	NA	NA	NA	seeds:temp_c:P
10	260590.8	1	7.7679809	0.0053180	seeds:temp_c:P
10	260590.8	NA	NA	NA	seeds:FFD_c:P
11	260593.7	1	2.9436533	0.0862155	seeds:FFD_c:P
11	260593.7	NA	NA	NA	seeds:temp_c:F
12	260593.8	1	0.0469794	0.8284052	seeds:temp_c:F
12	260593.8	NA	NA	NA	seeds:FFD_c:F
13	260593.8	1	0.0936052	0.7596429	seeds:FFD_c:F
13	260593.8	NA	NA	NA	seeds:P:F
14	260594.7	1	0.8728771	0.3501600	seeds:P:F
14	260594.7	NA	NA	NA	seeds:temp_c:FFD_c:P
15	260596.4	1	1.7131425	0.1905782	seeds:temp_c:FFD_c:P
15	260596.4	NA	NA	NA	seeds:temp_c:FFD_c:F
16	260596.9	1	0.4940422	0.4821297	seeds:temp_c:FFD_c:F
16	260596.9	NA	NA	NA	seeds:temp_c:P:F
17	260597.1	1	0.2224422	0.6371853	seeds:temp_c:P:F
17	260597.1	NA	NA	NA	seeds:FFD_c:P:F
18	260597.2	1	0.0210134	0.8847426	seeds:FFD_c:P:F
18	260597.2	NA	NA	NA	seeds:temp_c:FFD_c:P:F
19	260597.2	1	0.0022239	0.9623868	seeds:temp_c:FFD_c:P:F

Significant effects are seeds:temp_c:FFD_c and seeds:temp_c:P. Interaction seeds:FFD_c:P:temp_c is not significant (different from tresults of selection models).

Selection gradients

If we need to calculate selection coefficients (gradients), we can use code in Peschel et al. 2021, and in <https://groups.google.com/g/aster-analysis-user-group/c/6UlcFGBWM4s/m/eseUuoG3hmYJ>

```
# Predict the expected fitness for each individual from the aster model
paster1 <- predict(aster1)
paster1 <- matrix(paster1, nrow = nrow(aster1$x), ncol = ncol(aster1$x))
colnames(paster1) <- colnames(aster1$x)
mufit <- paster1[, grep("n_seeds", colnames(paster1))] # Expected fitness

# Calculate relative fitness (fitness~i~ / mean fitness) for each individual
wmu <- mufit/mean(mufit)

# Fit OLS regression with traits as predictors.
# Coefficients are the selection gradients.
wmout <- lm(wmu ~ temp_c*FFD_c*P*F+ros_area_c,
            bind_cols(subset(ping_20_data,!is.na(n_seeds)&!is.na(ros_area_std)),
                      wmu=wmu)%>%
            mutate(temp_c=temp-mean(temp),
                   FFD_c=FFD_corr-mean(FFD_corr),
```

```

ros_area_c=ros_area-mean(ros_area)))
summary(wmout)

##
## Call:
## lm(formula = wmu ~ temp_c * FFD_c * P * F + ros_area_c, data = bind_cols(subset(ping_20_data,
##   !is.na(n_seeds) & !is.na(ros_area_std))), wmu = wmu) %>% mutate(temp_c = temp -
##   mean(temp), FFD_c = FFD_corr - mean(FFD_corr), ros_area_c = ros_area -
##   mean(ros_area)))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.35653 -0.05528 -0.01184  0.04719  0.54677
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.1844823   0.0174524   67.869 < 2e-16 ***
## temp_c        -0.0658808   0.0100131   -6.579 2.65e-10 ***
## FFD_c         -0.0455064   0.0039526  -11.513 < 2e-16 ***
## P1             0.5728013   0.0294530   19.448 < 2e-16 ***
## F1            -0.0482939   0.0299979   -1.610  0.1087
## ros_area_c     0.0007305   0.0002844    2.569  0.0108 *
## temp_c:FFD_c    0.0057267   0.0003170   18.062 < 2e-16 ***
## temp_c:P1       0.3122013   0.0142979   21.836 < 2e-16 ***
## FFD_c:P1        0.0772016   0.0053000   14.566 < 2e-16 ***
## temp_c:F1      -0.0029020   0.0149322   -0.194  0.8461
## FFD_c:F1        0.0074699   0.0058636    1.274  0.2038
## P1:F1          -0.3891205   0.0443643   -8.771 2.53e-16 ***
## temp_c:FFD_c:P1  0.0152015   0.0006663   22.815 < 2e-16 ***
## temp_c:FFD_c:F1  0.0010159   0.0010279    0.988  0.3239
## temp_c:P1:F1     0.0098814   0.0206436    0.479  0.6326
## FFD_c:P1:F1     -0.0004161   0.0077185   -0.054  0.9570
## temp_c:FFD_c:P1:F1 -0.0058219  0.0013408   -4.342 2.03e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1269 on 256 degrees of freedom
## Multiple R-squared:  0.9296, Adjusted R-squared:  0.9252
## F-statistic: 211.4 on 16 and 256 DF,  p-value: < 2.2e-16

```

Calculating the *correct* standard errors is a bit more complicated but can be done following the code starting on p5 of TR #675(<http://conservancy.umn.edu/bitstream/11299/56394/1/betaTR.pdf>).

I am not sure we need to include this in the paper.

Reduced model without F

```

aster2 <- aster(resp ~ varb+seeds:(temp_c*FFD_c*P+ros_area_c),
  pred, fam, famlist=famlist, varb, id, root, data = redata)

```

```
summary(aster2, show.graph = TRUE, info.tol = 1e-11)
```

```
##
## Call:
## aster.formula(formula = resp ~ varb + seeds:(temp_c * FFD_c *
##     P + ros_area_c), pred = pred, fam = fam, varvar = varb, idvar = id,
##     root = root, data = redata, famlist = famlist)
##
##
## Graphical Model:
##   variable      predecessor
## seedpods_01 root
## n_seedpods  seedpods_01
## n_seeds      n_seedpods
## family
## bernoulli
## truncated.negative.binomial(size = 487.6975, truncation = 0)
## poisson
##
##              Estimate Std. Error  z value Pr(>|z|)
## (Intercept)    -2.098e+02  1.195e+00 -175.653 < 2e-16 ***
## varbn_seeds      2.152e+02  1.200e+00  179.265 < 2e-16 ***
## varbseedpods_01  4.350e+02  1.288e+00  337.872 < 2e-16 ***
## seeds:temp_c     -4.092e-04  4.693e-04   -0.872  0.38319
## seeds:FFD_c      -2.992e-04  1.940e-04   -1.542  0.12297
## seeds:P1          1.850e-03  1.396e-03    1.326  0.18499
## seeds:ros_area_c  9.796e-06  1.979e-05    0.495  0.62069
## seeds:temp_c:FFD_c 8.690e-05  4.022e-05    2.161  0.03072 *
## seeds:temp_c:P1   2.054e-03  6.665e-04    3.082  0.00206 **
## seeds:FFD_c:P1    4.651e-04  2.598e-04    1.791  0.07335 .
## seeds:temp_c:FFD_c:P1 6.926e-05  5.273e-05    1.314  0.18900
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Test some interactions

Submodels:

Testing the significance of seeds:temp_c:FFD_c:P

```
aster2_1 <- aster(resp ~ varb+seeds:(temp_c*FFD_c*P+ros_area_c)-
  seeds:temp_c:FFD_c:P,
  pred, fam, famlist=famlist, varb, id, root, data = redata)
```

```
anova(aster2_1, aster2)
```

```
## Analysis of Deviance Table
##
## Model 1: resp ~ varb + seeds:(temp_c * FFD_c * P + ros_area_c) - seeds:temp_c:FFD_c:P
## Model 2: resp ~ varb + seeds:(temp_c * FFD_c * P + ros_area_c)
##   Model Df Model Dev Df Deviance P(>|Chi|)
## 1      10    260592
## 2      11    260594 1    1.6276    0.202
```

Testing the significance of seeds:FFD_c:P

```
aster2_2 <- aster(resp ~ varb+seeds:(temp_c*FFD_c*P+ros_area_c)-
  seeds:temp_c:FFD_c:P-
  seeds:FFD_c:P,
  pred, fam, famlist=famlist, varb, id, root, data = redata)
```

```
anova(aster2_2,aster2_1)
```

```
## Analysis of Deviance Table
```

```
##
```

```
## Model 1: c("resp ~ varb + seeds:(temp_c * FFD_c * P + ros_area_c) - seeds:temp_c:FFD_c:P - ", " s
```

```
## Model 2: resp ~ varb + seeds:(temp_c * FFD_c * P + ros_area_c) - seeds:temp_c:FFD_c:P
```

```
##   Model Df Model Dev Df Deviance P(>|Chi|)
```

```
## 1         9    260589
```

```
## 2        10    260592 1     2.697    0.1005
```

Significances similar to full model, provide results with reduced model in Appendix (with full table LRTs).

Reduced model with terms included in averaged LMs (USE in main text)

```
aster3 <- aster(resp ~ varb+seeds:(temp_c*FFD_c*P+F+ros_area_c),
  pred, fam, famlist=famlist, varb, id, root, data = redata)
```

```
summary(aster3,show.graph = TRUE,info.tol = 1e-11)
```

```
##
```

```
## Call:
```

```
## aster.formula(formula = resp ~ varb + seeds:(temp_c * FFD_c *
##   P + F + ros_area_c), pred = pred, fam = fam, varvar = varb,
##   idvar = id, root = root, data = redata, famlist = famlist)
```

```
##
```

```
##
```

```
## Graphical Model:
```

```
##   variable      predecessor
```

```
## seedpods_01 root
```

```
## n_seedpods seedpods_01
```

```
## n_seeds      n_seedpods
```

```
## family
```

```
## bernoulli
```

```
## truncated.negative.binomial(size = 487.6975, truncation = 0)
```

```
## poisson
```

```
##
```

```
##               Estimate Std. Error  z value Pr(>|z|)
```

```
## (Intercept)    -2.098e+02  1.195e+00 -175.643 < 2e-16 ***
```

```
## varbn_seeds     2.152e+02  1.200e+00  179.254 < 2e-16 ***
```

```
## varbseedpods_01  4.351e+02  1.288e+00  337.689 < 2e-16 ***
```

```
## seeds:temp_c    -4.128e-04  4.702e-04   -0.878  0.37994
```

```
## seeds:FFD_c     -2.953e-04  1.943e-04   -1.520  0.12852
```

```
## seeds:P1        1.955e-03  1.398e-03    1.399  0.16193
```

```
## seeds:F1          -1.379e-03  1.024e-03   -1.347  0.17798
## seeds:ros_area_c    6.637e-06  1.996e-05    0.333  0.73943
## seeds:temp_c:FFD_c  8.737e-05  3.969e-05    2.201  0.02772 *
## seeds:temp_c:P1     2.128e-03  6.709e-04    3.171  0.00152 **
## seeds:FFD_c:P1      4.853e-04  2.608e-04    1.860  0.06283 .
## seeds:temp_c:FFD_c:P1 7.106e-05  5.241e-05    1.356  0.17514
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Test interactions

