

MEMgene paper analysis

Contents

1. Preparations	2
a) Packages	2
b) Import spatial coordinates	2
c) Parameter space	3
d) Create R replicate random samples of $n = 30$ sites	4
e) Create R replicate random samples of $n = 60$ sites	4
2. Genetic data	4
a) Select datasets	4
b) Run function ‘getGenData’ to extract the genetic data	5
c) Run function ‘getDgen’ to calculate sample Fst and genetic distances	6
d) Run function ‘getMEMgene’ to obtain RsqAdj and Moran’s I	8
3. Compile results for $n = 90$	11
4. Analyze data for subsets with 30 pops	13
a) Repeat MEMgene with $n = 30$	13
b) Compile results for $n = 30$	13
5. Compare $n = 30$ to $n = 90$	14
c) Paired t-tests (without IM)	14
d) Effect sizes (without IM)	15
6. Missing sites	16
a) Leave 30 sites out at a time, two modes	16
b) Compile results for $n = 60$, two modes	17
7. Summary Interaction Plots	18
a) Figure 1	19
b) Figure 2	20
c) Figure 3	21
8. Interaction Plots for supplementary material using the Dch as a measure for genetic distance . .	24
a) Figure S1	24
b) Figure S2	25
c) Figure S3	27

Analysis of simulated datasets from Lotterhos & Whitlock with the same random sampling of 90 sites as in Wagner et al. (2017).

1. Preparations

a) Packages

```
library(hierfstat)
```

```
## Registered S3 method overwritten by 'spdep':  
##   method      from  
##   plot.mst    ape
```

```
library(memgene)  
library(here)
```

```
## here() starts at /Users/helene/OneDrive - UofT/OneDrive - University of Toronto/R/MEMgene_paper
```

```
library(parallel)  
library(dplyr)
```

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

```
library(spdep)
```

```
## Loading required package: sp  
  
## Loading required package: spData  
  
## Loading required package: sf  
  
## Linking to GEOS 3.7.2, GDAL 2.4.2, PROJ 5.2.0
```

b) Import spatial coordinates

```
Coords <- list()  
Coords$Pairs <- list()  
Coords$Transect <- list()  
Coords$Random <- list()  
Coords$Random$E453 <- Coords$Random$E988 <- Coords$Random$E950 <-  
  read.table("SchemeRandom1.txt") # upload this file
```

```

for(i in 1:length(Coords))
{
  if(length(Coords[[i]]) > 0)
  {
    for(k in 1:length(Coords[[i]]))
    {
      b <- order(Coords[[i]][[k]][,3], Coords[[i]][[k]][,2]) # Sort by y, then x
      Coords[[i]][[k]] <- Coords[[i]][[k]][b,] # correct order!!
    }
  }
}

```

c) Parameter space

Genetic data: find all file names in the folder “SimFilesLFMM” that contain ‘lfmm’:

```

FileNames.lfmm <- list.files(paste0(here::here(), "/dryad//SimFilesLFMM/"), pattern="lfmm")

test <- Reduce(rbind, strsplit(FileNames.lfmm, split="_"))
test <- cbind(test, Reduce(rbind, strsplit(test[,ncol(test)], split=".")))
tmp <- strsplit(test[,2], split=".xs")
test2 <- matrix(NA, nrow(test), 3)
for(i in 1:nrow(test)) test2[i,1:length(tmp[[i]])] <- tmp[[i]]
test2 <- gsub("[A-Z, a-z]", "", test2)
test <- cbind(test, test2)
dimnames(test) <- list(NULL, c("Demography", "Design", "ID", "Env", "ID2",
                               "V6", "NumPops", "V8", "V9", "NumInd", "Design2",
                               "NumPops2", "NumTrans", "NumInd2"))

```

Design matrix (parameter space): each row is one combination of parameter settings:

- Demography: single refugium (1R), two refugia (2R), isolation by distance (IBD), island model (IM)
- Design: sampling design (R, P, T; see below) and number of pops sampled
- Env: which of the three replicate landscapes ‘453’, ‘950’, ‘988’
- NumPops: how many populations are sampled
- NumInd: how many individuals sampled per pop
- Type: random (R), pairs (P), transects (T)

```

Design <- as.data.frame(test[,c(1,2,4,7,10,13,14)])
Design$Env <- ordered(Design$Env, levels=c(453,988,950))
Design$Type <- ordered(substr(Design$Design, 1, 1), levels=c("P", "T", "R"))
Design$Design <- as.character(Design$Design)
Design$Design[Design$Design == "T30.T3x10"] <- "T30.3x10s"
Design$Design[Design$Design == "T30.T6x5s"] <- "T30.6x5s"
Design$NumPops <- as.numeric(as.character(Design$NumPops))
head(Design)

```

##	Demography	Design	Env	NumPops	NumInd	NumTrans	NumInd2	Type
## 1	1R	P30	453	30	20	<NA>	<NA>	P
## 2	1R	P30	453	30	60	<NA>	<NA>	P
## 3	1R	P30	988	30	20	<NA>	<NA>	P

```
## 4      1R    P30 988      30    60    <NA>    <NA>    P
## 5      1R    P30 950      30    20    <NA>    <NA>    P
## 6      1R    P30 950      30    60    <NA>    <NA>    P
```

d) Create R replicate random samples of $n = 30$ sites

```
R = 10
Sites.30 <- list()
set.seed(19)
for(r in 1:R)
{
  Sites.30[[r]] <- sort(sample(1:90, 30, replace=FALSE))
}

saveRDS(Sites.30, paste0(here::here(), "/output/Sites.30.rds"))
```

e) Create R replicate random samples of $n = 60$ sites

```
R = 10
Results.drop <- rep( list(list()), R)

set.seed(297)
Drop <-lapply(c(1:R), function(r) sort(sample(1:90, 30, replace=FALSE)))

saveRDS(Drop, paste0(here::here(), "/output/Drop.rds"))
```

2. Genetic data

a) Select datasets

Select all data files with the largest sample size (90) and random sampling (R)

```
Sites.R.90 <- c(1:nrow(Design))[Design$NumPops==90 & Design$Type=="R"]
```

Check parameters for selected datasets:

```
Design[Sites.R.90,]
```

##	Demography	Design	Env	NumPops	NumInd	NumTrans	NumInd2	Type
## 32	1R	R90	453	90	20	<NA>	<NA>	R
## 33	1R	R90	453	90	6	<NA>	<NA>	R
## 34	1R	R90	988	90	20	<NA>	<NA>	R
## 35	1R	R90	988	90	6	<NA>	<NA>	R
## 36	1R	R90	950	90	20	<NA>	<NA>	R
## 37	1R	R90	950	90	6	<NA>	<NA>	R
## 93	2R	R90	453	90	20	<NA>	<NA>	R
## 94	2R	R90	453	90	6	<NA>	<NA>	R
## 95	2R	R90	988	90	20	<NA>	<NA>	R

## 96	2R	R90 988	90	6	<NA>	<NA>	R
## 97	2R	R90 950	90	20	<NA>	<NA>	R
## 98	2R	R90 950	90	6	<NA>	<NA>	R
## 153	IBD	R90 453	90	20	<NA>	<NA>	R
## 154	IBD	R90 453	90	6	<NA>	<NA>	R
## 155	IBD	R90 988	90	20	<NA>	<NA>	R
## 156	IBD	R90 988	90	6	<NA>	<NA>	R
## 157	IBD	R90 950	90	20	<NA>	<NA>	R
## 158	IBD	R90 950	90	6	<NA>	<NA>	R
## 213	IM	R90 453	90	20	<NA>	<NA>	R
## 214	IM	R90 453	90	6	<NA>	<NA>	R
## 215	IM	R90 988	90	20	<NA>	<NA>	R
## 216	IM	R90 988	90	6	<NA>	<NA>	R
## 217	IM	R90 950	90	20	<NA>	<NA>	R
## 218	IM	R90 950	90	6	<NA>	<NA>	R

b) Run function ‘getGenData’ to extract the genetic data

This function extracts the genetic data, selects the 9900 neutral loci, and adds a first column ‘pop’ with site as a factor. This is the format for functions in the package `hierfstat`.

