

Behavior of PL metric

KE Lotterhos

10/7/2020

setwd("~/Users/lotterhos/Documents/GitHub/CnGV/notebook")

Description of PL metric

The PL metric was developed in Stamp and Hadfield 2020 (hereafter SH2020, doi: 10.1111/ele.13565). The metric can be calculated for a 2 population x 2 environment reciprocal transplant scenario (reciprocal transplant experiments are those in which two populations are assayed in their own and each other's environment). They used the metric to test whether phenotypic differences between populations are due to genetic differences or a plastic response to environmental variation. They also used the metric to test the prevalence of co-gradient vs. countergradient variation in the datasets. It is this latter application of the PL metric that we explore here.

The *in situ* divergence is the difference in mean phenotypes of two populations raised in their home environments ($P_A E_A - P_B E_B$, where $P_i E_j$ refers to the average phenotype of individuals from population i raised in the environment of population of j).

They define PL as:

$$PL = \Delta E / \Delta H$$

where

$$\Delta E_A = P_A E_A - P_A E_B$$

In SH2020 they state “the difference in phenotype between individuals from Population B in Environment A ($P_B E_A$) and from Population B in Environment B ($P_B E_B$) can be ascribed to plasticity (ΔE_B),” so it is a little ambiguous whether they mean $P_B E_B - P_B E_A$ or $P_B E_A - P_B E_B$. We use the latter definition because it would give $PL = 1$ under a “perfect plasticity” scenario, and would be consistent with the outcomes and interpretation in the paper.

$$\Delta E_B = P_B E_A - P_B E_B$$

$$\Delta E = (\Delta E_A + \Delta E_B) / 2 = ((P_A E_A - P_A E_B) + (P_B E_A - P_B E_B)) / 2$$

$$\Delta H = P_A E_A - P_B E_B$$

They also define:

$\Delta A = P_B E_A - P_A E_B$ (the estimated difference in the two populations phenotypes when away in each other's environment)

Interpretation of PL:

SH2020 interpret the plasticity metric $PL = \Delta E / \Delta H$ as lying between zero and one if there is co-gradient variation, negative if there is wrong-sign plasticity, or greater than one if there is hyperplasticity.

- $PL > 1$ countergradient with hyperplasticity
- $0 < PL < 1$ cgradient
- $PL = 1$ is interpreted as perfect plasticity (Figure 4)

- $PL < 1$ countegradient with wrong-sign plasticity

$$PL = \Delta E / \Delta H = (((P_A E_A - P_A E_B) + (P_B E_A - P_B E_B)) / 2) / (P_A E_A - P_B E_B)$$

When 100% of the divergence is due to plasticity with no genetic differentiation, in Environment A both populations would have the same trait value ($P_B E_A = P_A E_A$) and in Environment B both populations would have the same trait value ($P_A E_B = P_B E_B$)

In this case, plugging in the equivalencies to the PL equations shows that the numerator of PL reduces to 1:
 $PL = (((P_A E_A - P_B E_B) + (P_A E_A - P_B E_B)) / 2) / (P_A E_A - P_B E_B) = (2(P_A E_A - P_B E_B) / 2) / (P_A E_A - P_B E_B) = 1$

Therefore, values of $PL = 1$ are interpreted as “perfect plasticity” in the paper. (Below, we show that scenarios with GxE can also give $PL = 1$)

Function

This function takes a vector of the (hypothetical) measured phenotypes from a 2x2 reciprocal transplant and plots them with the calculated PL metric.

```
makeplot <- function(a, letr, main){
  # a is vector of "PAEA", "PAEB", "PBEA", "PBEB"

  plot(0:1, c(a[1:2]), ylim=c(-2,2),
       xaxt="n", type="l", col="blue", xlab="",
       ylab="Stnd. phenotype", bty="l",
       main=main, lwd=2, xlim=c(0,1.5))
  mtext("EA", side=1, line=1, adj=0, col="blue")
  points(0:1, c(a[3:4]), type="l", col="green", xlab="", lwd=1, lty=2)
  mtext("EB", side=1, line=1, adj=0.66, col="green")

  D_H <- a[1] - a[4]
  (D_EA <- a[1]-a[2])
  #(D_EB <- a[3]-a[4])
  (D_EB <- a[4]-a[3])
  (D_E <- mean(c(D_EA, D_EB)))

  text(1.1,a[1],"D_H", cex=0.8)
  arrows(1.2,a[4], 1.2, a[1], length=0.1)

  text(1.4, a[1], "D_E", cex=0.8)
  arrows(1.3, mean(c(a[2], a[4])),
        1.3, mean(c(a[2], a[4]))+D_E,
        length=0.1)

  PL <- D_E/D_H
  text(1,-1.9,paste("PL =",PL),adj=1)

  text(0,2, letr)
  return(data.frame(D_H=D_H, D_EA = D_EA,
                    D_EB=D_EB,D_E=D_E, PL=PL))
}
```

Cases 1 and 2: drastically different PL metrics for similar reaction norms

These two cases show how slight differences in the mean fitness in the home population can give extremely opposite values of the PL statistic.