Full model with different syntax, making it easier to remove terms:

```
aster3 <- aster(resp ~ varb+seeds:temp_c+seeds:FFD_c+seeds:P+seeds:F+
  seeds:ros_area_c+seeds:temp_c:FFD_c+seeds:temp_c:P+
  seeds:FFD_c:P+seeds:temp_c:FFD_c:P,
  pred, fam, famlist=famlist, varb, id, root, data = redata)
```

Submodels:

Testing the significance of seeds:temp_c:FFD_c:P

```
aster3_1 <- aster(resp ~ varb+seeds:temp_c+seeds:FFD_c+seeds:P+seeds:F+
  seeds:ros_area_c+seeds:temp_c:FFD_c+seeds:temp_c:P+
  seeds:FFD_c:P,
  pred, fam, famlist=famlist, varb, id, root, data = redata)
```

```
anova(aster3_1,aster3)
```

```
## Analysis of Deviance Table
##
## Model 1: resp ~ varb + seeds:temp_c + seeds:FFD_c + seeds:P + seeds:F +
## Model 2:      seeds:ros_area_c + seeds:temp_c:FFD_c + seeds:temp_c:P +
## Model 1:      seeds:FFD_c:P
## Model 2: resp ~ varb + seeds:temp_c + seeds:FFD_c + seeds:P + seeds:F +
## Model 1:      seeds:ros_area_c + seeds:temp_c:FFD_c + seeds:temp_c:P +
## Model 2:      seeds:FFD_c:P + seeds:temp_c:FFD_c:P
##   Model Df Model Dev Df Deviance P(>|Chi|)
## 1      11    260594
## 2      12    260595  1     1.731    0.1883
```

Testing the significance of seeds:FFD_c:P

```
aster3_2 <- aster(resp ~ varb+seeds:temp_c+seeds:FFD_c+seeds:P+seeds:F+
  seeds:ros_area_c+seeds:temp_c:FFD_c+seeds:temp_c:P,
  pred, fam, famlist=famlist, varb, id, root, data = redata)
```

```
anova(aster3_2,aster3_1)
```

```
## Analysis of Deviance Table
##
```

```
## Model 1: c("resp ~ varb + seeds:temp_c + seeds:FFD_c + seeds:P + seeds:F + ", " seeds:ros_area_c +
## Model 2: c("resp ~ varb + seeds:temp_c + seeds:FFD_c + seeds:P + seeds:F + ", " seeds:ros_area_c +
## Model Df Model Dev Df Deviance P(>|Chi|)
## 1      10      260591
## 2      11      260594 1      2.9436 0.08622 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Testing the significance of seeds:temp_c:P

```
aster3_3 <- aster(resp ~ varb+seeds:temp_c+seeds:FFD_c+seeds:P+seeds:F+
  seeds:ros_area_c+seeds:temp_c:FFD_c,
  pred, fam, famlist=famlist, varb, id, root, data = redata)
```

```
anova(aster3_3,aster3_2)
```

```
## Analysis of Deviance Table
##
## Model 1: resp ~ varb + seeds:temp_c + seeds:FFD_c + seeds:P + seeds:F +
## Model 2:      seeds:ros_area_c + seeds:temp_c:FFD_c
## Model 1: resp ~ varb + seeds:temp_c + seeds:FFD_c + seeds:P + seeds:F +
## Model 2:      seeds:ros_area_c + seeds:temp_c:FFD_c + seeds:temp_c:P
## Model Df Model Dev Df Deviance P(>|Chi|)
## 1      9      260583
## 2      10      260591 1      7.768 0.005318 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Testing the significance of seeds:temp_c:FFD_c

```
aster3_4 <- aster(resp ~ varb+seeds:temp_c+seeds:FFD_c+seeds:P+seeds:F+
  seeds:ros_area_c,
  pred, fam, famlist=famlist, varb, id, root, data = redata)
```

```
anova(aster3_4,aster3_3)
```

```
## Analysis of Deviance Table
##
## Model 1: resp ~ varb + seeds:temp_c + seeds:FFD_c + seeds:P + seeds:F +
## Model 2:      seeds:ros_area_c
## Model 1: resp ~ varb + seeds:temp_c + seeds:FFD_c + seeds:P + seeds:F +
## Model 2:      seeds:ros_area_c + seeds:temp_c:FFD_c
## Model Df Model Dev Df Deviance P(>|Chi|)
## 1      8      260552
## 2      9      260583 1      31.236 2.285e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Test main effects without interactions

```
aster3_mains <- aster(resp ~ varb+seeds:(temp_c+FFD_c+P+F+ros_area_c),
  pred, fam, famlist=famlist, varb, id, root, data = redata)
```

```
summary(aster3_mains,
  show.graph = TRUE, # Table about the graph structure in the printout
  info.tol = 1e-11) # To help the model to converge
```

```
##
## Call:
## aster.formula(formula = resp ~ varb + seeds:(temp_c + FFD_c +
##   P + F + ros_area_c), pred = pred, fam = fam, varvar = varb,
##   idvar = id, root = root, data = redata, famlist = famlist)
##
##
## Graphical Model:
##   variable      predecessor
## seedpods_01 root
## n_seedpods seedpods_01
## n_seeds     n_seedpods
## family
## bernoulli
## truncated.negative.binomial(size = 487.6975, truncation = 0)
## poisson
##
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -2.097e+02  1.192e+00 -175.895  <2e-16 ***
## varbn_seeds     2.150e+02  1.198e+00  179.509  <2e-16 ***
## varbseedpods_01  4.344e+02  1.272e+00  341.447  <2e-16 ***
## seeds:temp_c    -2.973e-05  2.773e-04  -0.107    0.915
## seeds:FFD_c     -6.018e-05  1.134e-04  -0.530    0.596
## seeds:P1        9.212e-05  9.437e-04   0.098    0.922
## seeds:F1       -7.296e-04  9.509e-04  -0.767    0.443
## seeds:ros_area_c -9.431e-07  1.707e-05  -0.055    0.956
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Submodels:

Testing the significance of seeds:ros_area_c

```
aster3_5 <- aster(resp ~ varb+seeds:(temp_c+FFD_c+P+F),
  pred, fam, famlist=famlist, varb, id, root, data = redata)
```

```
anova(aster3_5,aster3_mains)
```

```
## Analysis of Deviance Table
##
## Model 1: resp ~ varb + seeds:(temp_c + FFD_c + P + F)
## Model 2: resp ~ varb + seeds:(temp_c + FFD_c + P + F + ros_area_c)
##   Model Df Model Dev Df Deviance P(>|Chi|)
## 1       7    260552
## 2       8    260552  1 0.003057    0.9559
```

Testing the significance of seeds:F

```
aster3_6 <- aster(resp ~ varb+seeds:(temp_c+FFD_c+P+ros_area),  
                  pred, fam, famlist=famlist, varb, id, root, data = redata)
```

```
anova(aster3_6,aster3_mains)
```

```
## Analysis of Deviance Table  
##  
## Model 1: resp ~ varb + seeds:(temp_c + FFD_c + P + ros_area)  
## Model 2: resp ~ varb + seeds:(temp_c + FFD_c + P + F + ros_area_c)  
##   Model Df Model Dev Df Deviance P(>|Chi|)  
## 1       7    260551  
## 2       8    260552  1  0.59114    0.442
```

Testing the significance of seeds:P

```
aster3_7 <- aster(resp ~ varb+seeds:(temp_c+FFD_c+F+ros_area),  
                  pred, fam, famlist=famlist, varb, id, root, data = redata)
```

```
anova(aster3_7,aster3_mains)
```

```
## Analysis of Deviance Table  
##  
## Model 1: resp ~ varb + seeds:(temp_c + FFD_c + F + ros_area)  
## Model 2: resp ~ varb + seeds:(temp_c + FFD_c + P + F + ros_area_c)  
##   Model Df Model Dev Df Deviance P(>|Chi|)  
## 1       7    260552  
## 2       8    260552  1 0.0095281    0.9222
```

Testing the significance of seeds:FFD_c

```
aster3_8 <- aster(resp ~ varb+seeds:(temp_c+P+F+ros_area),  
                  pred, fam, famlist=famlist, varb, id, root, data = redata)
```

```
anova(aster3_8,aster3_mains)
```

```
## Analysis of Deviance Table  
##  
## Model 1: resp ~ varb + seeds:(temp_c + P + F + ros_area)  
## Model 2: resp ~ varb + seeds:(temp_c + FFD_c + P + F + ros_area_c)  
##   Model Df Model Dev Df Deviance P(>|Chi|)  
## 1       7    260551  
## 2       8    260552  1  0.28187    0.5955
```

Testing the significance of seeds:temp_c

```
aster3_9 <- aster(resp ~ varb+seeds:(FFD_c+P+F+ros_area),  
                  pred, fam, famlist=famlist, varb, id, root, data = redata)
```



```
anova(aster3_9,aster3_mains)
```

```
## Analysis of Deviance Table
##
## Model 1: resp ~ varb + seeds:(FFD_c + P + F + ros_area)
## Model 2: resp ~ varb + seeds:(temp_c + FFD_c + P + F + ros_area_c)
##   Model Df Model Dev Df Deviance P(>|Chi|)
## 1       7    260552
## 2       8    260552  1 0.011492    0.9146
```

