

# Untitled

KE Lotterhos

9/18/2020

## Calculating environment distances

- 1) Create 3 fake environments
  - Correlated with Env 1 (~0.5)
  - Correlated with Env 2 (~0.5)
  - A random environment with mean 0 and sd 1
- 2) Make sure each environment is standardized by subtracting the mean and dividing by the sd. (This should also be true for what is being put into GF)
- 3) Calculate the following between all populations (although technically, only needed for core and edge populations):
  - Euclidean distance for selective environments
  - Mahalanobis distance for selective environments
  - Euclidean distance for ALL environments
  - Mahalanobis distance for ALL environments

```
CGfit <- read.csv("Common_Garden_fit.csv")
Popsenv <- read.csv("Pops_env.csv")
```

```
head(CGfit)
```

##	Transplant	Home	Fitness	D_CI	D_CI_sel	Env_sel1	Env_sel2	dM
## 1	T1	H1	0.790825	0.000000000	0.00000000	-1	-1	3.9996
## 2	T2	H1	0.817109	0.003351372	0.01194520	-1	-1	3.9996
## 3	T3	H1	0.711378	0.005807787	0.02071331	-1	-1	3.9996
## 4	T4	H1	0.521529	0.012116530	0.04544032	-1	-1	3.9996
## 5	T5	H1	0.321762	0.016701773	0.05602929	-1	-1	3.9996
## 6	T6	H1	0.321762	0.016701773	0.05602929	-1	-1	3.9996

```
head(Popsenv)
```

##	Pop	envPop1	envPop2
## 1	P1	-1.0	-1
## 2	P2	-0.5	-1
## 3	P3	0.0	-1
## 4	P4	0.5	-1
## 5	P5	1.0	-1
## 6	P6	1.0	-1

## Create fake environments

```
fakeEnv1 <- Popsenv$envPop1 + rnorm(nrow(Popsenv),0,1.3)
# this standard deviation generally produces a correlation between 0.3 and 0.6
cor(Popsenv$envPop1, fakeEnv1)
```

```
## [1] 0.5616083
```

```
fakeEnv2 <- Popsenv$envPop2 + rnorm(nrow(Popsenv),0,1.3)
# this standard deviation generally produces a correlation between 0.3 and 0.6
cor(Popsenv$envPop2, fakeEnv2)
```

```
## [1] 0.3862077
```

```
fakeEnv3 <- rnorm(nrow(Popsenv),0,1)
```

```
Popsenv$fakeEnv1 <- fakeEnv1
Popsenv$fakeEnv2 <- fakeEnv2
Popsenv$fakeEnv3 <- fakeEnv3
```

## Standardize environments

```
head(Popsenv)
```

```
##   Pop envPop1 envPop2  fakeEnv1  fakeEnv2  fakeEnv3
## 1  P1     -1.0     -1  0.1527746 -1.6615225 -0.01140918
## 2  P2     -0.5     -1 -1.0136145 -1.6879409 -0.02823955
## 3  P3      0.0     -1 -2.7328556 -0.5389532  0.47746168
## 4  P4      0.5     -1  1.2059191 -0.8469736 -0.69761620
## 5  P5      1.0     -1  2.6030965 -0.9246118  0.96673902
## 6  P6      1.0     -1  0.6163429 -2.0746797 -0.95398558
```

```
means <- colMeans(Popsenv[2:6])
# beware of hard coding columns here
sds <- apply(Popsenv[2:6], 2, sd)
```

```
PopsenvStnd <- Popsenv
for (i in 2:6){
  PopsenvStnd[,i] <- (Popsenv[,i] - means[i-1])/sds[i-1]
}
head(PopsenvStnd)
```

```
##   Pop   envPop1  envPop2   fakeEnv1  fakeEnv2  fakeEnv3
## 1  P1 -1.4071247 -1.407125  0.08864111 -1.0259992  0.08492709
## 2  P2 -0.7035624 -1.407125 -0.75098240 -1.0430964  0.06844054
## 3  P3  0.0000000 -1.407125 -1.98857549 -0.2995067  0.56381119
## 4  P4  0.7035624 -1.407125  0.84674562 -0.4988481 -0.58726194
## 5  P5  1.4071247 -1.407125  1.85250164 -0.5490931  1.04309346
## 6  P6  1.4071247 -1.407125  0.42234001 -1.2933819 -0.83839414
```

```
# Check for mistakes
round(colMeans(PopsenvStnd[2:6]))
```

```
##   envPop1  envPop2 fakeEnv1 fakeEnv2 fakeEnv3
##         0         0         0         0         0
```

```
round(apply(PopsenvStnd[2:6], 2, sd))
```

```
##   envPop1  envPop2 fakeEnv1 fakeEnv2 fakeEnv3
##         1         1         1         1         1
```