- PAEA is the phenotype value for genotype A in Environment A
- PAEB is the phenotype value for genotype A in Environment B
- PBEA is the phenotype value for genotype B in Environment A
- PBEB is the phenotype value for genotype B in Environment B

Case 1 and Case 2 are both countergradient scenarios, with almost the exact same reaction norms. In both cases, PAEA = 0 and PAEB = 1.

However, there are very small differences in PBEA and PBEB, which generate large differences in the PL metric. In Case 1 PBEA = -0.99 while in Case 2 PBEA = -1.01 (a difference of -0.02), and in Case 1 PBEB = 0.01 while in Case 2 PBEB = -0.01 (a difference of +0.02).

```
# Case 1
case1 <- rep(NA, 4)
names(case1) <- c("PAEA", "PAEB", "PBEA", "PBEB")
a <- 0
case1[1:4] <- c(a, a+1, a-1+0.01, a+0.01)
case1
```

```
## PAEA PAEB PBEA PBEB
## 0.00 1.00 -0.99 0.01
```

```
# Case 2
case2 <- rep(NA, 4)
names(case2) <- names(case1)
a <- 0
case2[1:4] <- c(a, a+1, a-1-0.01, a-0.01)
case2
```

```
## PAEA PAEB PBEA PBEB
## 0.00 1.00 -1.01 -0.01
```

The following plot shows how Case 1 and Case 2 have almost equivalent reaction norms, and therefore similar plastic responses to the environment and similar amount of countergradient variation.

Case 1 has PL=100 (implying extreme hyperplasticity) while Case 2 has PL = -100 (implying extreme wrong-sign plasticity).

Note that the difference in means for the two cases are very small and are within the range expected by sampling error, but the PL metric implies that they are very different cases.

```
par(mfrow=c(2,1))
makeplot(case1, "", "Case 1")
```

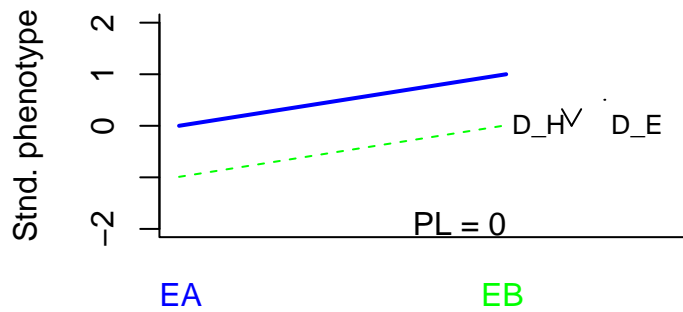
```
## Warning in arrows(1.3, mean(c(a[2], a[4])), 1.3, mean(c(a[2], a[4])) + D_E, :
## zero-length arrow is of indeterminate angle and so skipped
```

```
##      D_H D_EA D_EB D_E PL
## PAEA -0.01  -1    1    0  0
```

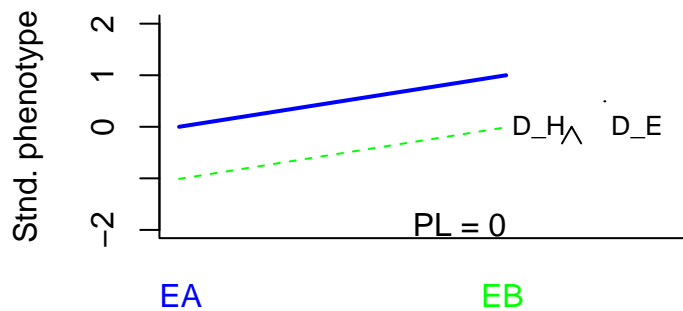
```
makeplot(case2, "", "Case 2")
```

```
## Warning in arrows(1.3, mean(c(a[2], a[4])), 1.3, mean(c(a[2], a[4])) + D_E, :
## zero-length arrow is of indeterminate angle and so skipped
```

Case 1



Case 2



```
##      D_H D_EA D_EB D_E PL
## PAEA 0.01  -1    1  0  0
```

Cases 3 and 4: PL = 1 for GxE reaction norms

In Cases 3 and 4, we compare two very different patterns of reaction norms that both give PL=1. In both cases, the “A” genotype is plastic, and the “B” genotype has little or no plasticity.