Predictions

seeds:temp_c:P:

```
# make prediction df
aster_predict1 <- data.frame(
  expand_grid(temp_c = seq(from = -5, to = 13, length.out = 100),
    P = as.factor(c(0,1))),
  FFD_c = 0, # centered so mean=0
  ros_area_c = 0, # centered so mean=0
  #P = as.factor(0), # no pollination
  F = as.factor(0),
  seedpods_01 = 1,
  n_seedpods = 1,
  n_seeds = 0,
  root = 1
)

# reshape data to long format
aster_predict1_long <- reshape(as.data.frame(aster_predict1),
  varying = list(vars), direction = "long",
  timevar = "varb", times = as.factor(vars),
  v.names = "resp")

# add artifice
aster_predict1_long$seeds <- as.integer(ifelse(
  aster_predict1_long$varb == "n_seeds", 1, 0))

aster_predict1.df <- aster_predict1_long

# fixed model predictions
aster3.p1 <- predict(aster3, newdata = aster_predict1_long,
  varvar = varb, idvar = id, root = root,
  info.tol = 1e-11, se.fit = T)

aster_predict1.df$fit <- aster3.p1$fit
aster_predict1.df$se <- aster3.p1$se.fit

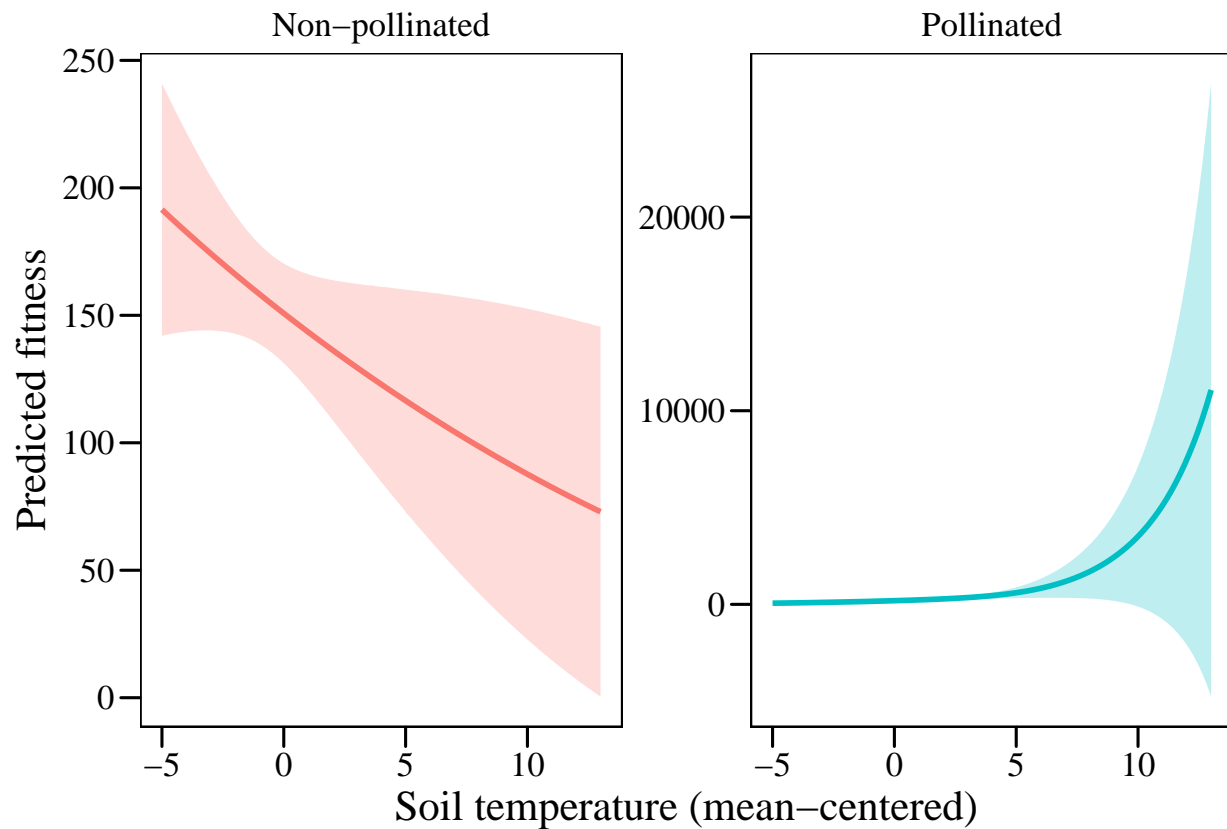
aster_predict1.df.n_seed <- aster_predict1.df %>%
  filter(varb == "n_seeds")
```

```

aster_predict1.df.n_seed<-aster_predict1.df.n_seed%>%
  mutate(ymin=fit-se,ymax=fit+se)

P.labs <- c("Non-pollinated","Pollinated")
names(P.labs) <- c(0,1)
ggplot(aster_predict1.df.n_seed)+
  geom_ribbon(aes(x=temp_c,ymin=ymin,ymax=ymax,fill=P),alpha=0.25)+
  geom_line(aes(x=temp_c,y=fit,color=P),size=1)+
  xlab("Soil temperature (mean-centered)") + ylab("Predicted fitness")+
  my_theme()+
  facet_wrap(~P,scales="free",labeller=labeller(P=P.labs))

```



```

ggsave(filename="output/figures/seeds_temp_c_P_Aster.tiff",device="tiff",
        width=16,height=8,units="cm",dpi=300,compression="lzw")

```

For non-pollinated plants, fitness is higher at cold soil temperatures, but for pollinated plants, fitness is higher at warm soil temperatures.

```

aster_predict1.df.n_seed%>%
  filter(temp_c==min(temp_c)|temp_c==max(temp_c))%>%
  dplyr::select(temp_c,P,fit)

```

```

##           temp_c P      fit
## 1.n_seeds    -5  0  191.46052

```

```
## 2.n_seeds      -5 1      59.92691
## 199.n_seeds    13 0      72.93292
## 200.n_seeds    13 1 11071.43721
```

seeds:temp_c:FFD_c:

```
# make prediction df
aster_predict2 <- data.frame(
  expand_grid(temp_c = seq(from = -5, to = 13, length.out = 100),
             FFD_c = seq(from = -29, to = 21, length.out = 100)),
  FFD_c = 0, # centered so mean=0
  ros_area_c = 0, # centered so mean=0
  P = as.factor(0), # no pollination
  F = as.factor(0),
  seedpods_01 = 1,
  n_seedpods = 1,
  n_seeds = 0,
  root = 1
)

# reshape data to long format
aster_predict2_long <- reshape(as.data.frame(aster_predict2),
                              varying = list(vars), direction = "long",
                              timevar = "varb", times = as.factor(vars),
                              v.names = "resp")

# add artifice
aster_predict2_long$seeds <- as.integer(ifelse(
  aster_predict2_long$varb == "n_seeds", 1, 0))

aster_predict2.df <- aster_predict2_long

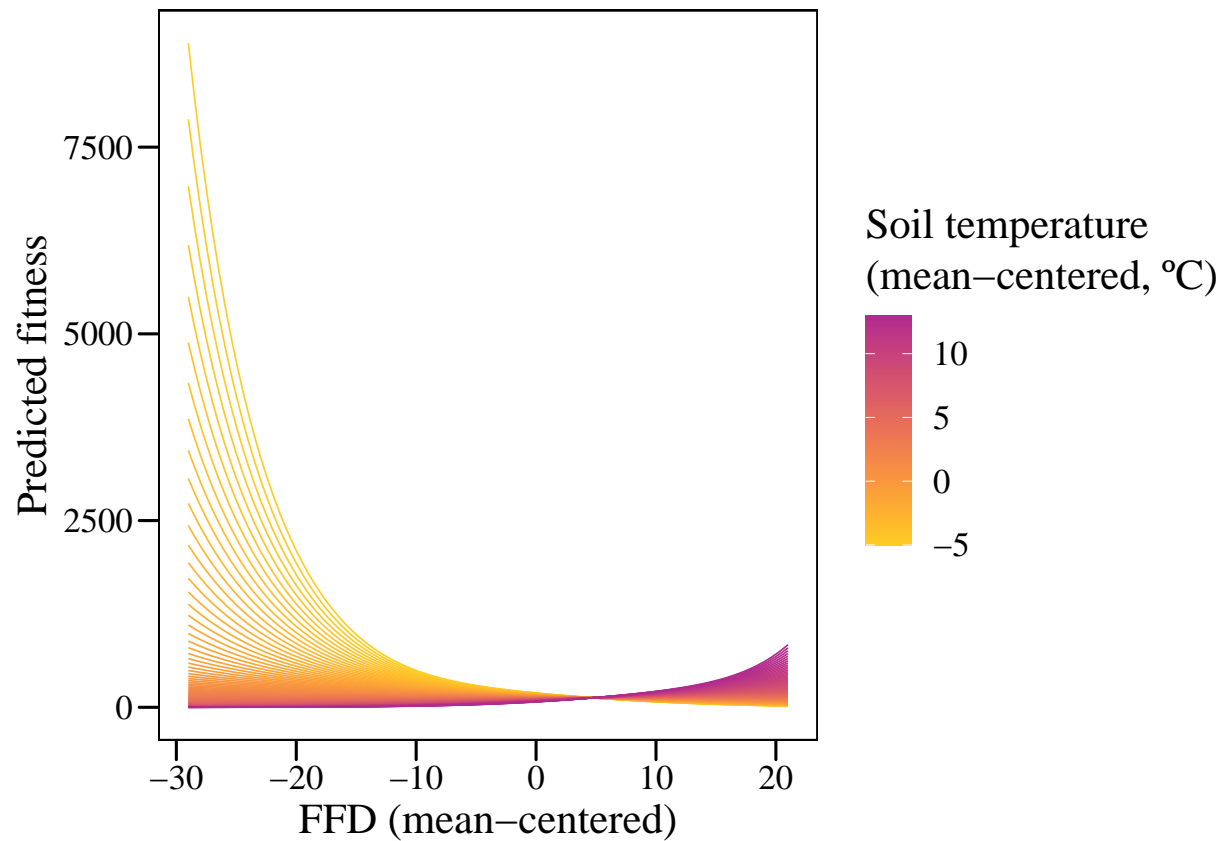
# fixed model predictions
aster3.p2 <- predict(aster3, newdata = aster_predict2_long,
                    varvar = varb, idvar = id, root = root,
                    info.tol = 1e-11, se.fit = T)

aster_predict2.df$fit <- aster3.p2$fit
aster_predict2.df$se <- aster3.p2$se.fit

aster_predict2.df.n_seed <- aster_predict2.df %>%
  filter(varb == "n_seeds")

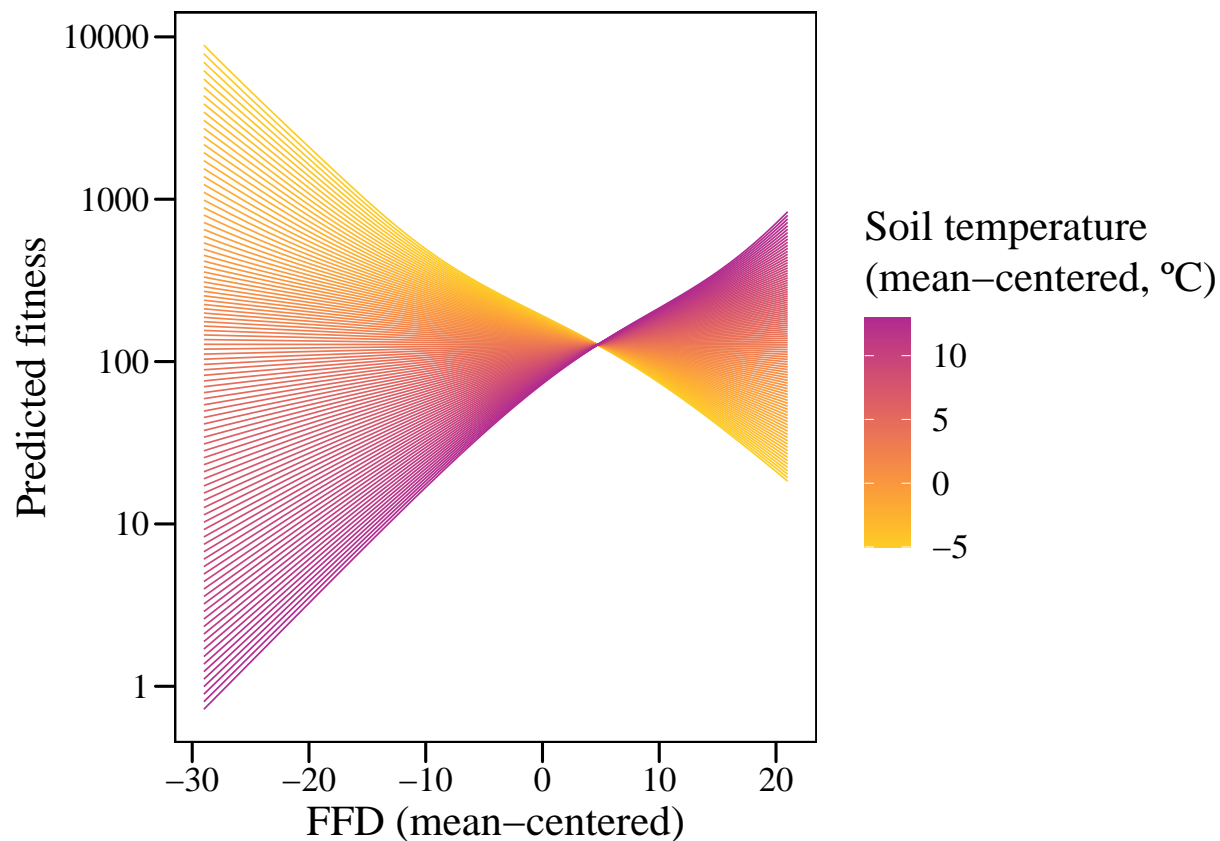
aster_predict2.df.n_seed <- aster_predict2.df.n_seed %>%
  mutate(ymin=fit-se, ymax=fit+se)

# Fitness in normal scale
ggplot(aster_predict2.df.n_seed, aes(x=FFD_c, y=fit, group=as.factor(temp_c)))+
  geom_line(aes(x=FFD_c, y=fit, color=temp_c), size=0.3)+
  xlab("FFD (mean-centered)") + ylab("Predicted fitness")+
  my_theme_legend()+labs(color="Soil temperature\n(mean-centered, °C)") +
  scale_colour_gradientn(colours = plasma(n=100, begin=0.4, end=0.9, direction=-1))
```