## Understand CG fit

In this dataframe, it appears **Home** is the site of the common garden. **Transplant** is the location that the genotype came from **Fitness** is the average fitness of the individuals from the source location

D\_CI is GF\_offset\_genome

D\_CI\_sel is GF\_offset for the causal loci

```
head(CGfit)
```

```
##   Transplant Home  Fitness      D_CI  D_CI_sel Env_sel1 Env_sel2      dM
## 1          T1   H1 0.790825 0.000000000 0.00000000      -1      -1 3.9996
## 2          T2   H1 0.817109 0.003351372 0.01194520      -1      -1 3.9996
## 3          T3   H1 0.711378 0.005807787 0.02071331      -1      -1 3.9996
## 4          T4   H1 0.521529 0.012116530 0.04544032      -1      -1 3.9996
## 5          T5   H1 0.321762 0.016701773 0.05602929      -1      -1 3.9996
## 6          T6   H1 0.321762 0.016701773 0.05602929      -1      -1 3.9996
```

## Understanding Mahalanobis

?mahalanobis We are interested in calculating the Mahalanobis distance between pop1 and pop2, while controlling for the covariance among the environmental variables in the population

Let's look at an example where we take the Md between population 1 and population 50, for all the environments

```
(envpop1 <- PopsenvStnd[1,2:6])
```

```
##      envPop1  envPop2  fakeEnv1  fakeEnv2  fakeEnv3
## 1 -1.407125 -1.407125 0.08864111 -1.025999 0.08492709
```

```
(envpop2 <- PopsenvStnd[50,2:6])
```

```
##      envPop1  envPop2  fakeEnv1  fakeEnv2  fakeEnv3
## 50 -1.407125 1.407125 -0.4140336 -1.175834 -1.791025
```

```
# We calculate the covariance based on the entire landscape:
```

```
cov_allEnv <- cov(PopsenvStnd[,2:6])
round(cov_allEnv,2)
```

```
##      envPop1  envPop2  fakeEnv1  fakeEnv2  fakeEnv3
## envPop1      1.00      0.00      0.56      0.14     -0.13
## envPop2      0.00      1.00     -0.02      0.39     -0.06
## fakeEnv1      0.56     -0.02      1.00      0.03      0.12
## fakeEnv2      0.14      0.39      0.03      1.00     -0.11
## fakeEnv3     -0.13     -0.06      0.12     -0.11      1.00
```

```
mahalanobis(as.numeric(envpop1),
            as.numeric(envpop2),
            cov_allEnv)
```

```
## [1] 13.21649
```

```
# sanity check
```

```
mahalanobis(as.numeric(envpop1),
            as.numeric(envpop1),
            cov_allEnv)
```

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

```
#compare to eucl.
dist(rbind(envpop1, envpop2))
```

```
##          1
## 50 3.42262
```

