

R Scripts used in "Performance trade-offs and Individual Quality in NCAA decathletes"

Jeffrey A. Walker*

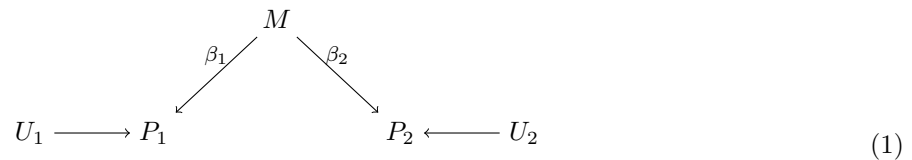
August 17, 2015

```
## Importing fonts may take a few minutes, depending on the number of fonts and the speed of the system.  
## Continue? [y/n]
```

1 Path Models of Functional Trade-offs masked by Quality

Path diagrams are models of causal relationships. Our path models are hypothetical and used only for clarification and proof of concept. They should be thought of as models for generating simulated data (which the scripts do) in order to demonstrate concepts and not as a model to fit to empirical data.

1.1 Simple model of correlation with single common cause

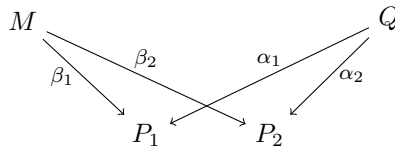


The following script explores the very simple model above of a correlation between two performance traits P_1 and P_2 due to a common cause M with effect β_1 and β_2 . U is unexplained variance, scaled so that P_1 and P_2 have unit variance, so the effects are variance-standardized. U is present in all models but is only explicitly illustrated in this first model. The expected correlation between P_1 and P_2 is $\beta_1\beta_2$.

```
n <- 10^4 # n is the sample size  
M <- rnorm(n) # M is normally distributed with mean zero and sd = 1  
beta_1 <- 0.6 # beta_1 is the path coefficient from M to P_1  
beta_2 <- -0.6 # beta_2 is the path coefficient from M to P_2  
# P are normally distributed, centered at 0, and have unit sd  
P_1 <- beta_1*M +  
  sqrt(1-beta_1^2)*rnorm(n) # the random part U  
P_2 <- beta_2*M +  
  sqrt(1-beta_1^2)*rnorm(n) # the random part U  
beta_1 * beta_2 # expected functional correlation  
  
## [1] -0.36  
  
cor(P_1,P_2) # sample correlation  
  
## [1] -0.3578166
```

*Department of Biological Sciences, University of Southern Maine, 70 Falmouth St., Portland, ME, 04103, U.S.A., E-mail: walker@maine.edu

1.2 Model of masking effect of Quality



(2)

The following script is the same as above but adds the "quality" variable Q , which has large, positive effects on both performance variables. As a consequence Q masks the intrinsic trade-off due to M

```

n <- 10^5 # n is the sample size
M <- rnorm(n) # M is normally distributed with mean zero and sd = 1
Q <- rnorm(n) # Q is normally distributed with mean zero and sd = 1 and independent of M
beta_1 <- 0.5 # beta_1 is the path coefficient from M to P_1
beta_2 <- -0.5 # beta_2 is the path coefficient from M to P_2
alpha_1 <- 0.7 # alpha_1 is the path coefficient from Q to P_1
alpha_2 <- 0.7 # alpha_2 is the path coefficient from Q to P_2
# P_1 and P_2 are normally distributed, centered at 0, and has unit sd
P_1 <- alpha_1*Q + beta_1*M + # the causal component
  sqrt(1-alpha_1^2 - beta_1^2)*rnorm(n) # the random component
P_2 <- alpha_2*Q + beta_2*M + # the causal component
  sqrt(1-alpha_2^2 - beta_1^2)*rnorm(n) # the random component

# results
alpha_1*alpha_2 + beta_1*beta_2 # expected performance correlation

## [1] 0.24

cor(P_1,P_2) # sample correlation

## [1] 0.240771

beta_1 * beta_2 # expected functional correlation

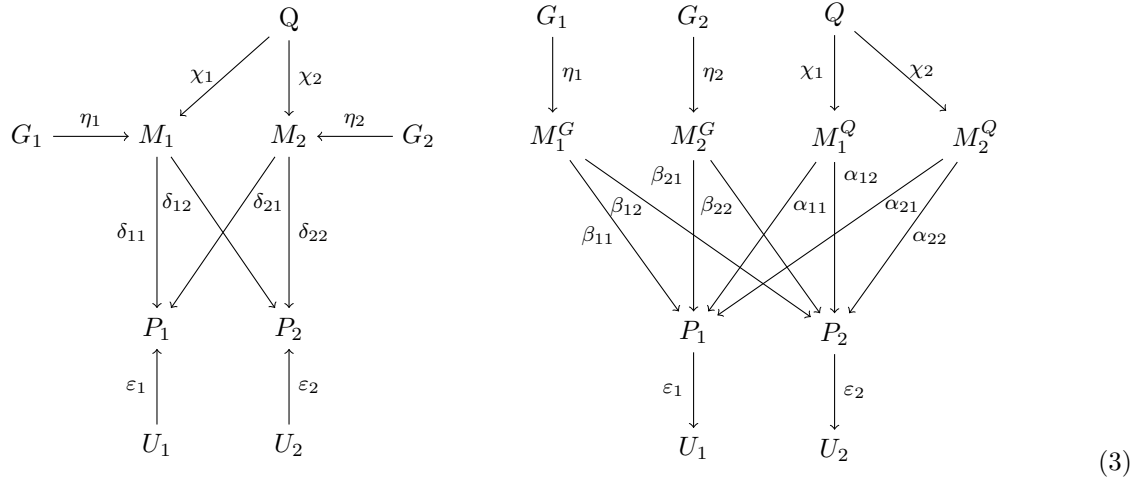
## [1] -0.25

cor(P_1,P_2) - alpha_1 * alpha_2 # sample functional correlation

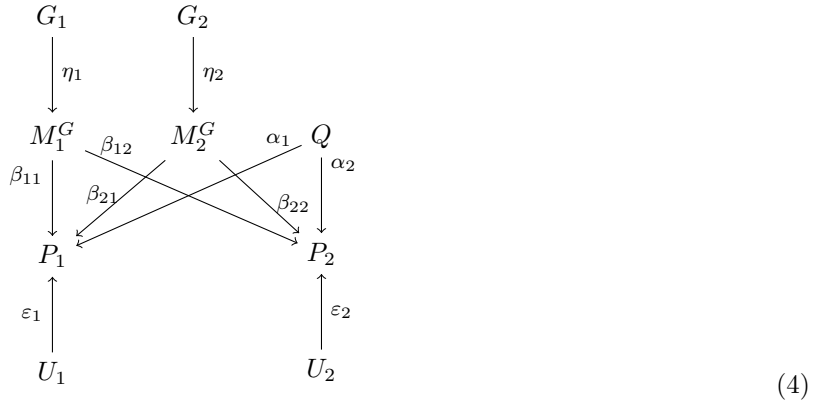
## [1] -0.249229