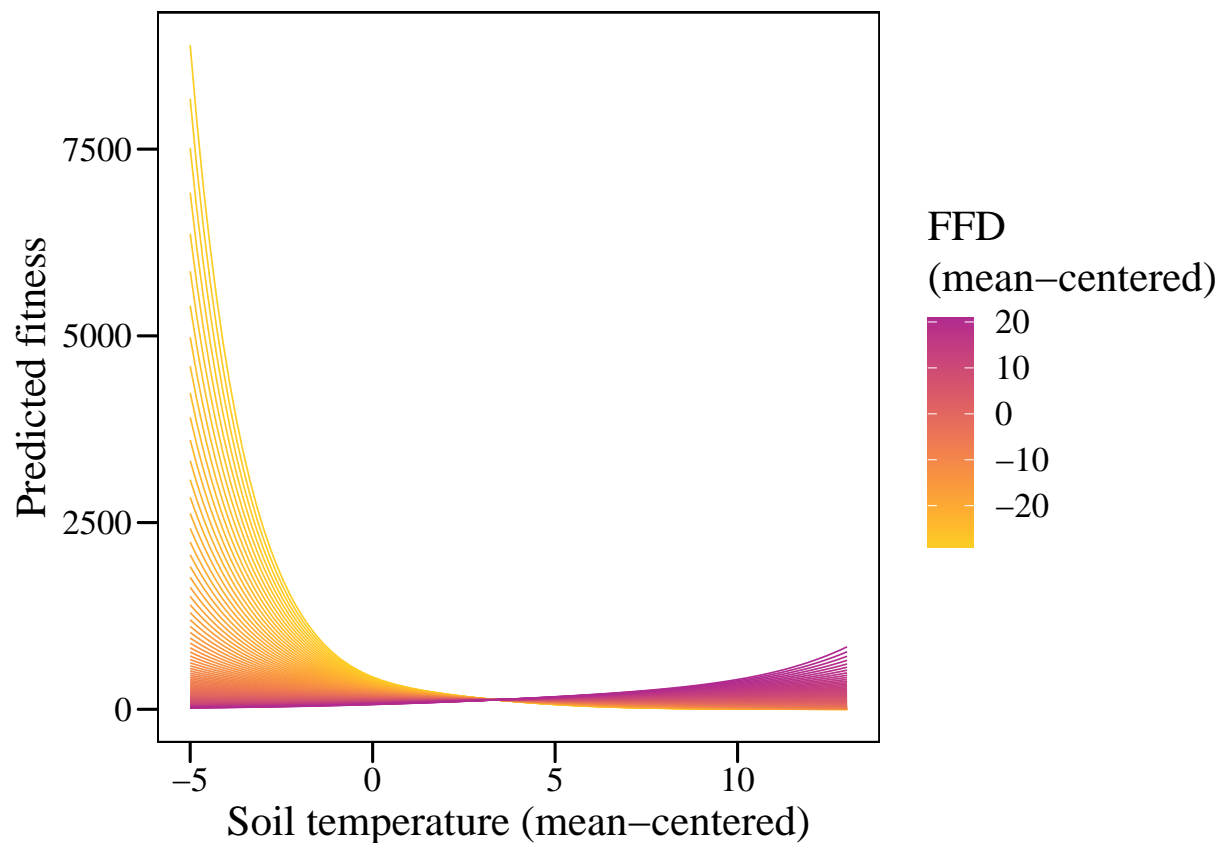
```
ggsave(filename="output/figures/seeds_temp_c_FFD_c_Aster_1.tiff",device="tiff",
        width=15,height=8,units="cm",dpi=300,compression="lzw")

# Fitness in log10 scale
ggplot(aster_predict2.df.n_seed,aes(x=FFD_c,y=fit,group=as.factor(temp_c)))+
  geom_line(aes(x=FFD_c,y=fit,color=temp_c),size=0.3)+
  xlab("FFD (mean-centered)") + ylab("Predicted fitness") +
  my_theme_legend() + scale_y_log10() +
  labs(color="Soil temperature\n(mean-centered, °C)") +
  scale_colour_gradientn(colours = plasma(n=100,begin=0.4,end=0.9,direction=-1))
```



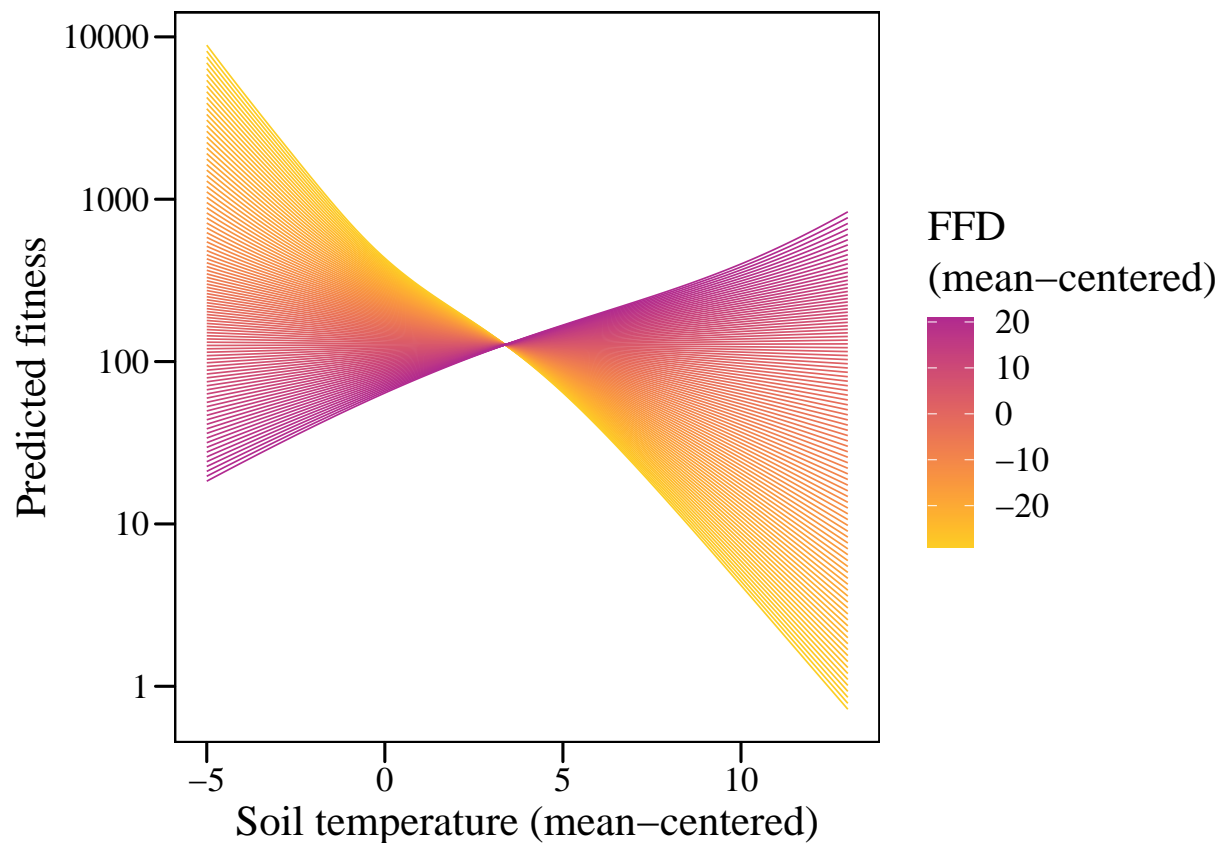
```
ggsave(filename="output/figures/seeds_temp_c_FFD_c_Aster_1_log.tiff",device="tiff",
        width=15,height=8,units="cm",dpi=300,compression="lzw")

# Fitness in normal scale
ggplot(aster_predict2.df.n_seed,aes(x=temp_c,y=fit,group=as.factor(FFD_c)))+
  geom_line(aes(x=temp_c,y=fit,color=FFD_c),size=0.3)+
  xlab("Soil temperature (mean-centered)") + ylab("Predicted fitness")+
  my_theme_legend()+labs(color="FFD\n(mean-centered)")+
  scale_colour_gradientn(colours = plasma(n=100,begin=0.4,end=0.9,direction=-1))
```



```
ggsave(filename="output/figures/seeds_temp_c_FFD_c_Aster_2.tiff",device="tiff",
        width=15,height=8,units="cm",dpi=300,compression="lzw")

# Fitness in log10 scale
ggplot(aster_predict2.df.n_seed,aes(x=temp_c,y=fit,group=as.factor(FFD_c)))+
  geom_line(aes(x=temp_c,y=fit,color=FFD_c),size=0.3)+
  xlab("Soil temperature (mean-centered)") + ylab("Predicted fitness")+
  my_theme_legend()+scale_y_log10()+labs(color="FFD\n(mean-centered)")+
  scale_colour_gradientn(colours = plasma(n=100,begin=0.4,end=0.9,direction=-1))
```



```
ggsave(filename="output/figures/seeds_temp_c_FFD_c_Aster_2_log.tiff",device="tiff",
        width=15,height=8,units="cm",dpi=300,compression="lzw")
```