Define function:

```
getGenData <- function(j)
{
  # Select the sites that need to be sampled for this run:
  i=Sites.R.90[j]
  cat("j:", j, ", i:", i, "\n")

  # Each file has NumPops x NumInd rows (sampled individuals) and up to 10000 columns (loci)
  tmp <- read.table(paste0(here::here(), "/dryad/SimFilesLFMM/", Filenames.lfmm[[i]]))

  # Drop non-neutral loci
  tmp <- tmp[,1:9900]

  # Randomize order of neutral loci
  tmp <- tmp[,sample(1:9900)]

  # Site: create vector of sites = pops (for each row = individual)
  Site <- rep(1:Design$NumPops[i], each=as.numeric(as.vector(Design$NumInd)[i]))

  #Data.hierfstat <- data.frame(pop=factor(Site), tmp)
  Data.hierfstat <- data.frame(pop=Site, tmp)

  return(Data.hierfstat)
}
```

Run in parallel and save results:

```
start_time <- Sys.time()

Index.j <- 1:length(Sites.R.90)
```

```
Data.R.90 <- mclapply(Index.j, function(j) getGenData(j),
                     mc.cores=detectCores())
names(Data.R.90) <- Sites.R.90

end_time <- Sys.time()
end_time - start_time
```

Time difference of 28.07318 secs

```
cat("This job took", (end_time - start_time)/length(Index.j), "per dataset.")
```

This job took 1.169716 per dataset.

```
saveRDS(Data.R.90, paste0(here::here(), "/output/Data.R.90.rds"))
```

Results are stored in a list (one element per site j with 90 samples), with each element:

- Data.frame with first column “pop” (factor of site IDs) and 9900 columns of neutral loci

c) Run function ‘getDgen’ to calculate sample Fst and genetic distances

Define function:

```
#Data.R.90 <- readRDS(paste0(here::here(), "/output/Data.R.90.rds"))

getDgen <- function(j, Data=Data.R.90, dist.method = c("Fst", "Dch"), nLoci=9900)
{
  Fst <- basic.stats(Data[[j]][,1:(nLoci+1)])$overall

  Fst.30 <- t(sapply(1:length(Sites.30), function(s)
    basic.stats(filter(Data[[j]][, 1:(nLoci+1)], is.element(pop, Sites.30[[s]])))$overall))

  D <- list()
  for(k in 1:length(dist.method))
  {
    D[[k]] <- genet.dist(Data[[j]][,1:(nLoci+1)], method = dist.method[k])
  }
  names(D) <- dist.method

  return(list(Fst=Fst, Fst.30 = Fst.30, Dgen=D))
}
```

Run in parallel and save results: (this takes a longer time: 10.4 min on Helene’s iMac)

```
start_time <- Sys.time()

Index.j <- 1:length(Sites.R.90)

Dgen.R.90 <- mclapply(Index.j, function(j) getDgen(j, Data=Data.R.90, nLoci=9900,
                                                    dist.method=c("Fst", "Dch")),
```

```

        mc.cores=detectCores())
names(Dgen.R.90) <- Sites.R.90

end_time <- Sys.time()
end_time - start_time

```

Time difference of 20.52821 mins

```
cat("This job took", (end_time - start_time)/length(Index.j), "per dataset.")
```

This job took 0.8553421 per dataset.

```
saveRDS(Dgen.R.90, paste0(here::here(), "/output/Dgen.R.90.rds"))
```

Repeat for 3300 loci

(Note: order of loci has been randomized)

```

start_time <- Sys.time()

Index.j <- 1:length(Sites.R.90)

Dgen.R.90.3 <- mclapply(Index.j, function(j) getDgen(j, Data=Data.R.90, nLoci=3300,
                                                    dist.method=c("Fst", "Dch")),
                        mc.cores=detectCores())
names(Dgen.R.90.3) <- Sites.R.90

end_time <- Sys.time()
end_time - start_time

```

Time difference of 6.045129 mins

```
cat("This job took", (end_time - start_time)/length(Index.j), "per dataset.")
```

This job took 0.2518804 per dataset.

```
saveRDS(Dgen.R.90.3, paste0(here::here(), "/output/Dgen.R.90.3.rds"))
```

Repeat for 500 loci

```

start_time <- Sys.time()

Index.j <- 1:length(Sites.R.90)

Dgen.R.90.500 <- mclapply(Index.j, function(j) getDgen(j, Data=Data.R.90, nLoci=500,
                                                    dist.method=c("Fst", "Dch")),
                        mc.cores=detectCores())
names(Dgen.R.90.500) <- Sites.R.90

end_time <- Sys.time()
end_time - start_time

```

```
## Time difference of 53.55367 secs
```

```
cat("This job took", (end_time - start_time)/length(Index.j), "per dataset.")
```

```
## This job took 2.231403 per dataset.
```

```
saveRDS(Dgen.R.90.500, paste0(here::here(), "/output/Dgen.R.90.500.rds"))
```

Results are stored in a list (one element per site j with 90 samples) with these elements:

- Fst: vector with ‘overall’ statistics returned by `basic.stats` function
- Fst.30: matrix with statistics for ten subsets of $n = 30$ sites
- Dgen: list of pairwise genetic distance matrices
 - Fst (Fst)
 - Cavalli-Sforza and Edwards Chord distance (Dch)

d) Run function ‘getMEMgene’ to obtain RsqAdj and Moran’s I

This function extracts the grid coordinates, performs memgene analysis with forward selection to obtain the adjusted Rsquare. It also calculates Moran’s I for the genetic data by obtaining a scalogram S (Rsquare for each MEM vector, which sum to 1 across all MEM vectors) and multiplying with the rescaled MEM eigenvectors (Moran’s I for each vector). It also returns the limits for Moran’s I of the genetic data (min and max of Moran’s I values of MEM vectors).

Argument ‘subset’ specifies a subset of sites to be used (metaanalysis mode, performing MEM with subset only). Argument ‘drop’ specifies which sites should be dropped (for comparative mode, keeping MEM of full dataset).

Define function:

```
getMEMgene <- function(j, Dgen = Dgen.R.90, subset=NULL, drop=NULL)
{
  i=Sites.R.90[j]
  cat("j:", j, ", i:", i, "\n")

  # Extract the grid coordinates of the sampled sites
  coord <- data.matrix(Coords[[as.numeric(Design$Type[i])]]
    [[as.numeric(Design$Env[i])]][,2:3])
  if(length(subset) > 0) {coord <- coord[subset,]}
  if(length(drop) > 0) { coord.drop <- coord[-drop,]}

  Res <- list()
  for(k in 1: length(Dgen[[j]]$Dgen))
  {

    Y <- as.matrix(Dgen[[j]]$Dgen[[k]])
    if(length(drop) > 0) {Y = Y[-drop, -drop]}
    if(length(subset) > 0) {Y <- Y[subset, subset]}

    # memGene

    MEM <- mgMEM(dist(coord))
```



```

if(length(drop) > 0) {MEM$vectorMEM <- MEM$vectorMEM[-drop,]}

Positive <- mgForward(Y, MEM$vectorMEM[, MEM$valuesMEM > 0])

RsqaAdjPos = 0
if(!is.na(Positive$selectedRsqaAdj)) { RsqaAdjPos = Positive$selectedRsqaAdj}

if(length(drop) > 0)
{
  MEM.drop <- mgMEM(dist(coord.drop))

  Positive <- mgForward(Y, MEM.drop$vectorMEM[, MEM.drop$valuesMEM > 0])
  RsqaAdjPos.drop = 0
  if(!is.na(Positive$selectedRsqaAdj)) { RsqaAdjPos.drop = Positive$selectedRsqaAdj}
}

# Centre distance matrix
n <- nrow(Y)
row.wt = rep(1, nrow(Y))
col.wt = rep(1, ncol(Y))
st <- sum(col.wt)
sr <- sum(row.wt)
row.wt <- row.wt/sr
col.wt <- col.wt/st
Y <- -0.5 * (Y * Y)
row.mean <- apply(row.wt * Y, 2, sum)
col.mean <- apply(col.wt * t(Y), 2, sum)
col.mean <- col.mean - sum(row.mean * col.wt)
Y <- sweep(Y, 2, row.mean)
G <- t(sweep(t(Y), 2, col.mean))

# Get Rsq for each MEM vector from a separate dbRDA for each vector m
X <- MEM$vectorMEM

S <- rep(0, ncol(X))
for(m in 1:ncol(X))
{
  if(var(X[,m]) > 0)
  {
    p = 1
    H <- X[,m] %*% solve(t(X[,m]) %*% X[,m]) %*% t(X[,m])
    I <- diag(n)
    res <- (I - H) %*% G %*% (I - H)
    S[m] <- 1 - sum(diag(res))/sum(diag(G))
  }
}

S.tot = sum(S)
S.min = min(S)
S[S < 0] <- 0

Values <- MEM$valuesMEM / abs(sum(MEM$valuesMEM))
Range <- range(Values)

```

```

Morans.I <- as.vector(S %*% Values)

Res[[k]] <- list(RsqAdjPos=RsqAdjPos, Morans.I=Morans.I, Range=Range)
}
names(Res) <- names(Dgen[[1]]$Dgen)

return(Res)
}

```

Run function in parallel:

```

#Dgen.R.90 <- readRDS(paste0(here::here(), "/output/Dgen.R.90.rds"))

start_time <- Sys.time()

Index.j <- 1:length(Sites.R.90)

Results.R.90 <- mclapply(Index.j, function(j) getMEMgene(j, Dgen = Dgen.R.90, subset=NULL),
                        mc.cores=detectCores())
names(Results.R.90) <- Sites.R.90

end_time <- Sys.time()
end_time - start_time

```

```
## Time difference of 1.351861 mins
```

```
cat("This job took", (end_time - start_time)/length(Index.j), "per dataset.")
```

```
## This job took 0.05632753 per dataset.
```

```
saveRDS(Results.R.90, paste0(here::here(), "/output/Results.R.90.rds"))
```

Repeat for 3300 loci

```

#Dgen.R.90.3 <- readRDS(paste0(here::here(), "/output/Dgen.R.90.3.rds"))

start_time <- Sys.time()

Index.j <- 1:length(Sites.R.90)

Results.R.90.3 <- mclapply(Index.j, function(j) getMEMgene(j, Dgen = Dgen.R.90.3, subset=NULL),
                        mc.cores=detectCores())
names(Results.R.90.3) <- Sites.R.90

end_time <- Sys.time()
end_time - start_time