As you can see in the figures below, in each case there is a pattern of genetic x environment interaction on the trait value:

```
case3 <- rep(NA, 4)
names(case3) <- names(case1)
case3[1:4] <- c(1,-1,0,0)
case3
```

```
## PAEA PAEB PBEA PBEB
##    1  -1    0    0
```

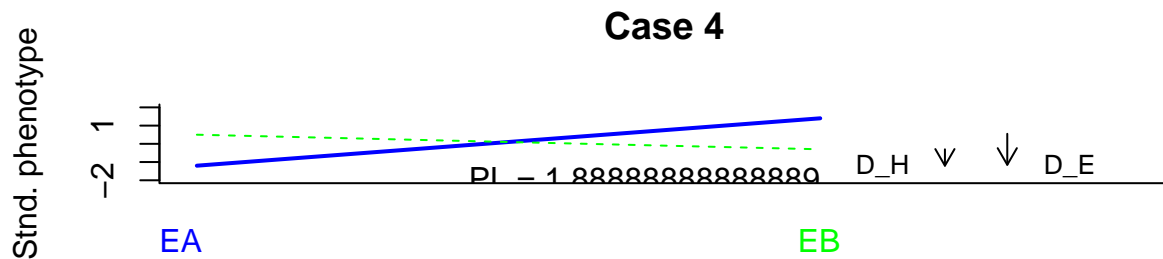
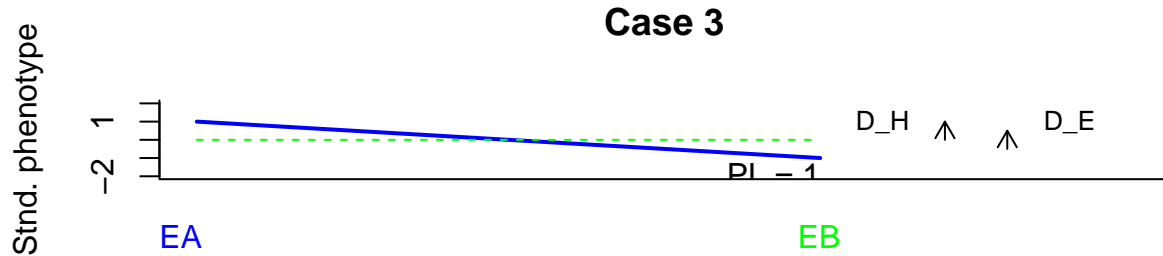
```
case4 <- rep(NA, 4)
names(case4) <- names(case1)
case4[1:4] <- c(-1.2, +1.4, +0.5, -0.3)
case4
```

```
## PAEA PAEB PBEA PBEB
## -1.2  1.4  0.5 -0.3
```

```
par(mfrow=c(2,1))
makeplot(case3, "", "Case 3")
```

```
##      D_H D_EA D_EB D_E PL
## PAEA  1   2   0   1   1
```

```
makeplot(case4, "", "Case 4")
```



```
##      D_H D_EA D_EB D_E PL
## PAEA -0.9 -2.6 -0.8 -1.7 1.888889
```

In the paper, $PL = 1$ is interpreted as “perfect plasticity” (which is interpreted as all the divergence due to plasticity), however these conceptual examples show that GxE reaction norms can also give $PL=1$ due to the averaging of the ΔE_A and ΔE_B .

PL = 1 for “perfect” plasticity scenario

In Cases 5 and 6, we show that are function also produces $PL=1$ in the perfect plasticity scenario. In both cases, there is no differentiation between the genotypes (both genotypes have the same reaction norms).

In Case 5, there is almost no plasticity as there is a very small change in the phenotype from one environment to the other.

In Case 6, there is a lot of plasticity because there is a relatively large change in the phenotype from one environment to the other.

Both these cases are considered “perfect plasticity” under the PL framework, since trait value of the local genotype is equal to the trait value of the foreign genotype in both environments.