It is much better to flower early at cold soil temperatures, but not at warm soil temperatures, where it is slightly better to flower late.

```
aster_predict2.df.n_seed%>%
  dplyr::select(temp_c,FFD_c,fit)%>%
  filter(temp_c>3.2&temp_c<3.8)
```

##		temp_c	FFD_c	fit
##	4601.n_seeds	3.363636	-29.0000000	127.7777
##	4602.n_seeds	3.363636	-28.4949495	127.7661
##	4603.n_seeds	3.363636	-27.9898990	127.7546
##	4604.n_seeds	3.363636	-27.4848485	127.7430
##	4605.n_seeds	3.363636	-26.9797980	127.7314
##	4606.n_seeds	3.363636	-26.4747475	127.7199
##	4607.n_seeds	3.363636	-25.9696970	127.7083
##	4608.n_seeds	3.363636	-25.4646465	127.6968
##	4609.n_seeds	3.363636	-24.9595960	127.6852
##	4610.n_seeds	3.363636	-24.4545455	127.6737
##	4611.n_seeds	3.363636	-23.9494949	127.6621
##	4612.n_seeds	3.363636	-23.4444444	127.6506
##	4613.n_seeds	3.363636	-22.9393939	127.6390
##	4614.n_seeds	3.363636	-22.4343434	127.6275

```

## 4615.n_seeds 3.363636 -21.9292929 127.6159
## 4616.n_seeds 3.363636 -21.4242424 127.6044
## 4617.n_seeds 3.363636 -20.9191919 127.5928
## 4618.n_seeds 3.363636 -20.4141414 127.5813
## 4619.n_seeds 3.363636 -19.9090909 127.5697
## 4620.n_seeds 3.363636 -19.4040404 127.5582
## 4621.n_seeds 3.363636 -18.8989899 127.5466
## 4622.n_seeds 3.363636 -18.3939394 127.5351
## 4623.n_seeds 3.363636 -17.8888889 127.5235
## 4624.n_seeds 3.363636 -17.3838384 127.5120
## 4625.n_seeds 3.363636 -16.8787879 127.5004
## 4626.n_seeds 3.363636 -16.3737374 127.4889
## 4627.n_seeds 3.363636 -15.8686869 127.4773
## 4628.n_seeds 3.363636 -15.3636364 127.4658
## 4629.n_seeds 3.363636 -14.8585859 127.4543
## 4630.n_seeds 3.363636 -14.3535354 127.4427
## 4631.n_seeds 3.363636 -13.8484848 127.4312
## 4632.n_seeds 3.363636 -13.3434343 127.4196
## 4633.n_seeds 3.363636 -12.8383838 127.4081
## 4634.n_seeds 3.363636 -12.3333333 127.3966
## 4635.n_seeds 3.363636 -11.8282828 127.3850
## 4636.n_seeds 3.363636 -11.3232323 127.3735
## 4637.n_seeds 3.363636 -10.8181818 127.3619
## 4638.n_seeds 3.363636 -10.3131313 127.3504
## 4639.n_seeds 3.363636 -9.8080808 127.3389
## 4640.n_seeds 3.363636 -9.3030303 127.3273
## 4641.n_seeds 3.363636 -8.7979798 127.3158
## 4642.n_seeds 3.363636 -8.2929293 127.3043
## 4643.n_seeds 3.363636 -7.7878788 127.2927
## 4644.n_seeds 3.363636 -7.2828283 127.2812
## 4645.n_seeds 3.363636 -6.7777778 127.2697
## 4646.n_seeds 3.363636 -6.2727273 127.2581
## 4647.n_seeds 3.363636 -5.7676768 127.2466
## 4648.n_seeds 3.363636 -5.2626263 127.2351
## 4649.n_seeds 3.363636 -4.7575758 127.2236
## 4650.n_seeds 3.363636 -4.2525253 127.2120
## 4651.n_seeds 3.363636 -3.7474747 127.2005
## 4652.n_seeds 3.363636 -3.2424242 127.1890
## 4653.n_seeds 3.363636 -2.7373737 127.1775
## 4654.n_seeds 3.363636 -2.2323232 127.1659
## 4655.n_seeds 3.363636 -1.7272727 127.1544
## 4656.n_seeds 3.363636 -1.2222222 127.1429
## 4657.n_seeds 3.363636 -0.7171717 127.1314
## 4658.n_seeds 3.363636 -0.2121212 127.1198
## 4659.n_seeds 3.363636 0.2929293 127.1083
## 4660.n_seeds 3.363636 0.7979798 127.0968
## 4661.n_seeds 3.363636 1.3030303 127.0853
## 4662.n_seeds 3.363636 1.8080808 127.0737
## 4663.n_seeds 3.363636 2.3131313 127.0622
## 4664.n_seeds 3.363636 2.8181818 127.0507
## 4665.n_seeds 3.363636 3.3232323 127.0392
## 4666.n_seeds 3.363636 3.8282828 127.0277
## 4667.n_seeds 3.363636 4.3333333 127.0162
## 4668.n_seeds 3.363636 4.8383838 127.0046

```



```

## 4669.n_seeds 3.363636 5.3434343 126.9931
## 4670.n_seeds 3.363636 5.8484848 126.9816
## 4671.n_seeds 3.363636 6.3535354 126.9701
## 4672.n_seeds 3.363636 6.8585859 126.9586
## 4673.n_seeds 3.363636 7.3636364 126.9471
## 4674.n_seeds 3.363636 7.8686869 126.9356
## 4675.n_seeds 3.363636 8.3737374 126.9240
## 4676.n_seeds 3.363636 8.8787879 126.9125
## 4677.n_seeds 3.363636 9.3838384 126.9010
## 4678.n_seeds 3.363636 9.8888889 126.8895
## 4679.n_seeds 3.363636 10.3939394 126.8780
## 4680.n_seeds 3.363636 10.8989899 126.8665
## 4681.n_seeds 3.363636 11.4040404 126.8550
## 4682.n_seeds 3.363636 11.9090909 126.8435
## 4683.n_seeds 3.363636 12.4141414 126.8320
## 4684.n_seeds 3.363636 12.9191919 126.8205
## 4685.n_seeds 3.363636 13.4242424 126.8090
## 4686.n_seeds 3.363636 13.9292929 126.7975
## 4687.n_seeds 3.363636 14.4343434 126.7860
## 4688.n_seeds 3.363636 14.9393939 126.7745
## 4689.n_seeds 3.363636 15.4444444 126.7629
## 4690.n_seeds 3.363636 15.9494949 126.7514
## 4691.n_seeds 3.363636 16.4545455 126.7399
## 4692.n_seeds 3.363636 16.9595960 126.7284
## 4693.n_seeds 3.363636 17.4646465 126.7169
## 4694.n_seeds 3.363636 17.9696970 126.7054
## 4695.n_seeds 3.363636 18.4747475 126.6939
## 4696.n_seeds 3.363636 18.9797980 126.6824
## 4697.n_seeds 3.363636 19.4848485 126.6710
## 4698.n_seeds 3.363636 19.9898990 126.6595
## 4699.n_seeds 3.363636 20.4949495 126.6480
## 4700.n_seeds 3.363636 21.0000000 126.6365
## 4701.n_seeds 3.545455 -29.0000000 119.3050
## 4702.n_seeds 3.545455 -28.4949495 119.4181
## 4703.n_seeds 3.545455 -27.9898990 119.5313
## 4704.n_seeds 3.545455 -27.4848485 119.6446
## 4705.n_seeds 3.545455 -26.9797980 119.7579
## 4706.n_seeds 3.545455 -26.4747475 119.8713
## 4707.n_seeds 3.545455 -25.9696970 119.9847
## 4708.n_seeds 3.545455 -25.4646465 120.0982
## 4709.n_seeds 3.545455 -24.9595960 120.2118
## 4710.n_seeds 3.545455 -24.4545455 120.3255
## 4711.n_seeds 3.545455 -23.9494949 120.4392
## 4712.n_seeds 3.545455 -23.4444444 120.5530
## 4713.n_seeds 3.545455 -22.9393939 120.6669
## 4714.n_seeds 3.545455 -22.4343434 120.7808
## 4715.n_seeds 3.545455 -21.9292929 120.8948
## 4716.n_seeds 3.545455 -21.4242424 121.0088
## 4717.n_seeds 3.545455 -20.9191919 121.1230
## 4718.n_seeds 3.545455 -20.4141414 121.2372
## 4719.n_seeds 3.545455 -19.9090909 121.3514
## 4720.n_seeds 3.545455 -19.4040404 121.4658
## 4721.n_seeds 3.545455 -18.8989899 121.5802
## 4722.n_seeds 3.545455 -18.3939394 121.6946