```

```
## Time difference of 1.308621 mins
```

```
cat("This job took", (end_time - start_time)/length(Index.j), "per dataset.")
```

```
## This job took 0.05452588 per dataset.
```

```
saveRDS(Results.R.90.3, paste0(here::here(), "/output/Results.R.90.3.rds"))
```

Repeat for 500 loci

```
start_time <- Sys.time()
```

```
Index.j <- 1:length(Sites.R.90)
```

```
Results.R.90.500 <- mclapply(Index.j, function(j) getMEMgene(j, Dgen = Dgen.R.90.500, subset=NULL),  
                             mc.cores=detectCores())
```

```
names(Results.R.90.500) <- Sites.R.90
```

```
end_time <- Sys.time()
```

```
end_time - start_time
```

```
## Time difference of 1.182783 mins
```

```
cat("This job took", (end_time - start_time)/length(Index.j), "per dataset.")
```

```
## This job took 0.04928262 per dataset.
```

```
saveRDS(Results.R.90.500, paste0(here::here(), "/output/Results.R.90.500.rds"))
```

Results are stored in a list (one element per site j with 90 samples) with these elements:

- List with one element per genetic distance measure:
 - RsqAdjPos: adjusted Rsquare from memgene, based on MEM with positive eigenvalues
 - Morans.I: Moran's I for the genetic data
 - Range: range (minimum and maximum) of Moran's I of MEM vectors (limits for Moran's I of genetic data)

3. Compile results for n = 90

Import results without overwriting:

```
Results <- readRDS(paste0(here::here(), "/output/Results.R.90.rds"))  
Results.3 <- readRDS(paste0(here::here(), "/output/Results.R.90.3.rds"))  
Results.500 <- readRDS(paste0(here::here(), "/output/Results.R.90.500.rds"))
```

```
D <- readRDS(paste0(here::here(), "/output/Dgen.R.90.rds"))
```

```
D.3 <- readRDS(paste0(here::here(), "/output/Dgen.R.90.3.rds"))
```

```
D.500 <- readRDS(paste0(here::here(), "/output/Dgen.R.90.500.rds"))
```

Function to extract response variables

```

getRes <- function(Results=Results, k = 1)
{
  RsqAdjPos = sapply(Results, function(ls) ls[[k]]$RsqAdjPos)
  Morans.I = sapply(Results, function(ls) ls[[k]]$Morans.I)
  I.max = sapply(Results, function(ls) max(ls[[k]]$Range))
  I.min = sapply(Results, function(ls) min(ls[[k]]$Range))
  I.scaled = Morans.I / I.max
  res <- data.frame(RsqAdjPos=RsqAdjPos, Morans.I=Morans.I, I.scaled=I.scaled)
}

```

Combine response variables with Design matrix

```

Ftab <- data.frame(Fst.90 = sapply(D, function(ls) ls$Fst[7]),
  Fst.30.m = sapply(D, function(ls) mean(ls$Fst.30[,7])),
  Fst.30.s = sapply(D, function(ls) sd(ls$Fst.30[,7])),
  Fst.90.3 = sapply(D.3, function(ls) ls$Fst[7]),
  Fst.30.m.3 = sapply(D.3, function(ls) mean(ls$Fst.30[,7])),
  Fst.30.s.3 = sapply(D.3, function(ls) sd(ls$Fst.30[,7])),
  Fst.90.500 = sapply(D.500, function(ls) ls$Fst[7]),
  Fst.30.m.500 = sapply(D.500, function(ls) mean(ls$Fst.30[,7])),
  Fst.30.s.500 = sapply(D.500, function(ls) sd(ls$Fst.30[,7])))

res <- lapply(c(1:length(Results[[1]])), function(k) getRes(Results, k))
res.3 <- lapply(c(1:length(Results.3[[1]])), function(k) getRes(Results.3, k))
res.500 <- lapply(c(1:length(Results.500[[1]])), function(k) getRes(Results.500, k))
names(res) <- names(res.3) <- names(res.500) <- names(Results[[1]])

res.combined <- Reduce(cbind, res)
res.3.combined <- Reduce(cbind, res.3)
res.500.combined <- Reduce(cbind, res.500)
names(res.combined) <- paste(rep(names(Results[[1]]), each=ncol(res[[1]])),
  names(res[[1]]), "90", sep=".")
names(res.3.combined) <- paste(names(res.combined), "3", sep=".")
names(res.500.combined) <- paste(names(res.combined), "500", sep=".")

Results.table <- data.frame(Design[ Sites.R.90, ], Ftab, res.combined,
  res.3.combined, res.500.combined)
saveRDS(Results.table, paste0(here::here(), "/output/Results.table.rds"))

rm(list=c("Results", "Results.3", "Results.500", "D", "D.3", "D.500"))

```

Analyze according to the following factors:

- Demography (1R, 2R, IBD, IM)
- NumInd: 20, 6
- Genetic distance measure (Fst, Dch)
- Number of loci (9900 or 3300) (where variable names with .3 denote 3300 loci)

Response variables:

- RsqAdj: based on all MEM selected
- RsqAdjPos: as RsqAdj in memgene output with default settings (only positive MEM)

- Rsq: unadjusted (not reported in memgene)
- Morans.I: assuming population is sampled. Correct with $(n-1)/n$ (where $n = 90$) for estimating population Moran's I
- Morans.I.scaled: this is experimental. Divide Morans.I by the maximum value ($\max(\text{Range})$).

Notes:

- Env: for each combination, there are three replicate datasets (Env: 453, 988, 950). Env could be used as a blocking variable.

4. Analyze data for subsets with 30 pops

a) Repeat MEMgene with $n = 30$

The $R = 10$ subsets of $n = 30$ sites for each dataset with $n = 90$ sites were defined above already in object Sites.30.

```
start_time <- Sys.time()

Index.j <- 1:length(Sites.R.90)
R = length(Sites.30)
Results.R.30 <- rep( list(list()), R)

for(r in 1:R)
{
  Results.R.30[[r]] <- mclapply(Index.j, function(j) getMEMgene(j, Dgen = Dgen.R.90,
                                                             subset= Sites.30[[r]]),
                               mc.cores=detectCores())
  names(Results.R.30[[r]]) <- Sites.R.90
}

end_time <- Sys.time()
end_time - start_time
```

Time difference of 37.34008 secs

```
cat("This job took", (end_time - start_time)/length(Index.j), "per dataset.")
```

This job took 1.555837 per dataset.

```
saveRDS(Results.R.30, paste0(here::here(), "/output/Results.R.30.rds"))
```

b) Compile results for $n = 30$

Determine mean and sdev among the $R = 10$ replicate subsets for each dataset with 90 pops.

Import results without overwriting:

```
Results.subsets <- readRDS(paste0(here::here(), "/output/Results.R.30.rds"))
```

Combine mean and sdev of response variables with Design matrix:

```

res.r <- lapply(Results.subsets, function(sub)
  lapply(c(1:length(Results.subsets[[1]][[1]])), function(k) getRes(sub, k)))

tmp <- list()
for(k in 1:length(res.r[[1]]))
{
  RsqAdjPos.30.m <- apply(sapply(res.r, function(sub) sub[[k]]$RsqAdjPos), 1, mean)
  RsqAdjPos.30.s <- apply(sapply(res.r, function(sub) sub[[k]]$RsqAdjPos), 1, sd)
  Morans.I.30.m <- apply(sapply(res.r, function(sub) sub[[k]]$Morans.I), 1, mean)
  Morans.I.30.s <- apply(sapply(res.r, function(sub) sub[[k]]$Morans.I), 1, sd)
  I.scaled.30.m <- apply(sapply(res.r, function(sub) sub[[k]]$I.scaled), 1, mean)
  I.scaled.30.s <- apply(sapply(res.r, function(sub) sub[[k]]$I.scaled), 1, sd)

  tmp[[k]] <- data.frame(RsqAdjPos.30.m=RsqAdjPos.30.m, RsqAdjPos.30.s=RsqAdjPos.30.s,
    Morans.I.30.m=Morans.I.30.m, Morans.I.30.s=Morans.I.30.s,
    I.scaled.30.m=I.scaled.30.m, I.scaled.30.s=I.scaled.30.s)
}

res.r.combined <- Reduce(cbind, tmp)
names(res.r.combined) <- paste(rep(names(Results.subsets[[1]][[1]]), each=ncol(tmp[[1]])),
  names(tmp[[1]]), sep=".")

Results.subsets.table <- data.frame(Results.table, res.r.combined)
saveRDS(Results.subsets.table, paste0(here::here(), "/output/Results.subsets.table.rds"))

```

5. Compare $n = 30$ to $n = 90$

c) Paired t-tests (without IM)

Drop IM:

```
a <- which(Results.subsets.table$Demograph != "IM")
```

Paired t-tests for difference between mean of subsamples with $n = 30$ and corresponding datasets with $n = 90$ sites.

```

with(Results.subsets.table[a,], t.test(Fst.RsqAdjPos.30.m, Fst.RsqAdjPos.90, paired=TRUE))

##
## Paired t-test
##
## data: Fst.RsqAdjPos.30.m and Fst.RsqAdjPos.90
## t = -10.134, df = 17, p-value = 1.274e-08
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -0.09234974 -0.06052216
## sample estimates:
## mean of the differences
## -0.07643595