## Calculate environment distances

```
cov_allEnv <- cov(PopsenvStd[,2:6])
cov_selEnv <- cov(PopsenvStd[,2:3])
# BEWARE HARD CODING HERE for selective envs.
```

```
head(PopsenvStd)
```

```
##   Pop   envPop1  envPop2   fakeEnv1  fakeEnv2  fakeEnv3
## 1  P1 -1.4071247 -1.407125  0.08864111 -1.0259992  0.08492709
## 2  P2 -0.7035624 -1.407125 -0.75098240 -1.0430964  0.06844054
## 3  P3  0.0000000 -1.407125 -1.98857549 -0.2995067  0.56381119
## 4  P4  0.7035624 -1.407125  0.84674562 -0.4988481 -0.58726194
## 5  P5  1.4071247 -1.407125  1.85250164 -0.5490931  1.04309346
## 6  P6  1.4071247 -1.407125  0.42234001 -1.2933819 -0.83839414
```

```
CGfit$EdSelEnv <- NA
# Euclidean distance for selective environments
```

```
CGfit$MdSelEnv <- NA
# Mahalanobis distance for selective environments
```

```
CGfit$EdAllEnv <- NA
# Euclidean distance for ALL environments
```

```
CGfit$MdAllEnv <- NA
# Mahalanobis distance for ALL environments
```

```
head(CGfit)
```

```
##   Transplant Home  Fitness      D_CI  D_CI_sel Env_sel1 Env_sel2    dM
## 1         T1   H1  0.790825  0.000000000  0.00000000      -1      -1  3.9996
## 2         T2   H1  0.817109  0.003351372  0.01194520      -1      -1  3.9996
## 3         T3   H1  0.711378  0.005807787  0.02071331      -1      -1  3.9996
## 4         T4   H1  0.521529  0.012116530  0.04544032      -1      -1  3.9996
## 5         T5   H1  0.321762  0.016701773  0.05602929      -1      -1  3.9996
## 6         T6   H1  0.321762  0.016701773  0.05602929      -1      -1  3.9996
##   EdSelEnv MdSelEnv EdAllEnv MdAllEnv
## 1      NA      NA      NA      NA
## 2      NA      NA      NA      NA
## 3      NA      NA      NA      NA
## 4      NA      NA      NA      NA
## 5      NA      NA      NA      NA
## 6      NA      NA      NA      NA
```

```
for (i in 1:nrow(CGfit)){
  # get the row in PopsenvStd for the transplant genotype
  row1 = which(PopsenvStd==gsub("T", "P", as.character(CGfit$Transplant[i])))
  # get the row in PopsenvStd for the common garden location
```

```

row2 = which(PopsenvStd==gsub("H","P",as.character(CGfit$Home[i])))

# Look up the envi
(envpop1_all <- PopsenvStd[row1,2:6])
(envpop2_all <- PopsenvStd[row2,2:6])
# Look up the envi
(envpop1_sel <- PopsenvStd[row1,2:3])
(envpop2_sel <- PopsenvStd[row2,2:3])
# BEWARE HARD CODING

### Calculate the environmental distance between the two rows

CGfit$EdSelEnv[i] <- dist(rbind(envpop1_sel,
                               envpop2_sel))
# Euclidean distance for selective environments

CGfit$MdSelEnv[i] <- mahalanobis(as.numeric(envpop1_sel),
                                as.numeric(envpop2_sel),
                                cov_selEnv)
# Mahalanobis distance for selective environments

CGfit$EdAllEnv[i] <- dist(rbind(envpop1_all,
                               envpop2_all))
# Euclidean distance for ALL environments

CGfit$MdAllEnv[i] <- mahalanobis(as.numeric(envpop1_all),
                                as.numeric(envpop2_all),
                                cov_allEnv)
# Mahalanobis distance for ALL environments
}