```

```

## 4723.n_seeds 3.545455 -17.8888889 121.8092
## 4724.n_seeds 3.545455 -17.3838384 121.9238
## 4725.n_seeds 3.545455 -16.8787879 122.0384
## 4726.n_seeds 3.545455 -16.3737374 122.1532
## 4727.n_seeds 3.545455 -15.8686869 122.2680
## 4728.n_seeds 3.545455 -15.3636364 122.3828
## 4729.n_seeds 3.545455 -14.8585859 122.4978
## 4730.n_seeds 3.545455 -14.3535354 122.6128
## 4731.n_seeds 3.545455 -13.8484848 122.7279
## 4732.n_seeds 3.545455 -13.3434343 122.8430
## 4733.n_seeds 3.545455 -12.8383838 122.9582
## 4734.n_seeds 3.545455 -12.3333333 123.0735
## 4735.n_seeds 3.545455 -11.8282828 123.1889
## 4736.n_seeds 3.545455 -11.3232323 123.3043
## 4737.n_seeds 3.545455 -10.8181818 123.4197
## 4738.n_seeds 3.545455 -10.3131313 123.5353
## 4739.n_seeds 3.545455 -9.8080808 123.6509
## 4740.n_seeds 3.545455 -9.3030303 123.7666
## 4741.n_seeds 3.545455 -8.7979798 123.8824
## 4742.n_seeds 3.545455 -8.2929293 123.9982
## 4743.n_seeds 3.545455 -7.7878788 124.1141
## 4744.n_seeds 3.545455 -7.2828283 124.2300
## 4745.n_seeds 3.545455 -6.7777778 124.3461
## 4746.n_seeds 3.545455 -6.2727273 124.4622
## 4747.n_seeds 3.545455 -5.7676768 124.5783
## 4748.n_seeds 3.545455 -5.2626263 124.6946
## 4749.n_seeds 3.545455 -4.7575758 124.8109
## 4750.n_seeds 3.545455 -4.2525253 124.9272
## 4751.n_seeds 3.545455 -3.7474747 125.0437
## 4752.n_seeds 3.545455 -3.2424242 125.1602
## 4753.n_seeds 3.545455 -2.7373737 125.2767
## 4754.n_seeds 3.545455 -2.2323232 125.3934
## 4755.n_seeds 3.545455 -1.7272727 125.5101
## 4756.n_seeds 3.545455 -1.2222222 125.6269
## 4757.n_seeds 3.545455 -0.7171717 125.7437
## 4758.n_seeds 3.545455 -0.2121212 125.8606
## 4759.n_seeds 3.545455 0.2929293 125.9776
## 4760.n_seeds 3.545455 0.7979798 126.0947
## 4761.n_seeds 3.545455 1.3030303 126.2118
## 4762.n_seeds 3.545455 1.8080808 126.3290
## 4763.n_seeds 3.545455 2.3131313 126.4462
## 4764.n_seeds 3.545455 2.8181818 126.5636
## 4765.n_seeds 3.545455 3.3232323 126.6810
## 4766.n_seeds 3.545455 3.8282828 126.7984
## 4767.n_seeds 3.545455 4.3333333 126.9160
## 4768.n_seeds 3.545455 4.8383838 127.0336
## 4769.n_seeds 3.545455 5.3434343 127.1512
## 4770.n_seeds 3.545455 5.8484848 127.2690
## 4771.n_seeds 3.545455 6.3535354 127.3868
## 4772.n_seeds 3.545455 6.8585859 127.5046
## 4773.n_seeds 3.545455 7.3636364 127.6226
## 4774.n_seeds 3.545455 7.8686869 127.7406
## 4775.n_seeds 3.545455 8.3737374 127.8587
## 4776.n_seeds 3.545455 8.8787879 127.9768

```

```

## 4777.n_seeds 3.545455 9.3838384 128.0951
## 4778.n_seeds 3.545455 9.8888889 128.2133
## 4779.n_seeds 3.545455 10.3939394 128.3317
## 4780.n_seeds 3.545455 10.8989899 128.4501
## 4781.n_seeds 3.545455 11.4040404 128.5686
## 4782.n_seeds 3.545455 11.9090909 128.6872
## 4783.n_seeds 3.545455 12.4141414 128.8058
## 4784.n_seeds 3.545455 12.9191919 128.9245
## 4785.n_seeds 3.545455 13.4242424 129.0433
## 4786.n_seeds 3.545455 13.9292929 129.1621
## 4787.n_seeds 3.545455 14.4343434 129.2811
## 4788.n_seeds 3.545455 14.9393939 129.4000
## 4789.n_seeds 3.545455 15.4444444 129.5191
## 4790.n_seeds 3.545455 15.9494949 129.6382
## 4791.n_seeds 3.545455 16.4545455 129.7574
## 4792.n_seeds 3.545455 16.9595960 129.8767
## 4793.n_seeds 3.545455 17.4646465 129.9960
## 4794.n_seeds 3.545455 17.9696970 130.1154
## 4795.n_seeds 3.545455 18.4747475 130.2348
## 4796.n_seeds 3.545455 18.9797980 130.3544
## 4797.n_seeds 3.545455 19.4848485 130.4740
## 4798.n_seeds 3.545455 19.9898990 130.5937
## 4799.n_seeds 3.545455 20.4949495 130.7134
## 4800.n_seeds 3.545455 21.0000000 130.8332
## 4801.n_seeds 3.727273 -29.0000000 111.1944
## 4802.n_seeds 3.727273 -28.4949495 111.4215
## 4803.n_seeds 3.727273 -27.9898990 111.6489
## 4804.n_seeds 3.727273 -27.4848485 111.8766
## 4805.n_seeds 3.727273 -26.9797980 112.1045
## 4806.n_seeds 3.727273 -26.4747475 112.3328
## 4807.n_seeds 3.727273 -25.9696970 112.5614
## 4808.n_seeds 3.727273 -25.4646465 112.7903
## 4809.n_seeds 3.727273 -24.9595960 113.0194
## 4810.n_seeds 3.727273 -24.4545455 113.2489
## 4811.n_seeds 3.727273 -23.9494949 113.4786
## 4812.n_seeds 3.727273 -23.4444444 113.7086
## 4813.n_seeds 3.727273 -22.9393939 113.9390
## 4814.n_seeds 3.727273 -22.4343434 114.1696
## 4815.n_seeds 3.727273 -21.9292929 114.4005
## 4816.n_seeds 3.727273 -21.4242424 114.6317
## 4817.n_seeds 3.727273 -20.9191919 114.8632
## 4818.n_seeds 3.727273 -20.4141414 115.0950
## 4819.n_seeds 3.727273 -19.9090909 115.3271
## 4820.n_seeds 3.727273 -19.4040404 115.5595
## 4821.n_seeds 3.727273 -18.8989899 115.7922
## 4822.n_seeds 3.727273 -18.3939394 116.0252
## 4823.n_seeds 3.727273 -17.8888889 116.2584
## 4824.n_seeds 3.727273 -17.3838384 116.4920
## 4825.n_seeds 3.727273 -16.8787879 116.7258
## 4826.n_seeds 3.727273 -16.3737374 116.9600
## 4827.n_seeds 3.727273 -15.8686869 117.1944
## 4828.n_seeds 3.727273 -15.3636364 117.4292
## 4829.n_seeds 3.727273 -14.8585859 117.6642
## 4830.n_seeds 3.727273 -14.3535354 117.8996