```

```
with(Results.subsets.table[a,], t.test(Fst.Morans.I.30.m, Fst.Morans.I.90, paired=TRUE))
```

```
##
## Paired t-test
##
## data: Fst.Morans.I.30.m and Fst.Morans.I.90
## t = -17.211, df = 17, p-value = 3.429e-12
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -0.09968834 -0.07791683
## sample estimates:
## mean of the differences
## -0.08880259
```

```
with(Results.subsets.table[a,], t.test(Fst.I.scaled.30.m, Fst.I.scaled.90, paired=TRUE))
```

```
##
## Paired t-test
##
## data: Fst.I.scaled.30.m and Fst.I.scaled.90
## t = -9.5475, df = 17, p-value = 3.039e-08
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -0.05254855 -0.03352733
## sample estimates:
## mean of the differences
## -0.04303794
```

d) Effect sizes (without IM)

Cohen's d for paired t-test.

```
with(Results.subsets.table[a,], mean(Fst.RsqAdjPos.30.m - Fst.RsqAdjPos.90)/
      sd(Fst.RsqAdjPos.30.m - Fst.RsqAdjPos.90))
```

```
## [1] -2.388539
```

```
with(Results.subsets.table[a,], mean(Fst.Morans.I.30.m - Fst.Morans.I.90)/
      sd(Fst.Morans.I.30.m - Fst.Morans.I.90))
```

```
## [1] -4.056722
```

```
with(Results.subsets.table[a,], mean(Fst.I.scaled.30.m - Fst.I.scaled.90)/
      sd(Fst.I.scaled.30.m - Fst.I.scaled.90))
```

```
## [1] -2.250356
```

6. Missing sites

Simulations to compare results between full sample and sample with some missing sites. Two possible methods:

- Meta-analysis mode: redo MEMgene (including MEM) for each dataset, based on available sites
- Comparative mode: keep MEM, do adj R2 based on positive MEM, which avoids overfitting

a) Leave 30 sites out at a time, two modes

Comparative mode: use MEM from full dataset

```
start_time <- Sys.time()

Index.j <- 1:length(Sites.R.90)
R = 10
Results.drop <- rep( list(list()), R)

set.seed(297)
Drop <-lapply(c(1:R), function(r) sort(sample(1:90, 30)))

for(r in 1:R)
{
  cat(r)
  Results.drop[[r]] <- mclapply(Index.j, function(j) getMEMgene(j, Dgen = Dgen.R.90,
                                                                subset=NULL, drop=Drop[[r]]),
                               mc.cores=detectCores())
  names(Results.drop[[r]]) <- Sites.R.90
}
```

```
## 12345678910
```

```
end_time <- Sys.time()
end_time - start_time
```

```
## Time difference of 6.408244 mins
```

```
cat("This job took", (end_time - start_time)/length(Index.j), "per dataset.")
```

```
## This job took 0.2670102 per dataset.
```

```
saveRDS(Results.drop, paste0(here::here(), "/output/Results.drop.rds"))
```

Meta-analysis mode: MEM based on subset of 60 sites

```
start_time <- Sys.time()

Index.j <- 1:length(Sites.R.90)
R = 10
Results.drop.MEM <- rep( list(list()), R)
```



```

for(r in 1:R)
{
  cat(r)
  Results.drop.MEM[[r]] <- mclapply(Index.j, function(j) getMEMgene(j, Dgen = Dgen.R.90,
                                                                    subset=c(1:90)[-Drop[[r]]], drop=NULL),
                                mc.cores=detectCores())
  names(Results.drop.MEM[[r]]) <- Sites.R.90
}

```

```
## 12345678910
```

```

end_time <- Sys.time()
end_time - start_time

```

```
## Time difference of 3.403358 mins
```

```
cat("This job took", (end_time - start_time)/length(Index.j), "per dataset.")
```

```
## This job took 0.1418066 per dataset.
```

```
saveRDS(Results.drop.MEM, paste0(here::here(), "/output/Results.drop.MEM.rds"))
```

b) Compile results for $n = 60$, two modes

Determine mean and sdev among the $R = 10$ replicate subsets for each dataset with 5 sites dropped

```

Results.drop <- readRDS(paste0(here::here(), "/output/Results.drop.rds"))
Results.drop.MEM <- readRDS(paste0(here::here(), "/output/Results.drop.MEM.rds"))
Results.subsets.table <- readRDS(paste0(here::here(), "/output/Results.subsets.table.rds"))

```

```

res.r <- lapply(Results.drop, function(sub)
  lapply(c(1:length(Results.drop[[1]] [[1]])), function(k) getRes(sub, k)))

tmp <- list()
for(k in 1:length(res.r[[1]]))
{
  RsqAdjPos.drop.m <- apply(sapply(res.r, function(sub) sub[[k]]$RsqAdjPos), 1, mean)
  RsqAdjPos.drop.s <- apply(sapply(res.r, function(sub) sub[[k]]$RsqAdjPos), 1, sd)
  Morans.I.drop.m <- apply(sapply(res.r, function(sub) sub[[k]]$Morans.I), 1, mean)
  Morans.I.drop.s <- apply(sapply(res.r, function(sub) sub[[k]]$Morans.I), 1, sd)
  I.scaled.drop.m <- apply(sapply(res.r, function(sub) sub[[k]]$I.scaled), 1, mean)
  I.scaled.drop.s <- apply(sapply(res.r, function(sub) sub[[k]]$I.scaled), 1, sd)

  tmp[[k]] <- data.frame(RsqAdjPos.drop.m=RsqAdjPos.drop.m, RsqAdjPos.drop.s=RsqAdjPos.drop.s,
                        Morans.I.drop.m=Morans.I.drop.m, Morans.I.drop.s=Morans.I.drop.s,
                        I.scaled.drop.m=I.scaled.drop.m, I.scaled.drop.s=I.scaled.drop.s)
}

res.r.combined <- Reduce(cbind, tmp)

```

```
names(res.r.combined) <- paste(rep(names(Results.drop[[1]][[1]]), each=ncol(tmp[[1]])),
                             names(tmp[[1]]), sep=".")
```

```
Results.drop.table <- data.frame(Results.subsets.table, res.r.combined)
saveRDS(Results.drop.table, paste0(here::here(), "/output/Results.drop.table.rds"))
```

Repeat with refitted MEM

```
res.r <- lapply(Results.drop.MEM, function(sub)
  lapply(c(1:length(Results.drop[[1]][[1]])), function(k) getRes(sub, k)))

tmp <- list()
for(k in 1:length(res.r[[1]]))
{
  RsqAdjPos.drop.m <- apply(sapply(res.r, function(sub) sub[[k]]$RsqAdjPos), 1, mean)
  RsqAdjPos.drop.s <- apply(sapply(res.r, function(sub) sub[[k]]$RsqAdjPos), 1, sd)
  Morans.I.drop.m <- apply(sapply(res.r, function(sub) sub[[k]]$Morans.I), 1, mean)
  Morans.I.drop.s <- apply(sapply(res.r, function(sub) sub[[k]]$Morans.I), 1, sd)
  I.scaled.drop.m <- apply(sapply(res.r, function(sub) sub[[k]]$I.scaled), 1, mean)
  I.scaled.drop.s <- apply(sapply(res.r, function(sub) sub[[k]]$I.scaled), 1, sd)

  tmp[[k]] <- data.frame(RsqAdjPos.drop.MEM.m=RsqAdjPos.drop.m, RsqAdjPos.drop.MEM.s=RsqAdjPos.drop.s,
                        Morans.I.drop.MEM.m=Morans.I.drop.m, Morans.I.drop.MEM.s=Morans.I.drop.s,
                        I.scaled.drop.MEM.m=I.scaled.drop.m, I.scaled.drop.MEM.s=I.scaled.drop.s)
}

res.r.combined <- Reduce(cbind, tmp)

names(res.r.combined) <- paste(rep(names(Results.drop[[1]][[1]]), each=ncol(tmp[[1]])),
                             names(tmp[[1]]), sep=".")

Results.drop.MEM.table <- data.frame(Results.drop.table, res.r.combined)
saveRDS(Results.drop.MEM.table, paste0(here::here(), "/output/Results.drop.MEM.table.rds"))
```