```

## Calculate environment distances

```
head(CGfit)
```

```

##   Transplant Home  Fitness      D_CI  D_CI_sel Env_sel1 Env_sel2      dM
## 1          T1   H1 0.790825 0.000000000 0.00000000      -1      -1 3.9996
## 2          T2   H1 0.817109 0.003351372 0.01194520      -1      -1 3.9996
## 3          T3   H1 0.711378 0.005807787 0.02071331      -1      -1 3.9996
## 4          T4   H1 0.521529 0.012116530 0.04544032      -1      -1 3.9996
## 5          T5   H1 0.321762 0.016701773 0.05602929      -1      -1 3.9996
## 6          T6   H1 0.321762 0.016701773 0.05602929      -1      -1 3.9996
##   EdSelEnv MdSelEnv EdAllEnv  MdAllEnv
## 1 0.0000000 0.000 0.000000 0.000000
## 2 0.7035624 0.495 1.095688 2.951265
## 3 1.4071247 1.980 2.655551 16.587568
## 4 2.1106871 4.455 2.399885 4.843426
## 5 2.8142495 7.920 3.489517 9.746331
## 6 2.8142495 7.920 2.992553 10.945709

```

```

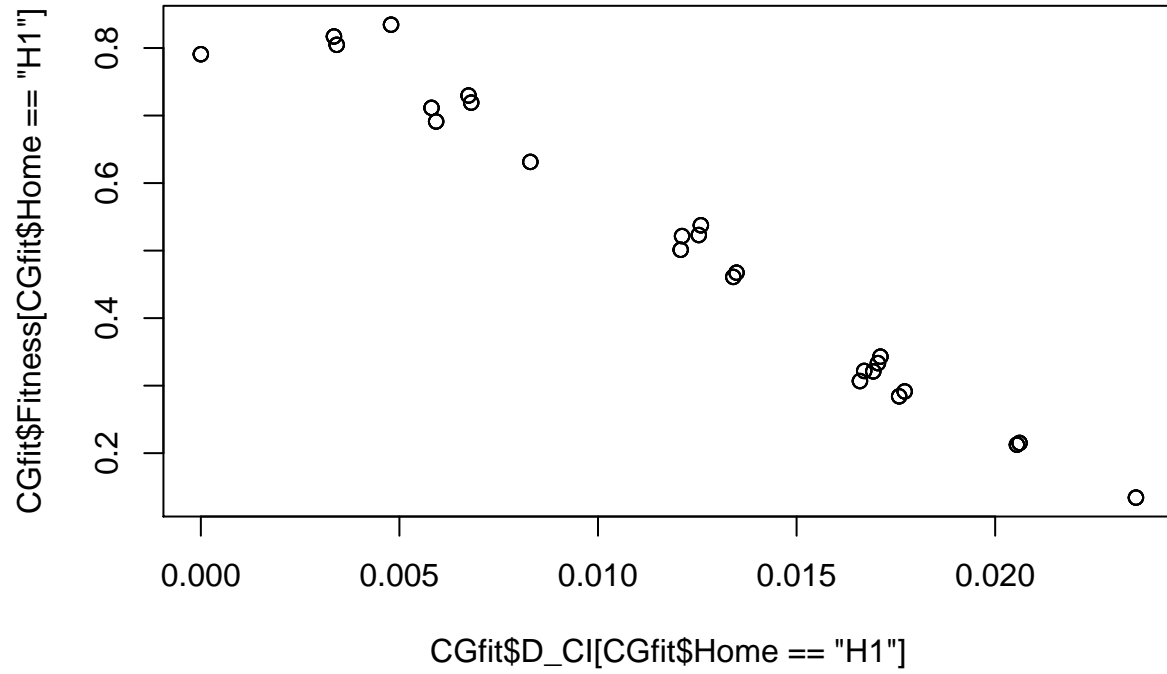
#any missing data?
sum(!complete.cases(CGfit))