```

```

## 4831.n_seeds 3.727273 -13.8484848 118.1352
## 4832.n_seeds 3.727273 -13.3434343 118.3711
## 4833.n_seeds 3.727273 -12.8383838 118.6073
## 4834.n_seeds 3.727273 -12.3333333 118.8438
## 4835.n_seeds 3.727273 -11.8282828 119.0807
## 4836.n_seeds 3.727273 -11.3232323 119.3178
## 4837.n_seeds 3.727273 -10.8181818 119.5552
## 4838.n_seeds 3.727273 -10.3131313 119.7929
## 4839.n_seeds 3.727273 -9.8080808 120.0309
## 4840.n_seeds 3.727273 -9.3030303 120.2691
## 4841.n_seeds 3.727273 -8.7979798 120.5077
## 4842.n_seeds 3.727273 -8.2929293 120.7466
## 4843.n_seeds 3.727273 -7.7878788 120.9858
## 4844.n_seeds 3.727273 -7.2828283 121.2253
## 4845.n_seeds 3.727273 -6.7777778 121.4651
## 4846.n_seeds 3.727273 -6.2727273 121.7051
## 4847.n_seeds 3.727273 -5.7676768 121.9455
## 4848.n_seeds 3.727273 -5.2626263 122.1862
## 4849.n_seeds 3.727273 -4.7575758 122.4272
## 4850.n_seeds 3.727273 -4.2525253 122.6684
## 4851.n_seeds 3.727273 -3.7474747 122.9100
## 4852.n_seeds 3.727273 -3.2424242 123.1518
## 4853.n_seeds 3.727273 -2.7373737 123.3940
## 4854.n_seeds 3.727273 -2.2323232 123.6365
## 4855.n_seeds 3.727273 -1.7272727 123.8792
## 4856.n_seeds 3.727273 -1.2222222 124.1223
## 4857.n_seeds 3.727273 -0.7171717 124.3657
## 4858.n_seeds 3.727273 -0.2121212 124.6093
## 4859.n_seeds 3.727273 0.2929293 124.8533
## 4860.n_seeds 3.727273 0.7979798 125.0975
## 4861.n_seeds 3.727273 1.3030303 125.3421
## 4862.n_seeds 3.727273 1.8080808 125.5870
## 4863.n_seeds 3.727273 2.3131313 125.8321
## 4864.n_seeds 3.727273 2.8181818 126.0776
## 4865.n_seeds 3.727273 3.3232323 126.3234
## 4866.n_seeds 3.727273 3.8282828 126.5694
## 4867.n_seeds 3.727273 4.3333333 126.8158
## 4868.n_seeds 3.727273 4.8383838 127.0625
## 4869.n_seeds 3.727273 5.3434343 127.3094
## 4870.n_seeds 3.727273 5.8484848 127.5567
## 4871.n_seeds 3.727273 6.3535354 127.8043
## 4872.n_seeds 3.727273 6.8585859 128.0522
## 4873.n_seeds 3.727273 7.3636364 128.3004
## 4874.n_seeds 3.727273 7.8686869 128.5489
## 4875.n_seeds 3.727273 8.3737374 128.7977
## 4876.n_seeds 3.727273 8.8787879 129.0468
## 4877.n_seeds 3.727273 9.3838384 129.2962
## 4878.n_seeds 3.727273 9.8888889 129.5459
## 4879.n_seeds 3.727273 10.3939394 129.7959
## 4880.n_seeds 3.727273 10.8989899 130.0462
## 4881.n_seeds 3.727273 11.4040404 130.2968
## 4882.n_seeds 3.727273 11.9090909 130.5478
## 4883.n_seeds 3.727273 12.4141414 130.7990
## 4884.n_seeds 3.727273 12.9191919 131.0506

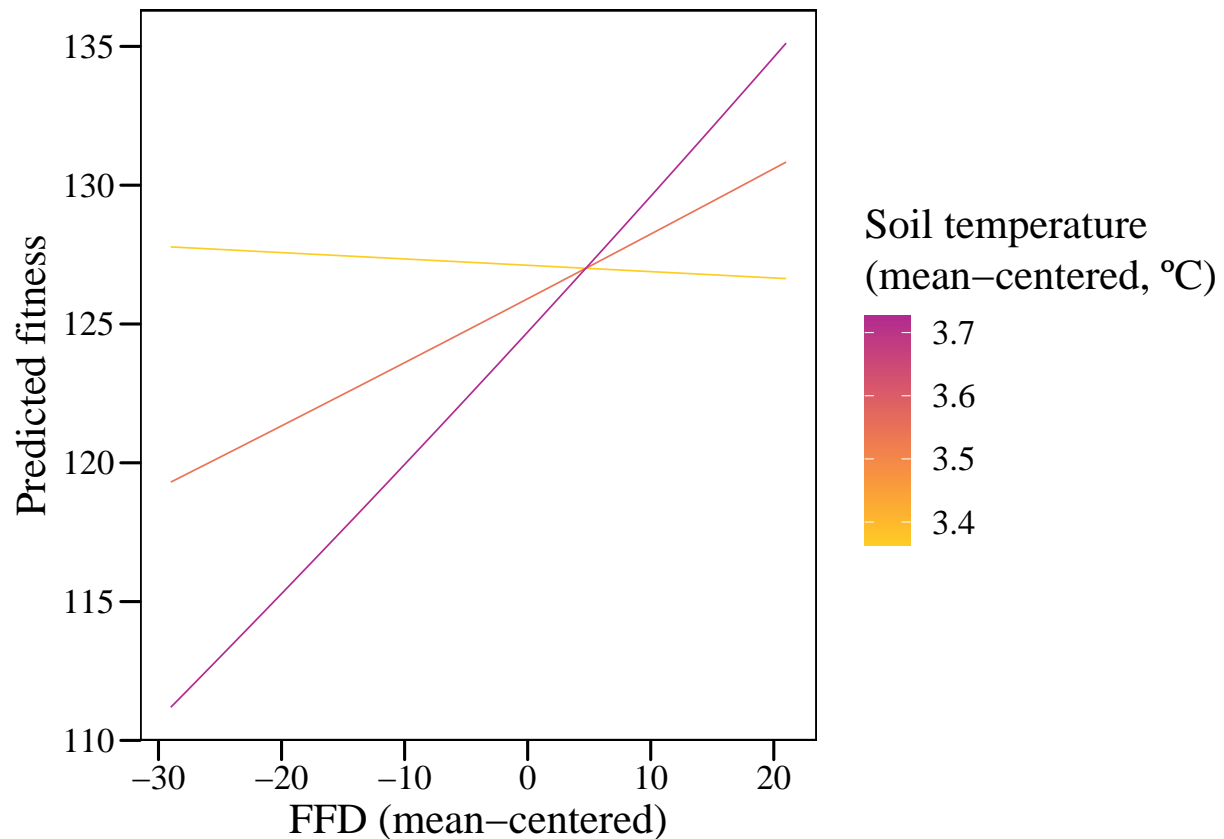
```

```
## 4885.n_seeds 3.727273 13.4242424 131.3024
## 4886.n_seeds 3.727273 13.9292929 131.5546
## 4887.n_seeds 3.727273 14.4343434 131.8070
## 4888.n_seeds 3.727273 14.9393939 132.0598
## 4889.n_seeds 3.727273 15.4444444 132.3129
## 4890.n_seeds 3.727273 15.9494949 132.5663
## 4891.n_seeds 3.727273 16.4545455 132.8200
## 4892.n_seeds 3.727273 16.9595960 133.0740
## 4893.n_seeds 3.727273 17.4646465 133.3283
## 4894.n_seeds 3.727273 17.9696970 133.5829
## 4895.n_seeds 3.727273 18.4747475 133.8379
## 4896.n_seeds 3.727273 18.9797980 134.0931
## 4897.n_seeds 3.727273 19.4848485 134.3487
## 4898.n_seeds 3.727273 19.9898990 134.6046
## 4899.n_seeds 3.727273 20.4949495 134.8608
## 4900.n_seeds 3.727273 21.0000000 135.1173
```

```
# selection favours earlier flowering at temp_c up to 3.4,
# and later flowering at higher temp_c
3.4+mean(redata$temp)
```

```
## [1] 16.85861
```

```
# temp_c=3.4 corresponds to temp=16.9
ggplot(subset(aster_predict2.df.n_seed,temp_c>3.2&temp_c<3.8),
  aes(x=FFD_c,y=fit,group=as.factor(temp_c)))+
  geom_line(aes(x=FFD_c,y=fit,color=temp_c),size=0.3)+
  xlab("FFD (mean-centered)")+ylab("Predicted fitness")+
  my_theme_legend()+labs(color="Soil temperature\n(mean-centered, °C)")+
  scale_colour_gradientn(colours = plasma(n=100,begin=0.4,end=0.9,direction=-1))
```

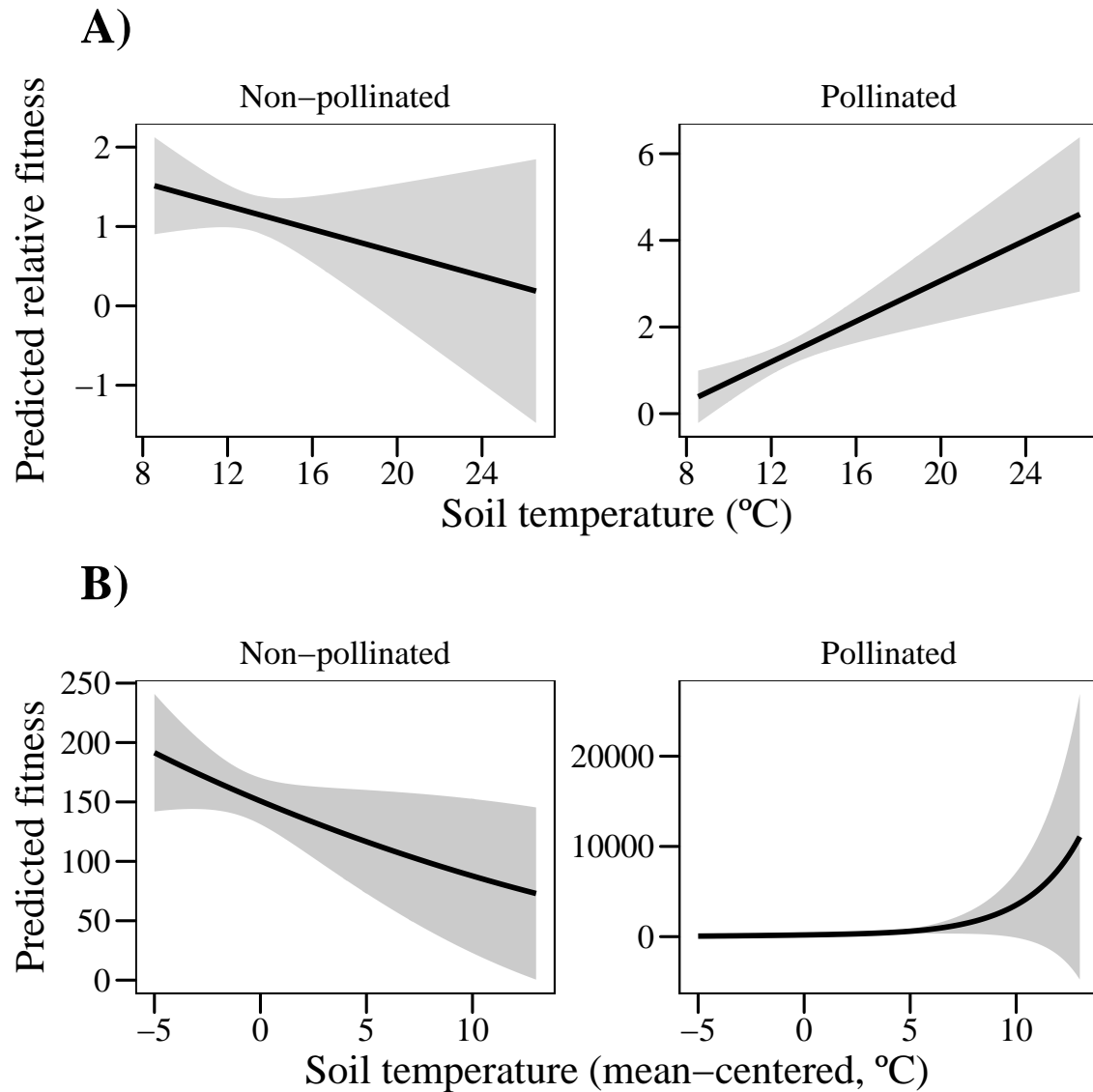


Figures 2-3

Figure 2

Without points

```
fig2a<-ggplot(predict_relfitness3,aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high,
                                     group=group))+
  geom_ribbon(alpha=0.2)+
  geom_line(size=1)+
  xlab("Soil temperature (°C)")+ylab("Predicted relative fitness")+
  my_theme()+ggtitle("A")+
  facet_wrap(~group,scales="free",labeller=labeller(group=P.labs))
fig2b<-ggplot(aster_predict1.df.n_seed)+
  geom_ribbon(aes(x=temp_c,ymin=ymin,ymax=ymax),alpha=0.25)+
  geom_line(aes(x=temp_c,y=fit),size=1)+
  xlab("Soil temperature (mean-centered, °C)")+ylab("Predicted fitness")+
  my_theme()+ggtitle("B")+
  facet_wrap(~P,scales="free",labeller=labeller(P=P.labs))
(fig2<-plot_grid(fig2a, fig2b, ncol=1,align = "v"))
```



```
ggsave(fig2,filename="output/figures/fig2.tiff",device="tiff",
        width=16,height=17,units="cm",dpi=300,compression="lzw")
```