7. Summary Interaction Plots

```
library("rcompanion")
```

```
## Registered S3 method overwritten by 'DescTools':
##   method      from
##   reorder.factor gdata
```

```
library("ggpubr")
```

```
## Loading required package: ggplot2
```

```
library("ggplot2")
library("gridExtra")
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##      combine

library("reshape2")
library("tidyr")

##
## Attaching package: 'tidyr'

## The following object is masked from 'package:reshape2':
##
##      smiths

source(paste0(here::here(), "/output/InteractionPlotLegend.R"))

Results.table <- readRDS(paste0(here::here(), "/output/Results.drop.MEM.table.rds"))
```

a) Figure 1

With $n=90$, $\#IND=20$ & the 9900 loci: Difference demographies in response variable (Fst, adjusted R^2 and Moran's I).

```
b <- which(Results.table$NumInd == 20)
Results.table$Demography <- factor(Results.table$Demography, levels = c("IM", "IBD", "1R", "2R"))

Sum1.p <- groupwiseMean(Fst.90 ~ Demography, data = Results.table[b,], conf = 0.95, digits = 3)
Pan.a.p <- ggplot(Sum1.p, aes(x = Demography, y = Mean)) + ylim(0.0, 0.015) +
  geom_errorbar(aes(ymin = Trad.lower, ymax = Trad.upper), width = 0.05, size = 0.5) +
  geom_point(shape = 1, size = 2) +
  theme_bw() + theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  ggtitle("(a)") + xlab("Demography") + ylab("Fst") +
  theme(axis.text = element_text(size = 12),
        plot.title = element_text(size = 16, face = "bold"),
        axis.title.x = element_text(size = 12, face = "bold"),
        axis.title.y = element_text(size = 12, face = "bold"))

Sum2.p <- groupwiseMean(Fst.RsqAdjPos.90 ~ Demography, data = Results.table[b,], conf = 0.95, digits = 3)
Pan.b.p <- ggplot(Sum2.p, aes(x = Demography, y = Mean)) + ylim(-0.02, 0.8) +
  geom_errorbar(aes(ymin = Trad.lower, ymax = Trad.upper), width = 0.05, size = 0.5) +
  geom_point(shape = 1, size = 2) +
  theme_bw() + theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  ggtitle("(b)") + xlab("Demography") + ylab(expression(bold('Adjusted R^2'))) +
  theme(axis.text = element_text(size = 12),
        plot.title = element_text(size = 16, face = "bold"),
        axis.title.x = element_text(size = 12, face = "bold"),
        axis.title.y = element_text(size = 12, face = "bold"))
```

```

    )
Sum3.p <- groupwiseMean(Fst.Morans.I.90 ~ Demography, data = Results.table[b,], conf = 0.95, digits =
Pan.c.p<- ggplot(Sum3.p, aes(x = Demography, y = Mean)) + ylim(-0.02,0.8) +
  geom_errorbar(aes(ymin = Trad.lower, ymax = Trad.upper), width = 0.05, size = 0.5) +
  geom_point(shape = 1, size = 2) +
  theme_bw() + theme (panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  ggtitle("(c)") + xlab("Demography") + ylab("Moran's I") +
  theme(axis.text = element_text(size = 12),
        plot.title = element_text(size=16, face="bold"),
        axis.title.x = element_text(size=12, face="bold"),
        axis.title.y = element_text(size=12, face="bold")
  )
figure.1.P <- ggarrange(Pan.a.p, Pan.b.p, Pan.c.p, ncol = 3, nrow = 1)
#figure.1.P

g <- arrangeGrob(Pan.a.p, Pan.b.p, Pan.c.p, ncol = 3, nrow = 1) #generates g
ggsave(file=paste0(here::here(), "/output/Figure1.pdf"), g)

```

Saving 6 x 2.5 in image

b) Figure 2

With n=90 and removing IM: Interaction between #Ind (20, 6) and # Loci (9900, 3300, 500) in response variabe (Fst, adjusted R², and Moran's I).

```

Results.sub.wide <- Results.table %>%
  mutate(Demography = as.character(Demography), NumInd = as.character(NumInd)) %>%
  select(Demography, NumPops, NumInd, Fst.90, Fst.90.3, Fst.90.500, Fst.RsqAdjPos.90, Fst.Morans.I.90,
        Fst.I.scaled.90, Fst.RsqAdjPos.90.3, Fst.Morans.I.90.3, Fst.I.scaled.90.3,
        Fst.RsqAdjPos.90.500, Fst.Morans.I.90.500, Fst.I.scaled.90.500)

# Convert to long format
Results.sub.long <- data.frame(
  Results.sub.wide %>%
    select(Demography, NumPops, NumInd, Fst.90, Fst.90.3, Fst.90.500) %>%
    melt(id.vars=c("Demography", "NumPops", "NumInd")) %>% rename(Fst = value) %>%
    select(Demography, NumPops, NumInd, Fst),
  Results.sub.wide %>%
    select(Demography, NumPops, NumInd, Fst.RsqAdjPos.90, Fst.RsqAdjPos.90.3, Fst.RsqAdjPos.90.500) %>%
    melt(id.vars=c("Demography", "NumPops", "NumInd")) %>% rename(Adj.Rsq.Pos = value) %>%
    select(Adj.Rsq.Pos),
  Results.sub.wide %>%
    select(Demography, NumPops, NumInd, Fst.Morans.I.90, Fst.Morans.I.90.3, Fst.Morans.I.90.500) %>%
    melt(id.vars=c("Demography", "NumPops", "NumInd")) %>% rename(Morans.I = value) %>%
    select(Morans.I),
  Results.sub.wide %>%
    select(Demography, NumPops, NumInd, Fst.I.scaled.90, Fst.I.scaled.90.3, Fst.I.scaled.90.500) %>%
    melt(id.vars=c("Demography", "NumPops", "NumInd")) %>% rename(I.scaled = value) %>%
    select(I.scaled))

# Add number of loci, drop IM

```

```

Results.sub.long <- Results.sub.long %>% mutate(NumLoci = rep(c(9900, 3300, 500), each=24)) %>%
  mutate(NumLoci = factor(NumLoci, levels = c("500", "3300", "9900"))) %>%
  filter(Demography != "IM")

pdf(file = paste0(here::here(), "/output/Figure2.pdf"), width=8, height=3.5)

par(mfcol=c(1,3), mar = c(5, 5, 3, 1))

with(Results.sub.long, {interaction.plot(NumInd, NumLoci, Fst,
                                         col=c("grey1","grey20","grey50"),
                                         ylab=" mean Fst", ylim=c(0.0,0.015), lwd= 3,
                                         cex.lab=1.5, cex.axis=1.5, cex.sub=1.5,
                                         xleg = "bottomright")})

  title("(a)", adj = 0, line =1, cex.main=2 )

with(Results.sub.long, {interaction.plot(NumInd, NumLoci, Adj.Rsq.Pos,
                                         col=c("grey1","grey20","grey50"),
                                         ylab=expression("mean adjusted R"^2), ylim=c(0.0,0.6),
                                         lwd= 3, cex.lab=1.5, cex.axis=1.5, cex.sub=1.5,
                                         legend=FALSE)})

  title("(b)", adj = 0, line =1, cex.main=2 )

with(Results.sub.long, {interaction.plot(NumInd, NumLoci, Morans.I,
                                         col=c("grey1","grey20","grey50"),
                                         ylab="mean Moran's I", ylim=c(0.0,0.6),
                                         lwd= 3, cex.lab=1.5, cex.axis=1.5, cex.sub=1.5,
                                         legend=FALSE)})

  title("(c)", adj = 0, line =1, cex.main=2 )

dev.off()

## pdf
## 2

```

c) Figure 3

Multipanel fig. one figure for each demography all on the same scale. boxplots (20 ind & 9900 loci)

Respo= Adj R^2 & MI & scaled MI. (x axis = 60 refitting, missing values, 30 refitting)... show variability as proportion (divide each value by the value of the 90)

```

# Fst Adjusted Rsq.

D11<- with(Results.table %>% filter(Demography == "IBD" & NumInd == 20),
  data.frame(Comp.60=Fst.RsqAdjPos.drop.m / Fst.RsqAdjPos.90,
             Meta.60=Fst.RsqAdjPos.drop.MEM.m / Fst.RsqAdjPos.90,
             Meta.30=Fst.RsqAdjPos.30.m / Fst.RsqAdjPos.90))
D11<- gather(D11, "Comparison", "Fst.Rsq")
D11.s <- groupwiseMean(Fst.Rsq ~ Comparison, data =D11, conf = 0.95, digits = 3)
D11.s$Comparison <- factor(D11.s$Comparison, levels = c("Comp.60", "Meta.60", "Meta.30"))
D11.a.s<- ggplot(D11.s, aes(x = Comparison, y = Mean)) + ylim(0,1.0) +
  geom_errorbar(aes(ymin = Trad.lower, ymax = Trad.upper), width = 0.05, size = 0.5) +
  geom_point(shape = 1, size = 2) +