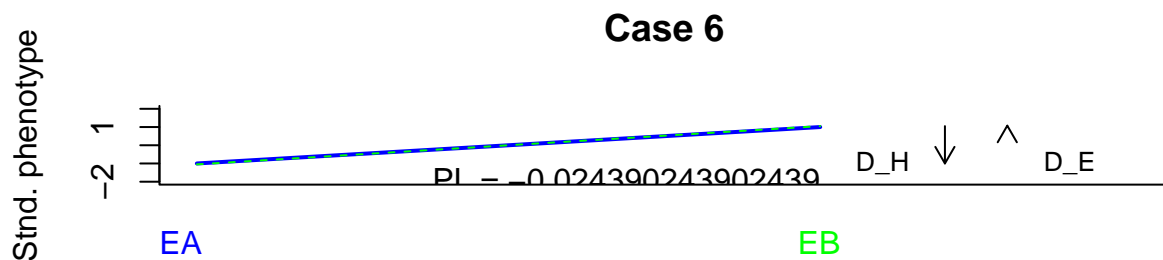
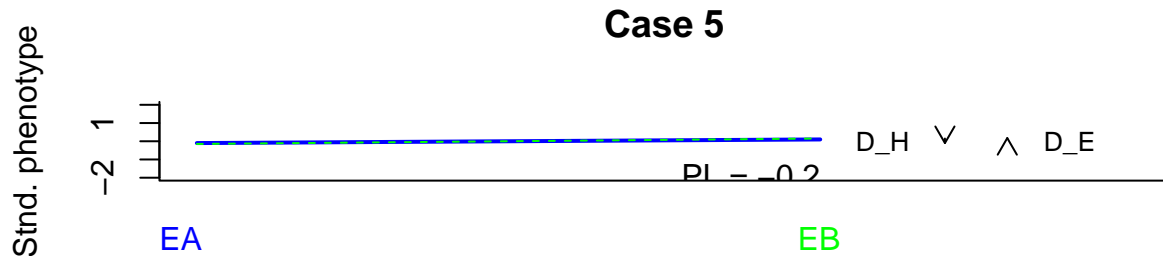
```
case5 <- rep(NA, 4)
names(case5) <- names(case1)
case6 <- case5

case5[1:4] <- c(-0.1, 0.1, -0.15, 0.15)
case6[1:4] <- c(-1, 1, -1.05, 1.05)
```

```
par(mfrow=c(2,1))
makeplot(case5, "", "Case 5")
```

```
##          D_H D_EA D_EB D_E PL
## PAEA -0.25 -0.2  0.3 0.05 -0.2
```

```
makeplot(case6, "", "Case 6")
```



```
##          D_H D_EA D_EB D_E PL
## PAEA -2.05  -2   2.1 0.05 -0.02439024
```

Summary of PL behavior

Under countergradient variation, PL is bimodal and can give extremely different values for datasets with very similar reaction norms (Cases 1-2).

PL can also give the same value for very different patterns of reaction norms, varying from those with high genetic variation and GxE (Cases 3-4) to those with no genetic differentiation (Cases 5-6).

```
final_1 <- rbind(case1, case2, case3, case4, case5, case6)
```

```
# Make plot
#pdf("")
par(mfrow=c(3,2), mar=c(3,1,1,1), oma=c(0,3,0,0))
a <- makeplot(case1, "A", "Case 1")
```

```
## Warning in arrows(1.3, mean(c(a[2], a[4])), 1.3, mean(c(a[2], a[4])) + D_E, :
## zero-length arrow is of indeterminate angle and so skipped
```