```

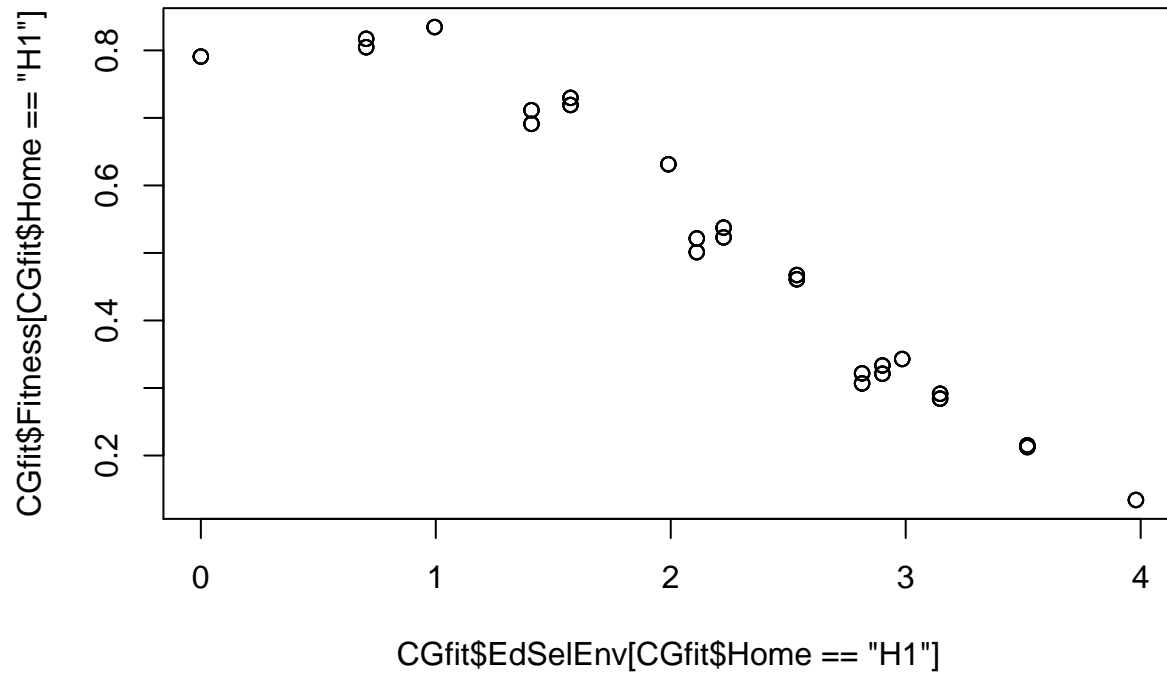
```
## [1] 0
```

```
# should be 0
```

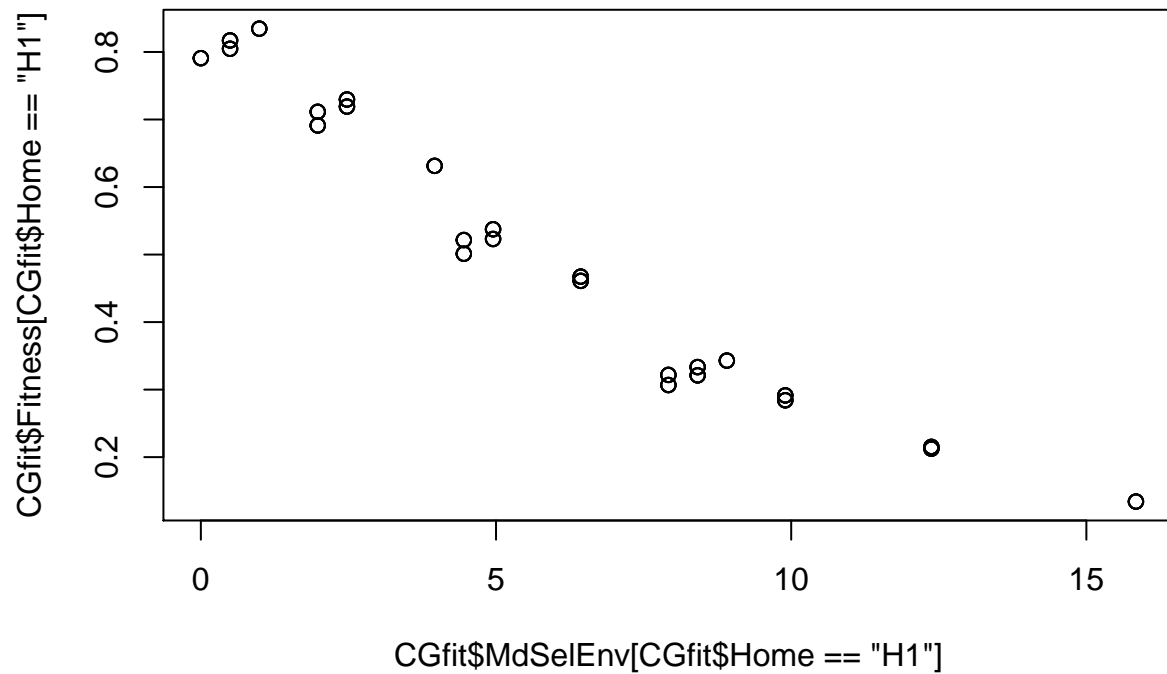
```
plot(CGfit$Fitness[CGfit$Home=="H1"] ~  
      CGfit$D_CI[CGfit$Home=="H1"])
```