```

```

theme_bw() + theme (panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  ggtitle("IBD") + xlab("") + ylab(expression(bold('Adjusted R'^2))) +
  theme(axis.text = element_text(size = 10),
        plot.title = element_text(size=16, face="bold", hjust = 0.5),
        axis.title.y = element_text(size=12, face="bold")
  )

D22<- with(Results.table %>% filter(Demography == "1R" & NumInd == 20),
  data.frame(Comp.60=Fst.RsqAdjPos.drop.m / Fst.RsqAdjPos.90,
             Meta.60=Fst.RsqAdjPos.drop.MEM.m / Fst.RsqAdjPos.90,
             Meta.30=Fst.RsqAdjPos.30.m / Fst.RsqAdjPos.90))
D22<- gather(D22, "Comparison", "Fst.Rsq")
D22.s <- groupwiseMean(Fst.Rsq ~ Comparison, data =D22, conf = 0.95, digits = 3)
D22.s$Comparison <- factor(D22.s$Comparison, levels = c("Comp.60", "Meta.60", "Meta.30"))
D22.a.s<- ggplot(D22.s, aes(x = Comparison, y = Mean)) + ylim(0,1.0) +
  geom_errorbar(aes(ymin = Trad.lower, ymax = Trad.upper), width = 0.05, size = 0.5) +
  geom_point(shape = 1, size = 2) +
  theme_bw() + theme (panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  ggtitle("1R") + xlab("") + ylab("") +
  theme(axis.text = element_text(size = 10),
        plot.title = element_text(size=16, face="bold", hjust = 0.5),
        axis.title.y = element_text(size=12, face="bold")
  )

D33<- with(Results.table %>% filter(Demography == "2R" & NumInd == 20),
  data.frame(Comp.60=Fst.RsqAdjPos.drop.m / Fst.RsqAdjPos.90,
             Meta.60=Fst.RsqAdjPos.drop.MEM.m / Fst.RsqAdjPos.90,
             Meta.30=Fst.RsqAdjPos.30.m / Fst.RsqAdjPos.90))
D33<- gather(D33, "Comparison", "Fst.Rsq")
D33.s <- groupwiseMean(Fst.Rsq ~ Comparison, data =D33, conf = 0.95, digits = 3)
D33.s$Comparison <- factor(D33.s$Comparison, levels = c("Comp.60", "Meta.60", "Meta.30"))
D33.a.s<- ggplot(D33.s, aes(x = Comparison, y = Mean)) + ylim(0,1.0) +
  geom_errorbar(aes(ymin = Trad.lower, ymax = Trad.upper), width = 0.05, size = 0.5) +
  geom_point(shape = 1, size = 2) +
  theme_bw() + theme (panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  ggtitle("2R") + xlab("") + ylab("") +
  theme(axis.text = element_text(size = 10),
        plot.title = element_text(size=16, face="bold", hjust = 0.5),
        axis.title.y = element_text(size=12, face="bold")
  )

D44<- with(Results.table %>% filter(Demography == "IBD" & NumInd == 20),
  data.frame(Comp.60=Fst.Morans.I.drop.m / Fst.Morans.I.90,
             Meta.60=Fst.Morans.I.drop.MEM.m / Fst.Morans.I.90,
             Meta.30=Fst.Morans.I.30.m / Fst.Morans.I.90))
D44<- gather(D44, "Comparison", "Fst.Rsq")
D44.s <- groupwiseMean(Fst.Rsq ~ Comparison, data =D44, conf = 0.95, digits = 3)
D44.s$Comparison <- factor(D44.s$Comparison, levels = c("Comp.60", "Meta.60", "Meta.30"))
D44.a.s<- ggplot(D44.s, aes(x = Comparison, y = Mean)) + ylim(0,1.0) +
  geom_errorbar(aes(ymin = Trad.lower, ymax = Trad.upper), width = 0.05, size = 0.5) +
  geom_point(shape = 1, size = 2) +
  theme_bw() + theme (panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +

```

```

xlab("") + ylab("Moran's I") +
  theme(axis.text = element_text(size = 10),
        axis.title.y = element_text(size=12, face="bold")
  )

D55<- with(Results.table %>% filter(Demography == "1R" & NumInd == 20),
  data.frame(Comp.60=Fst.Morans.I.drop.m / Fst.Morans.I.90,
             Meta.60=Fst.Morans.I.drop.MEM.m / Fst.Morans.I.90,
             Meta.30=Fst.Morans.I.30.m / Fst.Morans.I.90))
D55<- gather(D55, "Comparison", "Fst.Rsq")
D55.s <- groupwiseMean(Fst.Rsq ~ Comparison, data =D55, conf = 0.95, digits = 3)
D55.s$Comparison <- factor(D55.s$Comparison, levels = c("Comp.60", "Meta.60", "Meta.30"))
D55.a.s<- ggplot(D55.s, aes(x = Comparison, y = Mean)) + ylim(0,1.0) +
  geom_errorbar(aes(ymin = Trad.lower, ymax = Trad.upper), width = 0.05, size = 0.5) +
  geom_point(shape = 1, size = 2) +
  theme_bw() + theme (panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  xlab("") + ylab("") +
  theme(axis.text = element_text(size = 10),
        axis.title.y = element_text(size=12, face="bold")
  )

D66<- with(Results.table %>% filter(Demography == "2R" & NumInd == 20),
  data.frame(Comp.60=Fst.Morans.I.drop.m / Fst.Morans.I.90,
             Meta.60=Fst.Morans.I.drop.MEM.m / Fst.Morans.I.90,
             Meta.30=Fst.Morans.I.30.m / Fst.Morans.I.90))
D66<- gather(D66, "Comparison", "Fst.Rsq")
D66.s <- groupwiseMean(Fst.Rsq ~ Comparison, data =D66, conf = 0.95, digits = 3)
D66.s$Comparison <- factor(D66.s$Comparison, levels = c("Comp.60", "Meta.60", "Meta.30"))
D66.a.s<- ggplot(D66.s, aes(x = Comparison, y = Mean)) + ylim(0,1.0) +
  geom_errorbar(aes(ymin = Trad.lower, ymax = Trad.upper), width = 0.05, size = 0.5) +
  geom_point(shape = 1, size = 2) +
  theme_bw() + theme (panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  xlab("") + ylab("") +
  theme(axis.text = element_text(size = 10),
        axis.title.y = element_text(size=12, face="bold")
  )

# Scaled Moran's I

D77<- with(Results.table %>% filter(Demography == "IBD" & NumInd == 20),
  data.frame(Comp.60=Fst.I.scaled.drop.m / Fst.I.scaled.90,
             Meta.60=Fst.I.scaled.drop.MEM.m / Fst.I.scaled.90,
             Meta.30=Fst.I.scaled.30.m / Fst.I.scaled.90))
D77<- gather(D77, "Comparison", "Fst.Rsq")
D77.s <- groupwiseMean(Fst.Rsq ~ Comparison, data =D77, conf = 0.95, digits = 3)
D77.s$Comparison <- factor(D77.s$Comparison, levels = c("Comp.60", "Meta.60", "Meta.30"))
D77.a.s<- ggplot(D77.s, aes(x = Comparison, y = Mean)) + ylim(0,1.0) +
  geom_errorbar(aes(ymin = Trad.lower, ymax = Trad.upper), width = 0.05, size = 0.5) +
  geom_point(shape = 1, size = 2) +
  theme_bw() + theme (panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  xlab("") + ylab("Scaled Moran's I") +
  theme(axis.text = element_text(size = 10),
        axis.title.y = element_text(size=12, face="bold")
  )

```



```

)

D88<- with(Results.table %>% filter(Demography == "1R" & NumInd == 20),
  data.frame(Comp.60=Fst.I.scaled.drop.m / Fst.I.scaled.90,
    Meta.60=Fst.I.scaled.drop.MEM.m / Fst.I.scaled.90,
    Meta.30=Fst.I.scaled.30.m / Fst.I.scaled.90))
D88<- gather(D88, "Comparison", "Fst.Rsq")
D88.s <- groupwiseMean(Fst.Rsq ~ Comparison, data =D88, conf = 0.95, digits = 3)
D88.s$Comparison <- factor(D88.s$Comparison, levels = c("Comp.60", "Meta.60", "Meta.30"))
D88.a.s<- ggplot(D88.s, aes(x = Comparison, y = Mean)) + ylim(0,1.0) +
  geom_errorbar(aes(ymin = Trad.lower, ymax = Trad.upper), width = 0.05, size = 0.5) +
  geom_point(shape = 1, size = 2) +
  theme_bw() + theme (panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  xlab("") + ylab("") +
  theme(axis.text = element_text(size = 10),
    axis.title.y = element_text(size=12, face="bold")
  )

D99<- with(Results.table %>% filter(Demography == "2R" & NumInd == 20),
  data.frame(Comp.60=Fst.I.scaled.drop.m / Fst.I.scaled.90,
    Meta.60=Fst.I.scaled.drop.MEM.m / Fst.I.scaled.90,
    Meta.30=Fst.I.scaled.30.m / Fst.I.scaled.90))
D99<- gather(D99, "Comparison", "Fst.Rsq")
D99.s <- groupwiseMean(Fst.Rsq ~ Comparison, data =D99, conf = 0.95, digits = 3)
D99.s$Comparison <- factor(D99.s$Comparison, levels = c("Comp.60", "Meta.60", "Meta.30"))
D99.a.s<- ggplot(D99.s, aes(x = Comparison, y = Mean)) + ylim(0,1.02) +
  geom_errorbar(aes(ymin = Trad.lower, ymax = Trad.upper), width = 0.05, size = 0.5) +
  geom_point(shape = 1, size = 2) +
  theme_bw() + theme (panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  xlab("") + ylab("") +
  theme(axis.text = element_text(size = 10),
    axis.title.y = element_text(size=12, face="bold")
  )

figure.3.P <- ggarrange(D11.a.s, D22.a.s, D33.a.s, D44.a.s, D55.a.s, D66.a.s ,D77.a.s, D88.a.s, D99.a.s,
  #figure.3.P

g <- arrangeGrob(D11.a.s, D22.a.s, D33.a.s, D44.a.s, D55.a.s, D66.a.s ,D77.a.s, D88.a.s, D99.a.s, ncol = 3)
ggsave(file=paste0(here::here(), "/output/Figure3.pdf"), g)