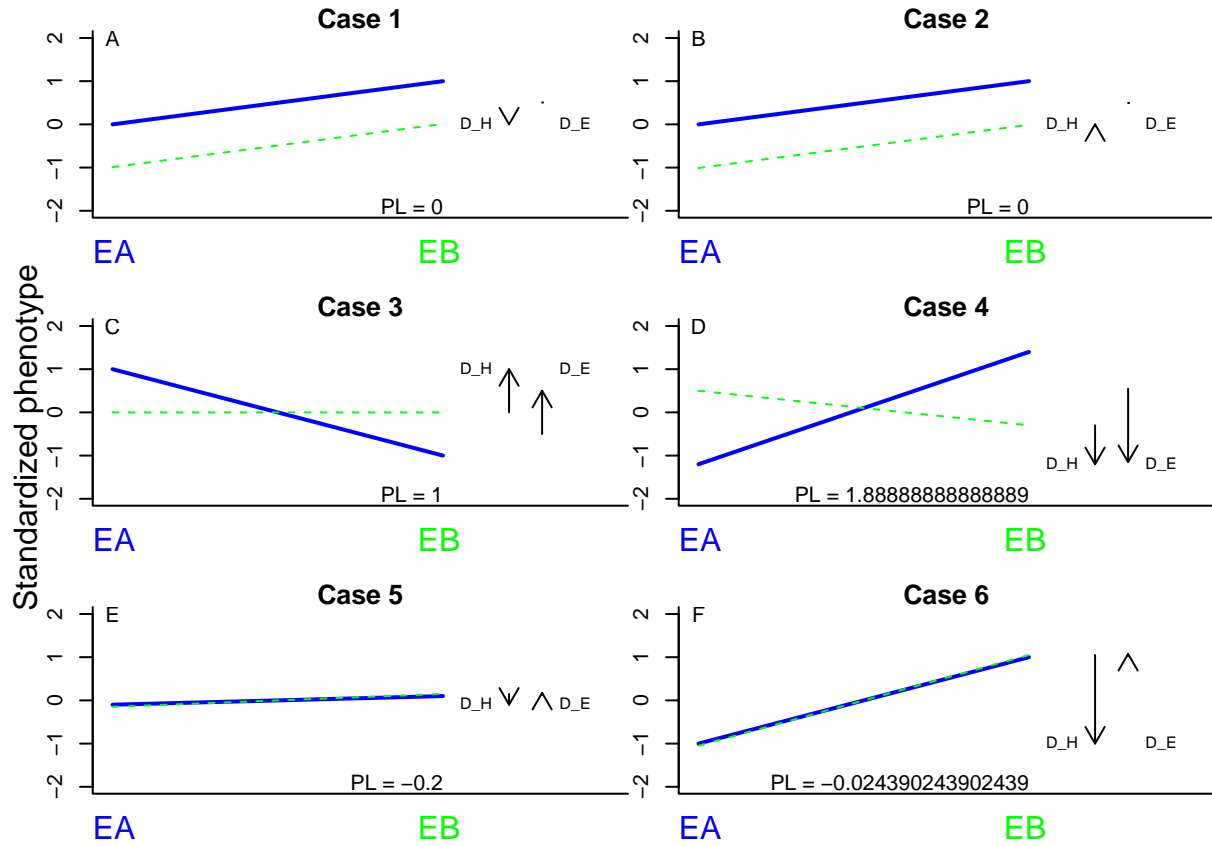
```
b <- makeplot(case2, "B", "Case 2")
```

```
## Warning in arrows(1.3, mean(c(a[2], a[4])), 1.3, mean(c(a[2], a[4])) + D_E, :
## zero-length arrow is of indeterminate angle and so skipped
```

```

c <- makeplot(case3, "C", "Case 3")
d <- makeplot(case4, "D", "Case 4")
e <- makeplot(case5, "E", "Case 5")
f <- makeplot(case6, "F", "Case 6")
mtext("Standardized phenotype", side=2,
      outer=TRUE, line=1)

```



```
#dev.off()
```

```
final_2 <- rbind(a,b,c,d,e, f)
```

```
(final <- cbind(final_1, final_2))
```

```

##      PAEA PAEB PBEA PBEB  D_H D_EA D_EB  D_E      PL
## case1  0.0  1.0 -0.99  0.01 -0.01 -1.0  1.0  0.00  0.00000000
## case2  0.0  1.0 -1.01 -0.01  0.01 -1.0  1.0  0.00  0.00000000
## case3  1.0 -1.0  0.00  0.00  1.00  2.0  0.0  1.00  1.00000000
## case4 -1.2  1.4  0.50 -0.30 -0.90 -2.6 -0.8 -1.70  1.88888889
## case5 -0.1  0.1 -0.15  0.15 -0.25 -0.2  0.3  0.05 -0.20000000
## case6 -1.0  1.0 -1.05  1.05 -2.05 -2.0  2.1  0.05 -0.02439024

```

Compare PL to CovGE

We compare the PL metric to the CovGE metric from Albecker, Trussell, and Lotterhos 2022 (hereafter ATL2022 <https://doi.org/10.1111/ele.14020>) to show that the two are related.

The PL metric can only be calculated for a 2x2 reciprocal transplant, however the CovGE metric can be calculated for any reciprocal transplant experiment as well as a specifically designed common garden

experiment with at least 2 common gardens and multiple genotypes (see ATL2022 for details.)

First, we load a complete results of the simulations from Albecker, which included a calculation of the PL metric.

```
powdf <- read.csv("../results/Results_10.06.2020/Archive/Power_output_results.csv")
vardf <- read.csv("../results/Results_10.06.2020/Archive/Variance_output_results.csv")
PLdf <- read.csv("../results/Results_10.06.2020/Archive/PL_output_results.csv")
phendf <- read.csv("~/Desktop/phenotype_output_results.csv")

#head(powdf)
#head(vardf)
#head(PLdf)
#head(phendf)
```

First, we have to subset the data to cases with 2 genotypes and 2 environments because SH2020 only derived the PL metric for those cases. However, note that CovGE can be calculated for many different designs.

```
cases <- powdf$row[powdf$n_env==2 & powdf$n_pop==2]
length(cases)
```

```
## [1] 800
```

There are 800 cases that met this criteria in our simulations.