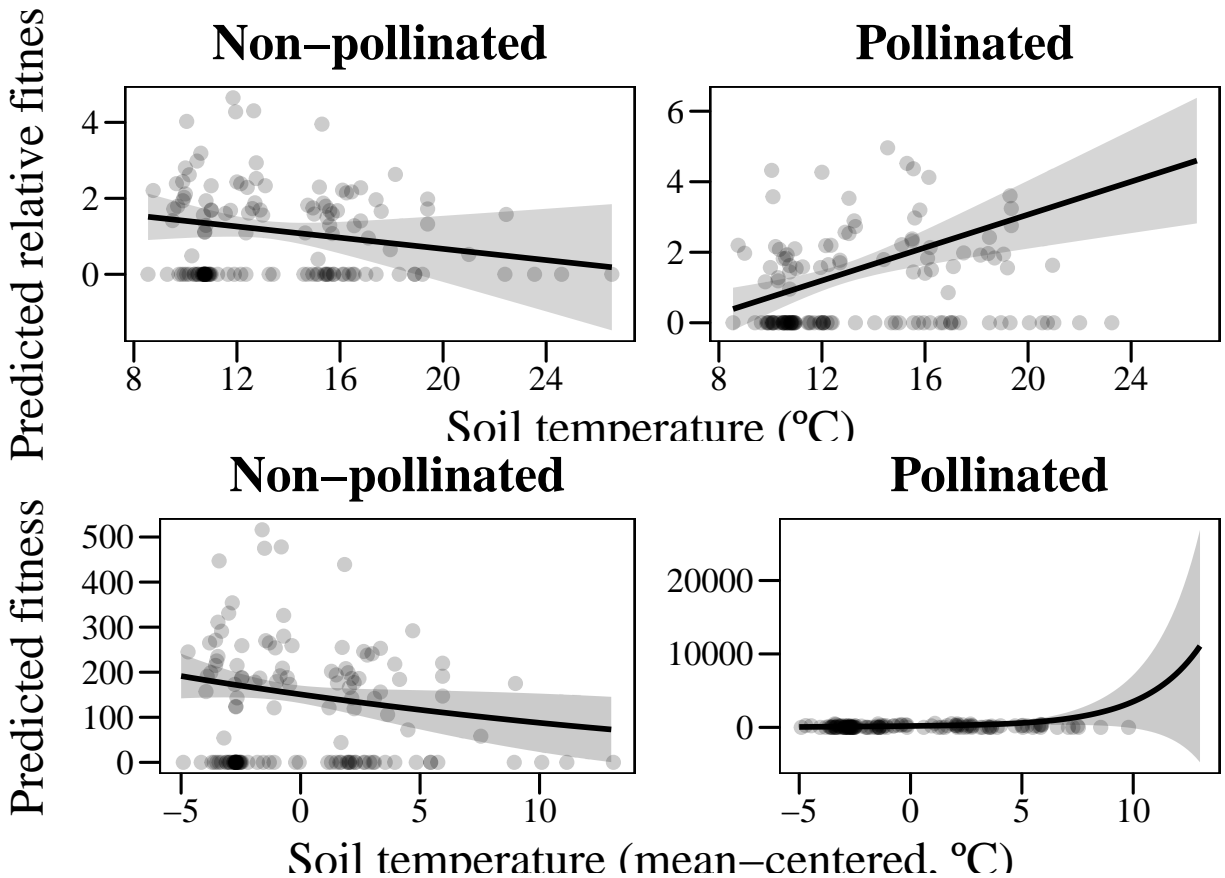
With points

```
fig2apoints<-
  plot_grid(
    ggdraw()+draw_label("Predicted relative fitness",y = 0.5,angle=90,
                        fontfamily="serif",size=18),
    plot_grid(
      plot_grid(
        ggplot(subset(ping_20_data,P==0),aes(x=temp,y=n_seeds_rel))+
          geom_point(size=2,alpha=0.2)+
          geom_ribbon(data=subset(predict_relfitness3,group==0),
                    aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),
```

```

      alpha=0.2))+
    geom_line(data=subset(predict_relfitness3,group==0),
      aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),size=1)+
    xlab(NULL)+ylab(NULL)+
    my_theme()+ggtitle("Non-pollinated")+
    theme(plot.title = element_text(hjust = 0.5)),
    ggplot(subset(ping_20_data,P==1),aes(x=temp,y=n_seeds_rel))+
    geom_point(size=2,alpha=0.2)+
    geom_ribbon(data=subset(predict_relfitness3,group==1),
      aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),
      alpha=0.2)+
    geom_line(data=subset(predict_relfitness3,group==1),
      aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high),size=1)+
    xlab(NULL)+ylab(NULL)+
    my_theme()+ggtitle("Pollinated")+
    theme(plot.title = element_text(hjust = 0.5)),
    ncol=2),
  ggdraw()+draw_label("Soil temperature (°C)",x = 0.5,
    fontfamily="serif",size=18),
  ncol=1,rel_heights=c(1.5, 0.1)),
  ncol=2,rel_widths=c(0.1,1.5)
)
fig2bpoints<-
  plot_grid(
    ggdraw()+draw_label("Predicted fitness",y = 0.5,angle=90,
      fontfamily="serif",size=18),
    plot_grid(
      plot_grid(
        ggplot()+geom_point(data=subset(redata,varb=="n_seeds"&P==0),
          aes(x=temp_c,y=resp),size=2,alpha=0.2)+
        geom_ribbon(data=subset(aster_predict1.df.n_seed,P==0),
          aes(x=temp_c,ymin=ymin,ymax=ymax),alpha=0.25)+
        geom_line(data=subset(aster_predict1.df.n_seed,P==0),
          aes(x=temp_c,y=fit),size=1)+
        xlab(NULL)+ylab(NULL)+
        my_theme()+ggtitle("Non-pollinated")+
        theme(plot.title = element_text(hjust = 0.5)),
        ggplot()+geom_point(data=subset(redata,varb=="n_seeds"&P==1),
          aes(x=temp_c,y=resp),size=2,alpha=0.2)+
        geom_ribbon(data=subset(aster_predict1.df.n_seed,P==1),
          aes(x=temp_c,ymin=ymin,ymax=ymax),alpha=0.25)+
        geom_line(data=subset(aster_predict1.df.n_seed,P==1),
          aes(x=temp_c,y=fit),size=1)+
        xlab(NULL)+ylab(NULL)+
        my_theme()+ggtitle("Pollinated")+
        theme(plot.title = element_text(hjust = 0.5)),
        ncol=2),
      ggdraw()+draw_label("Soil temperature (mean-centered, °C)",x = 0.5,
        fontfamily="serif",size=18),
      ncol=1,rel_heights=c(1.5, 0.1)),
      ncol=2,rel_widths=c(0.1,1.5)
    )
  )
(fig2points<-plot_grid(fig2apoints, fig2bpoints, ncol=1,align = "v"))

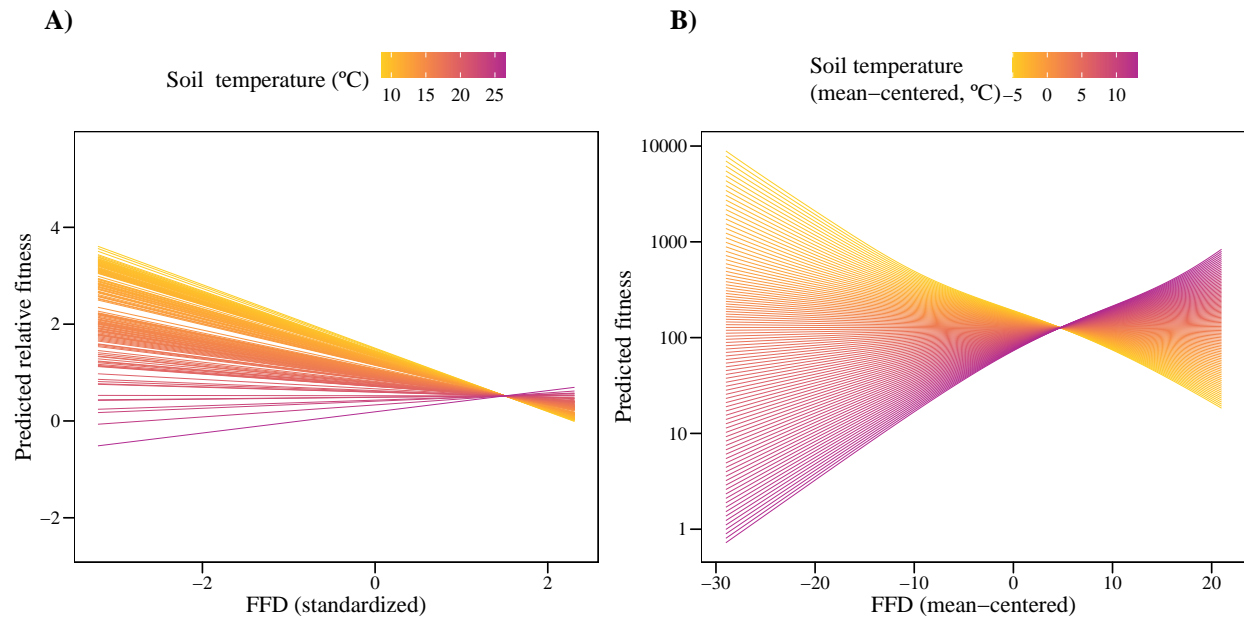
```

```
ggsave(fig2points,filename="output/figures/fig2points.tiff",device="tiff",
       width=16,height=17,units="cm",dpi=300,compression="lzw")
```

Figure 3

```
fig3a<-ggplot(predict_relfitness1,aes(x=x,y=predicted,ymin=conf.low,ymax=conf.high,
                                     group=group))+
  geom_line(aes(color=as.numeric(as.character(group))),size=0.3)+
  xlab("FFD (standardized)")+ylab("Predicted relative fitness")+
  my_theme_legend()+
  scale_colour_gradientn(colours = plasma(n=100,begin=0.4,end=0.9,direction=-1))+
  labs(colour="Soil temperature (°C)")+
  theme(legend.position="top")+ggtitle("A")
fig3b<-ggplot(aster_predict2.df.n_seed,aes(x=FFD_c,y=fit,group=as.factor(temp_c)))+
  geom_line(aes(x=FFD_c,y=fit,color=temp_c),size=0.3)+
  xlab("FFD (mean-centered)")+ylab("Predicted fitness")+
  my_theme_legend()+scale_y_log10()+
  labs(colour="Soil temperature\n(mean-centered, °C)")+
  scale_colour_gradientn(colours = plasma(n=100,begin=0.4,end=0.9,direction=-1))+
  theme(legend.position="top")+ggtitle("B")
(fig3<-plot_grid(fig3a, fig3b,align="h",rel_widths=c(1, 1.08)))
```



```
ggsave(fig3,filename="output/figures/fig3.tiff",device="tiff",
        width=19,height=11,units="cm",dpi=300,compression="lzw")
```

Session info