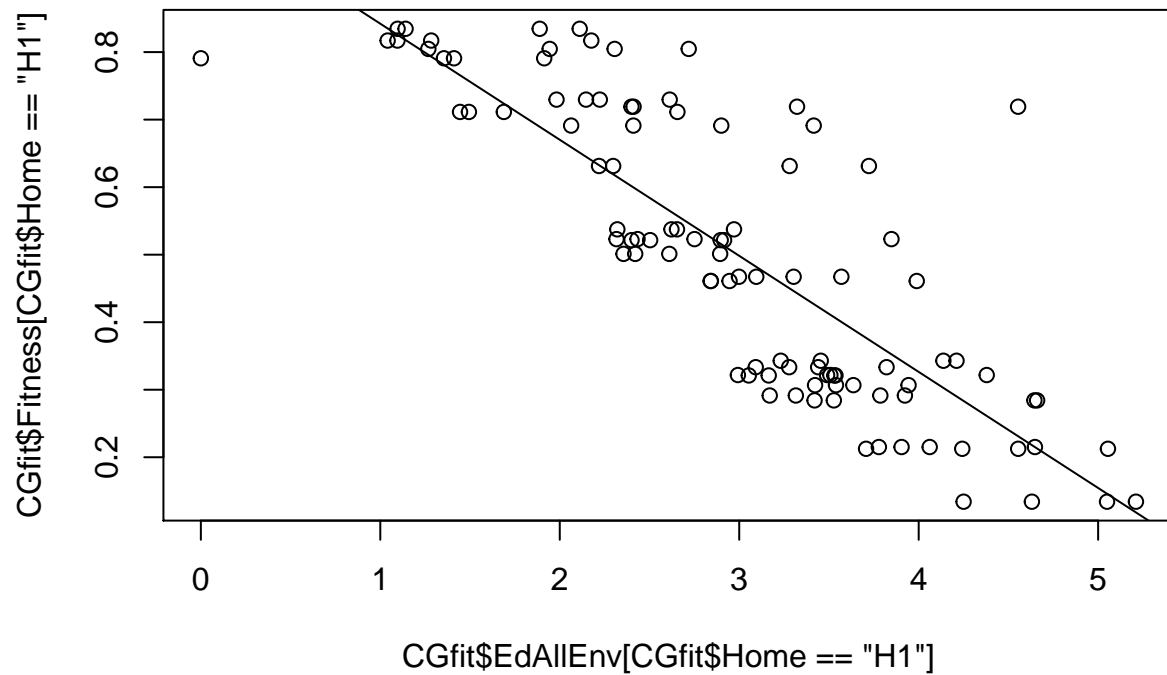
```
plot(CGfit$Fitness[CGfit$Home=="H1"] ~  
      CGfit$EdSelEnv[CGfit$Home=="H1"])
```



```
plot(CGfit$Fitness[CGfit$Home=="H1"] ~  
      CGfit$MdSelEnv[CGfit$Home=="H1"])
```



```
plot(CGfit$Fitness[CGfit$Home=="H1"] ~
      CGfit$EdAllEnv[CGfit$Home=="H1"])
abline(lm(CGfit$Fitness[CGfit$Home=="H1"] ~
           CGfit$EdAllEnv[CGfit$Home=="H1"])))
```