Let's subset the data to those cases:

```
powdf2 <- powdf[powdf$row %in% cases,]
vardf2 <- vardf[vardf$row %in% cases,]

#dim(powdf2)
#head(PLdf)

newdf <- merge(powdf2, PLdf, by = "row")
#dim(newdf)
#str(newdf)

newdf$G1E1mean <- NA
newdf$G1E2mean <- NA
newdf$G2E1mean <- NA
newdf$G2E2mean <- NA
```

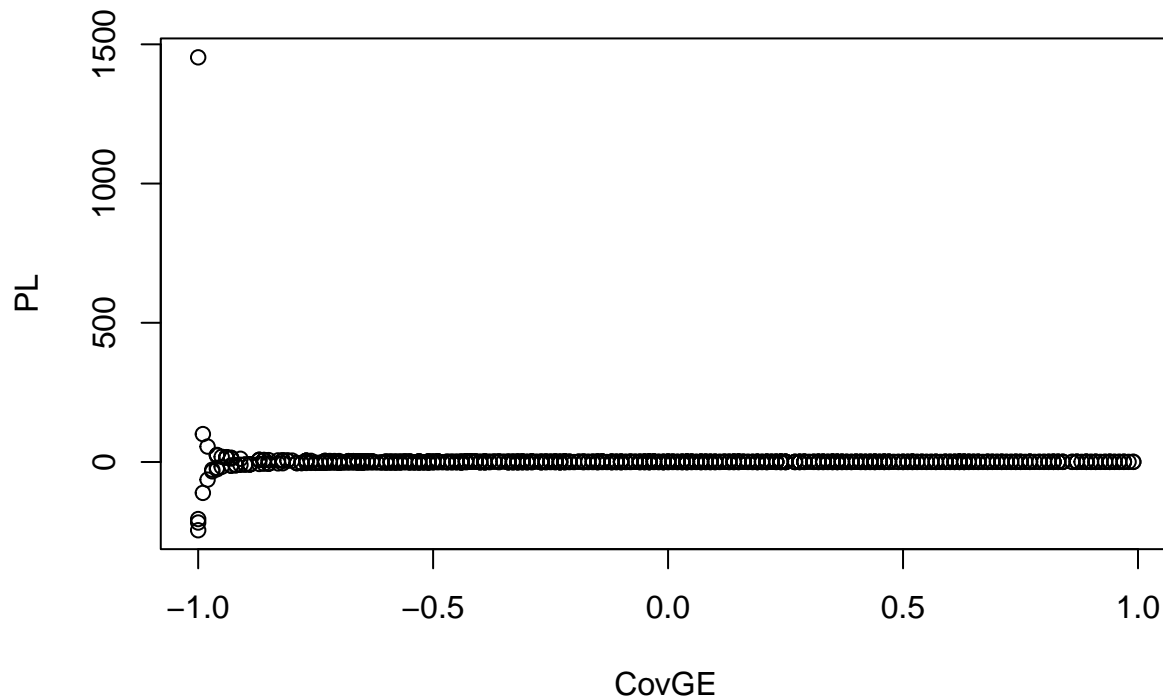
Here is an example of how to extract the mean phenotype value for 2 genotypes grown in 2 environments:

```
i = 1001
cond <- which(phendf$row==i)
tapply(phendf$phen_corrected[cond], as.character(phendf$GE_factor[cond]), mean, na.rm=TRUE)

##          G1E1          G1E2          G2E1          G2E2
## -0.7891405  0.6349688 -2.4337788  0.6584051
```

First, let's compare CovGE and PL for all the simulations:

```
## compare CovGE and PL
plot(newdf$covariance.x, newdf$PL, xlab="CovGE", ylab="PL"
)
```