```

```
## Saving 6 x 6 in image
```

```
=====
```

8. Interaction Plots for supplementary material using the Dch as a measure for genetic distance

a) Figure S1

With n=90, #IND= 20 & the 9900 loci: Difference demographies in response variabe (Fst, adjusted R^2 and Moran's I).


```

#b <- which(Results.table$NumInd == 20)
#Results.table$Demography <- factor(Results.table$Demography, levels = c("IM", "IBD", "1R", "2R"))

Sum1.p <- groupwiseMean(Fst.90 ~ Demography, data = Results.table[b,], conf = 0.95, digits = 3)
Pan.a.p <- ggplot(Sum1.p, aes(x = Demography, y = Mean)) + ylim(0.0, 0.015) +
  geom_errorbar(aes(ymin = Trad.lower, ymax = Trad.upper), width = 0.05, size = 0.5) +
  geom_point(shape = 1, size = 2) +
  theme_bw() + theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  ggtitle("(a)") + xlab("Demography") + ylab("Fst") +
  theme(axis.text = element_text(size = 12),
        plot.title = element_text(size = 16, face = "bold"),
        axis.title.x = element_text(size = 12, face = "bold"),
        axis.title.y = element_text(size = 12, face = "bold"))

Sum2.p <- groupwiseMean(Dch.RsqAdjPos.90 ~ Demography, data = Results.table[b,], conf = 0.95, digits = 3)
Pan.b.p <- ggplot(Sum2.p, aes(x = Demography, y = Mean)) + ylim(-0.02, 0.8) +
  geom_errorbar(aes(ymin = Trad.lower, ymax = Trad.upper), width = 0.05, size = 0.5) +
  geom_point(shape = 1, size = 2) +
  theme_bw() + theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  ggtitle("(b)") + xlab("Demography") + ylab(expression(bold('Adjusted R'^2))) +
  theme(axis.text = element_text(size = 12),
        plot.title = element_text(size = 16, face = "bold"),
        axis.title.x = element_text(size = 12, face = "bold"),
        axis.title.y = element_text(size = 12, face = "bold"))

Sum3.p <- groupwiseMean(Dch.Morans.I.90 ~ Demography, data = Results.table[b,], conf = 0.95, digits = 3)
Pan.c.p <- ggplot(Sum3.p, aes(x = Demography, y = Mean)) + ylim(-0.02, 0.8) +
  geom_errorbar(aes(ymin = Trad.lower, ymax = Trad.upper), width = 0.05, size = 0.5) +
  geom_point(shape = 1, size = 2) +
  theme_bw() + theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  ggtitle("(c)") + xlab("Demography") + ylab("Moran's I") +
  theme(axis.text = element_text(size = 12),
        plot.title = element_text(size = 16, face = "bold"),
        axis.title.x = element_text(size = 12, face = "bold"),
        axis.title.y = element_text(size = 12, face = "bold"))

figure.1.P <- ggarrange(Pan.a.p, Pan.b.p, Pan.c.p, ncol = 3, nrow = 1)
#figure.1.P

g <- arrangeGrob(Pan.a.p, Pan.b.p, Pan.c.p, ncol = 3, nrow = 1) #generates g
ggsave(file=paste0(here::here(), "/output/FigureS1.pdf"), g)

```

Saving 6 x 2.5 in image

b) Figure S2

With n=90 and removing IM: Interaction between #Ind (20, 6) and # Loci (9900, 3300, 500) in response variable (Fst, adjusted R², and Moran's I).

```

Results.sub.wide <- Results.table %>%
  mutate(Demography = as.character(Demography), NumInd = as.character(NumInd)) %>%
  select(Demography, NumPops, NumInd, Fst.90, Fst.90.3, Fst.90.500, Dch.RsqAdjPos.90, Dch.Morans.I.90,
    Dch.I.scaled.90, Dch.RsqAdjPos.90.3, Dch.Morans.I.90.3, Dch.I.scaled.90.3,
    Dch.RsqAdjPos.90.500, Dch.Morans.I.90.500, Dch.I.scaled.90.500)

# Convert to long format
Results.sub.long <- data.frame(
  Results.sub.wide %>%
    select(Demography, NumPops, NumInd, Fst.90, Fst.90.3, Fst.90.500) %>%
    melt(id.vars=c("Demography", "NumPops", "NumInd")) %>% rename(Fst = value) %>%
    select(Demography, NumPops, NumInd, Fst),
  Results.sub.wide %>%
    select(Demography, NumPops, NumInd, Dch.RsqAdjPos.90, Dch.RsqAdjPos.90.3, Dch.RsqAdjPos.90.500) %>%
    melt(id.vars=c("Demography", "NumPops", "NumInd")) %>% rename(Adj.Rsq.Pos = value) %>%
    select(Adj.Rsq.Pos),
  Results.sub.wide %>%
    select(Demography, NumPops, NumInd, Dch.Morans.I.90, Dch.Morans.I.90.3, Dch.Morans.I.90.500) %>%
    melt(id.vars=c("Demography", "NumPops", "NumInd")) %>% rename(Morans.I = value) %>%
    select(Morans.I),
  Results.sub.wide %>%
    select(Demography, NumPops, NumInd, Dch.I.scaled.90, Dch.I.scaled.90.3, Dch.I.scaled.90.500) %>%
    melt(id.vars=c("Demography", "NumPops", "NumInd")) %>% rename(I.scaled = value) %>%
    select(I.scaled))

# Add number of loci, drop IM
Results.sub.long <- Results.sub.long %>% mutate(NumLoci = rep(c(9900, 3300, 500), each=24)) %>%
  mutate(NumLoci = factor(NumLoci, levels = c("500", "3300", "9900"))) %>%
  filter(Demography != "IM")

pdf(file = paste0(here::here(), "/output/FigureS2.pdf"), width=8, height=3.5)

par(mfcol=c(1,3), mar = c(5, 5, 3, 1))
with(Results.sub.long, {interaction.plot(NumInd, NumLoci, Fst,
  col=c("grey1","grey20","grey50"),
  ylab=" mean Fst", ylim=c(0.0,0.015), lwd= 3,
  cex.lab=1.5, cex.axis=1.5, cex.sub=1.5,
  xleg = "bottomright")})
  title("(a)", adj = 0, line =1, cex.main=2 )

with(Results.sub.long, {interaction.plot(NumInd, NumLoci, Adj.Rsq.Pos,
  col=c("grey1","grey20","grey50"),
  ylab=expression("mean adjusted R"^2), ylim=c(0.0,0.6)
  lwd= 3, cex.lab=1.5, cex.axis=1.5, cex.sub=1.5,
  legend=FALSE)})
  title("(b)", adj = 0, line =1, cex.main=2 )

with(Results.sub.long, {interaction.plot(NumInd, NumLoci, Morans.I,
  col=c("grey1","grey20","grey50"),
  ylab="mean Moran's I", ylim=c(0.0,0.6),
  lwd= 3, cex.lab=1.5, cex.axis=1.5, cex.sub=1.5,
  legend=FALSE)})
  title("(c)", adj = 0, line =1, cex.main=2 )

```

```
dev.off()
```

```
## pdf
## 2
```

c) Figure S3

Multipanel fig. one figure for each demography all on the same scale. boxplots (20 ind & 9900 loci)