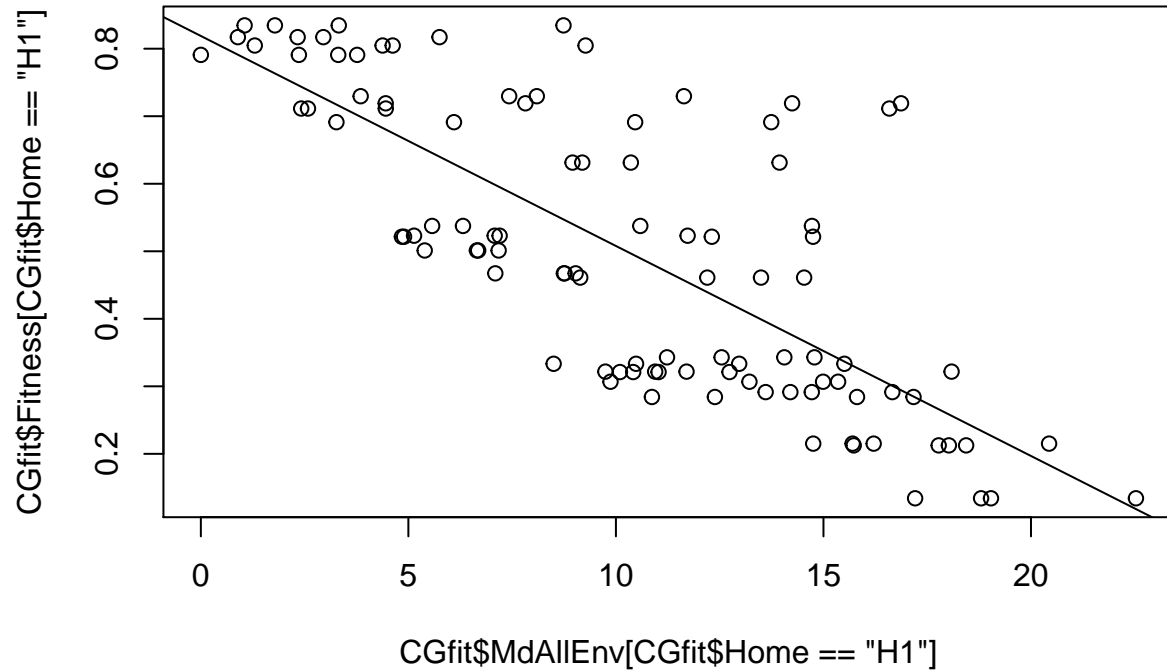


```
(cor(CGfit$Fitness[CGfit$Home=="H1"],
      CGfit$EdAllEnv[CGfit$Home=="H1"])))
```

```
## [1] -0.8243983
```

```
plot(CGfit$Fitness[CGfit$Home=="H1"] ~
      CGfit$MdAllEnv[CGfit$Home=="H1"])
```

```
abline(lm(CGfit$Fitness[CGfit$Home=="H1"] ~
  CGfit$MdAllEnv[CGfit$Home=="H1"]))
```



```
cor(CGfit$Fitness[CGfit$Home=="H1"],
  CGfit$MdAllEnv[CGfit$Home=="H1"])
```

```
## [1] -0.7592982
```

Results for Euclidean Dist and Mahalanobis are similar because we standardize the environments to have an SD=1 prior to analysis