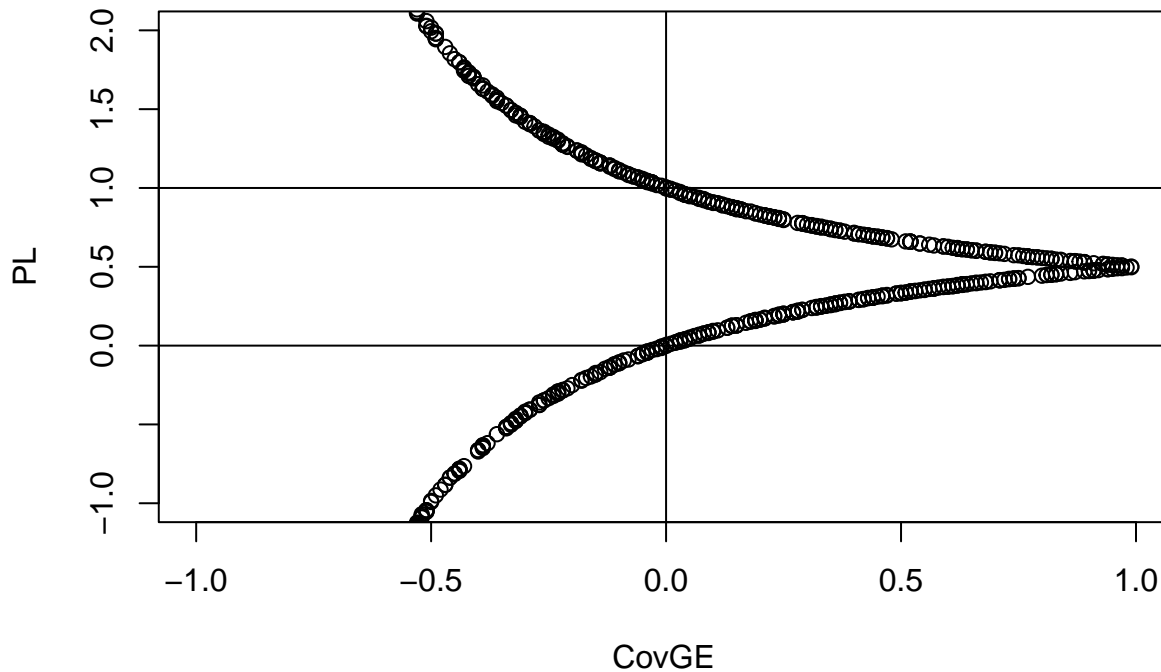



The above plot shows that as the dataset approaches countergradient variation ($\text{CovGE} = -1$), the PL metric can have very large positive or negative values.

Let's zoom in on the y-axis:

```
plot(newdf$covariance.x, newdf$PL, ylim=c(-1,2),  
     xlab="CovGE", ylab="PL", main="Figure 1")  
abline(1,0)  
abline(0,0)  
abline(v=0)
```

Figure 1



This plot shows that in 2x2 cases, the PL metric is directly related to CovGE. At PL = 0 or PL = 1, there is no gradient variation (CovGE=0). The vertical line shows where CovGE = 0 (no gradient variation). The horizontal lines show where PL = 0 and PL = 1 (perfect plasticity).

Can PL be used to infer the overall distribution of cogradients or countergradient variation in nature?

The last figure shows that PL is bimodally distributed for countergradient variation.

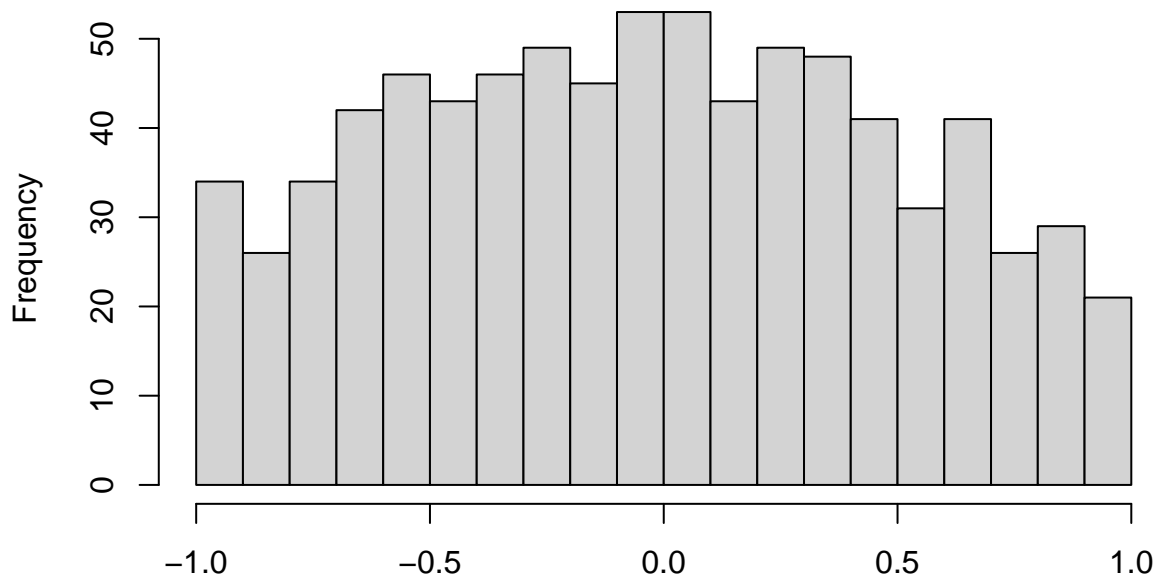
The Albecker simulations were designed to capture a range of possible patterns that may arise in 2 genotype x 2 environment experiments in nature. The distribution of simulated parameters gave an approximately uniform distribution from extreme countergradient, to no gradient variation, to extreme cogradients.