Respo= Adj R² & MI & scaled MI. (x axis = 60 refitting, missing values, 30 refitting)... show variability as proportion (divide each value by the value of the 90)

```
# Dch Adjusted Rsq.
```

```
D11<- with(Results.table %>% filter(Demography == "IBD" & NumInd == 20),
  data.frame(Comp.60=Dch.RsqAdjPos.drop.m / Dch.RsqAdjPos.90,
    Meta.60=Dch.RsqAdjPos.drop.MEM.m / Dch.RsqAdjPos.90,
    Meta.30=Dch.RsqAdjPos.30.m / Dch.RsqAdjPos.90))
D11<- gather(D11, "Comparison", "Dch.Rsq")
D11.s <- groupwiseMean(Dch.Rsq ~ Comparison, data =D11, conf = 0.95, digits = 3)
D11.s$Comparison <- factor(D11.s$Comparison, levels = c("Comp.60", "Meta.60", "Meta.30"))
D11.a.s<- ggplot(D11.s, aes(x = Comparison, y = Mean)) + ylim(0,1.0) +
  geom_errorbar(aes(ymin = Trad.lower, ymax = Trad.upper), width = 0.05, size = 0.5) +
  geom_point(shape = 1, size = 2) +
  theme_bw() + theme (panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  ggtitle("IBD") + xlab("") + ylab(expression(bold('Adjusted R'^2))) +
  theme(axis.text = element_text(size = 10),
    plot.title = element_text(size=16, face="bold", hjust = 0.5),
    axis.title.y = element_text(size=12, face="bold")
  )

D22<- with(Results.table %>% filter(Demography == "1R" & NumInd == 20),
  data.frame(Comp.60=Dch.RsqAdjPos.drop.m / Dch.RsqAdjPos.90,
    Meta.60=Dch.RsqAdjPos.drop.MEM.m / Dch.RsqAdjPos.90,
    Meta.30=Dch.RsqAdjPos.30.m / Dch.RsqAdjPos.90))
D22<- gather(D22, "Comparison", "Dch.Rsq")
D22.s <- groupwiseMean(Dch.Rsq ~ Comparison, data =D22, conf = 0.95, digits = 3)
D22.s$Comparison <- factor(D22.s$Comparison, levels = c("Comp.60", "Meta.60", "Meta.30"))
D22.a.s<- ggplot(D22.s, aes(x = Comparison, y = Mean)) + ylim(0,1.0) +
  geom_errorbar(aes(ymin = Trad.lower, ymax = Trad.upper), width = 0.05, size = 0.5) +
  geom_point(shape = 1, size = 2) +
  theme_bw() + theme (panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  ggtitle("1R") + xlab("") + ylab("") +
  theme(axis.text = element_text(size = 10),
    plot.title = element_text(size=16, face="bold", hjust = 0.5),
    axis.title.y = element_text(size=12, face="bold")
  )

D33<- with(Results.table %>% filter(Demography == "2R" & NumInd == 20),
  data.frame(Comp.60=Dch.RsqAdjPos.drop.m / Dch.RsqAdjPos.90,
    Meta.60=Dch.RsqAdjPos.drop.MEM.m / Dch.RsqAdjPos.90,
    Meta.30=Dch.RsqAdjPos.30.m / Dch.RsqAdjPos.90))
```

```

D33<- gather(D33, "Comparison", "Dch.Rsq")
D33.s <- groupwiseMean(Dch.Rsq ~ Comparison, data =D33, conf = 0.95, digits = 3)
D33.s$Comparison <- factor(D33.s$Comparison, levels = c("Comp.60", "Meta.60", "Meta.30"))
D33.a.s<- ggplot(D33.s, aes(x = Comparison, y = Mean)) + ylim(0,1.0) +
  geom_errorbar(aes(ymin = Trad.lower, ymax = Trad.upper), width = 0.05, size = 0.5) +
  geom_point(shape = 1, size = 2) +
  theme_bw() + theme (panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  ggtitle("2R") + xlab("") + ylab("") +
  theme(axis.text = element_text(size = 10),
        plot.title = element_text(size=16, face="bold", hjust = 0.5),
        axis.title.y = element_text(size=12, face="bold")
  )

D44<- with(Results.table %>% filter(Demography == "IBD" & NumInd == 20),
  data.frame(Comp.60=Dch.Morans.I.drop.m / Dch.Morans.I.90,
             Meta.60=Dch.Morans.I.drop.MEM.m / Dch.Morans.I.90,
             Meta.30=Dch.Morans.I.30.m / Dch.Morans.I.90))
D44<- gather(D44, "Comparison", "Dch.Rsq")
D44.s <- groupwiseMean(Dch.Rsq ~ Comparison, data =D44, conf = 0.95, digits = 3)
D44.s$Comparison <- factor(D44.s$Comparison, levels = c("Comp.60", "Meta.60", "Meta.30"))
D44.a.s<- ggplot(D44.s, aes(x = Comparison, y = Mean)) + ylim(0,1.0) +
  geom_errorbar(aes(ymin = Trad.lower, ymax = Trad.upper), width = 0.05, size = 0.5) +
  geom_point(shape = 1, size = 2) +
  theme_bw() + theme (panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  xlab("") + ylab("Moran's I") +
  theme(axis.text = element_text(size = 10),
        axis.title.y = element_text(size=12, face="bold")
  )

D55<- with(Results.table %>% filter(Demography == "1R" & NumInd == 20),
  data.frame(Comp.60=Dch.Morans.I.drop.m / Dch.Morans.I.90,
             Meta.60=Dch.Morans.I.drop.MEM.m / Dch.Morans.I.90,
             Meta.30=Dch.Morans.I.30.m / Dch.Morans.I.90))
D55<- gather(D55, "Comparison", "Dch.Rsq")
D55.s <- groupwiseMean(Dch.Rsq ~ Comparison, data =D55, conf = 0.95, digits = 3)
D55.s$Comparison <- factor(D55.s$Comparison, levels = c("Comp.60", "Meta.60", "Meta.30"))
D55.a.s<- ggplot(D55.s, aes(x = Comparison, y = Mean)) + ylim(0,1.0) +
  geom_errorbar(aes(ymin = Trad.lower, ymax = Trad.upper), width = 0.05, size = 0.5) +
  geom_point(shape = 1, size = 2) +
  theme_bw() + theme (panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  xlab("") + ylab("") +
  theme(axis.text = element_text(size = 10),
        axis.title.y = element_text(size=12, face="bold")
  )

D66<- with(Results.table %>% filter(Demography == "2R" & NumInd == 20),
  data.frame(Comp.60=Dch.Morans.I.drop.m / Dch.Morans.I.90,
             Meta.60=Dch.Morans.I.drop.MEM.m / Dch.Morans.I.90,
             Meta.30=Dch.Morans.I.30.m / Dch.Morans.I.90))
D66<- gather(D66, "Comparison", "Dch.Rsq")
D66.s <- groupwiseMean(Dch.Rsq ~ Comparison, data =D66, conf = 0.95, digits = 3)
D66.s$Comparison <- factor(D66.s$Comparison, levels = c("Comp.60", "Meta.60", "Meta.30"))

```

```

D66.a.s<- ggplot(D66.s, aes(x = Comparison, y = Mean)) + ylim(0,1.0) +
  geom_errorbar(aes(ymin = Trad.lower, ymax = Trad.upper), width = 0.05, size = 0.5) +
  geom_point(shape = 1, size = 2) +
  theme_bw() + theme (panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  xlab("") + ylab("") +
  theme(axis.text = element_text(size = 10),
        axis.title.y = element_text(size=12, face="bold")
  )

# Scaled Moran's I

D77<- with(Results.table %>% filter(Demography == "IBD" & NumInd == 20),
  data.frame(Comp.60=Dch.I.scaled.drop.m / Dch.I.scaled.90,
    Meta.60=Dch.I.scaled.drop.MEM.m / Dch.I.scaled.90,
    Meta.30=Dch.I.scaled.30.m / Dch.I.scaled.90))
D77<- gather(D77, "Comparison", "Dch.Rsq")
D77.s <- groupwiseMean(Dch.Rsq ~ Comparison, data =D77, conf = 0.95, digits = 3)
D77.s$Comparison <- factor(D77.s$Comparison, levels = c("Comp.60", "Meta.60", "Meta.30"))
D77.a.s<- ggplot(D77.s, aes(x = Comparison, y = Mean)) + ylim(0,1.0) +
  geom_errorbar(aes(ymin = Trad.lower, ymax = Trad.upper), width = 0.05, size = 0.5) +
  geom_point(shape = 1, size = 2) +
  theme_bw() + theme (panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  xlab("") + ylab("Scaled Moran's I") +
  theme(axis.text = element_text(size = 10),
        axis.title.y = element_text(size=12, face="bold")
  )

D88<- with(Results.table %>% filter(Demography == "1R" & NumInd == 20),
  data.frame(Comp.60=Dch.I.scaled.drop.m / Dch.I.scaled.90,
    Meta.60=Dch.I.scaled.drop.MEM.m / Dch.I.scaled.90,
    Meta.30=Dch.I.scaled.30.m / Dch.I.scaled.90))
D88<- gather(D88, "Comparison", "Dch.Rsq")
D88.s <- groupwiseMean(Dch.Rsq ~ Comparison, data =D88, conf = 0.95, digits = 3)
D88.s$Comparison <- factor(D88.s$Comparison, levels = c("Comp.60", "Meta.60", "Meta.30"))
D88.a.s<- ggplot(D88.s, aes(x = Comparison, y = Mean)) + ylim(0,1.0) +
  geom_errorbar(aes(ymin = Trad.lower, ymax = Trad.upper), width = 0.05, size = 0.5) +
  geom_point(shape = 1, size = 2) +
  theme_bw() + theme (panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  xlab("") + ylab("") +
  theme(axis.text = element_text(size = 10),
        axis.title.y = element_text(size=12, face="bold")
  )

D99<- with(Results.table %>% filter(Demography == "2R" & NumInd == 20),
  data.frame(Comp.60=Dch.I.scaled.drop.m / Dch.I.scaled.90,
    Meta.60=Dch.I.scaled.drop.MEM.m / Dch.I.scaled.90,
    Meta.30=Dch.I.scaled.30.m / Dch.I.scaled.90))
D99<- gather(D99, "Comparison", "Dch.Rsq")
D99.s <- groupwiseMean(Dch.Rsq ~ Comparison, data =D99, conf = 0.95, digits = 3)
D99.s$Comparison <- factor(D99.s$Comparison, levels = c("Comp.60", "Meta.60", "Meta.30"))
D99.a.s<- ggplot(D99.s, aes(x = Comparison, y = Mean)) + ylim(0,1.02) +
  geom_errorbar(aes(ymin = Trad.lower, ymax = Trad.upper), width = 0.05, size = 0.5) +
  geom_point(shape = 1, size = 2) +

```

```

theme_bw() + theme (panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  xlab("") + ylab("") +
  theme(axis.text = element_text(size = 10),
        axis.title.y = element_text(size=12, face="bold")
  )

figure.3.P <- ggarrange(D11.a.s, D22.a.s, D33.a.s, D44.a.s, D55.a.s, D66.a.s ,D77.a.s, D88.a.s, D99.a.s,
#figure.3.P

g <- arrangeGrob(D11.a.s, D22.a.s, D33.a.s, D44.a.s, D55.a.s, D66.a.s ,D77.a.s, D88.a.s, D99.a.s, ncol = 3)
ggsave(file=paste0(here::here(), "/output/FigureS3.pdf"), g)

## Saving 6 x 6 in image

```