We can visualize this with the distribution of CovGE in the simulations:

```
summary(newdf$covariance.x)
```

```
##      Min.   1st Qu.   Median     Mean  3rd Qu.     Max.
## -1.00000 -0.44000 -0.03000 -0.02753  0.37000  0.99000
```

```
hist(newdf$covariance.x, main="", xlab="Distribution of CovGE in simulations", breaks=seq(-1,1,0.1))
```



Distribution of CovGE in simulations

```
sum(newdf$covariance.x>0)/length(newdf$covariance.x) # proportion of cogradient simulations
## [1] 0.4775
sum(newdf$covariance.x<0)/length(newdf$covariance.x) # proportion of countergradient simulations
## [1] 0.515
```

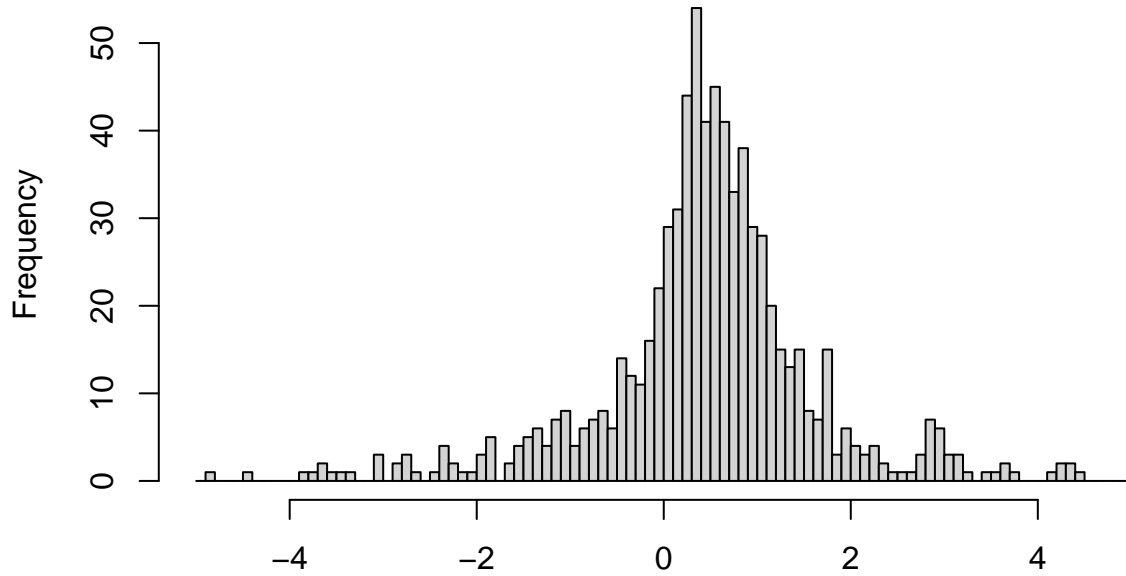
Note that the distribution is symmetric around 0, with a slight skew toward countergradient scenarios.

We can compare this to the distribution of the PL metric from the same simulations, which has a *median of 0.5* (implying that the average scenario in the simulations is perfect co-gradient variation). This happens because when $PL < 0$ (countergradient variation) it is bimodally distributed, and these values essentially cancel each other out. The consequence is that even when the average pattern in nature is no gradient variation (as it was in the simulations), the PL metric would erroneously infer cogradient variation as having the highest probability of occurring.

```
summary(newdf$PL)
```

##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
##	-245.0634	-0.0121	0.4889	1.4148	1.0360	1453.2121

```
hist(newdf$PL[newdf$PL<5 & newdf$PL>-5], main="", xlab="Distribution of PL in simulations", breaks=seq(
```



Distribution of PL in simulations

The approach of SH2020 was to look at the posterior means for the the *median* value of PL (second sentences of *Results*), which, as shown above, might be problematic for inferring the prevalence of cogradient vs. countergradient variation.

Note that if we look at the probability density of PL between 0 and 1, it gives an accurate estimate of the proportion of cogradient simulations:

```
sum(newdf$PL>0 & newdf$PL<1)/length(newdf$PL)
```

```
## [1] 0.48125
```

Summary of PL vs CovGE

Context	PL	CovGE
sampling design	2x2 reciprocal transplant only	some common garden designs and any size reciprocal transplant
Median when the distribution of gradient variation is uniform	0.5 (wrongly implying cogradient)	0 (correctly implying equal number of counter and co-gradient datasets)
Shape of distribution under countergradient variation	Bimodal (can create extremely opposite values for similar reaction norms)	Unimodal (creates similar values for similar reaction norms)
Gives insight into hyperplasticity and wrong-sign plasticity	Yes	No