```

1.3 Why we model Q independent of M



The left side path model above shows that the variance of the M-P traits (M) is due to an extrinsic quality component (Q) and an intrinsic component (G), which includes genetic and developmental factors. Variation in the response to the same training is in G while variation in training is in Q . The right side path model shows the decomposition of the M into components due to Q and G , so that the path coefficients δ from the left side model decomposed into α and β in the right side model. In our text, and in the model below, we further simplify the right side by removing M_1^Q and M_2^Q as mediators, so that Q directly links to P_1 and P_2 with effects α_1 and α_2 . We show in the script below that this simplified model is precisely equivalent to both of the more complex models above.



```
n <- 10^4
Q <- rnorm(n) # Q is extrinsic quality
G_1 <- rnorm(n) # G are the intrinsic factors contributing to M
G_2 <- rnorm(n)
# standard deviations of M due to Q and G are...
chi_1 <- .1 # sd due to Q = unstandardized path coefficient from Q to M_1
chi_2 <- .7 # sd due to Q = unstandardized path coefficient from Q to M_2
eta_1 <- .6 # sd due to G = unstandardized path coefficient from G to M_1
eta_2 <- .2 # sd due to G = unstandardized path coefficient from G to M_2
M_1 <- chi_1*Q + eta_1*G_1 # centered but not standardized
M_2 <- chi_2*Q + eta_2*G_2 # centered but not standardized
# check the empirical and expected sd. Also need expected sd for later calcs
sd_m1 <- sqrt(chi_1^2 + eta_1^2)
sd_m1
```

```

## [1] 0.6082763

sd(M_1)

## [1] 0.6037943

sd_m2 <- sqrt(chi_2^2 + eta_2^2)
sd_m2

## [1] 0.728011

sd(M_2)

## [1] 0.7257984

#expected covar between M_1 and M_2
chi_1*chi_2

## [1] 0.07

cov(M_1,M_2)

## [1] 0.0705446

#expected and empirical correlation between M_1 and M_2
chi_1*chi_2/(sd_m1*sd_m2)

## [1] 0.1580736

cor(M_1,M_2)

## [1] 0.1609751

delta_11 <- -.4 # unstandardized path coefficient from M_1 to P_1
delta_12 <- .3 # unstandardized path coefficient from M_1 to P_2
delta_21 <- .5 # unstandardized path coefficient from M_2 to P_1
delta_22 <- .6 # unstandardized path coefficient from M_2 to P_2
epsilon_1 <- .3 # sd (or path coefficient) of unexplained error of P_1
epsilon_2 <- .4 # sd (or path coefficient) of unexplained error of P_1
U_1 <- rnorm(n)
U_2 <- rnorm(n)
P_1 <- delta_11*M_1 + delta_21*M_2 + epsilon_1*U_1
P_2 <- delta_12*M_1 + delta_22*M_2 + epsilon_2*U_2
# expected and empirical sd of P_1 and P_2
sd_p1 <- sqrt(delta_11^2*sd_m1^2 + delta_21^2*sd_m2^2 +
  2*delta_11*delta_21*chi_1*chi_2 + epsilon_1^2)
sd_p1

## [1] 0.5036864

sd(P_1)

## [1] 0.5018429

sd_p2<- sqrt(delta_12^2*sd_m1^2 + delta_22^2*sd_m2^2 +
  2*delta_12*delta_22*chi_1*chi_2 + epsilon_2^2)
sd_p2

```

```

## [1] 0.6397656

sd(P_2)

## [1] 0.6393697

# decomposition of the variances, just checking
sd_p1.Q <- sqrt(delta_11^2*chi_1^2 + delta_21^2*chi_2^2 +
  2*delta_11*delta_21*chi_1*chi_2)
sd_p1.G <- sqrt(delta_11^2*eta_1^2 + delta_21^2*eta_2^2)
sd_p1.U <- epsilon_1

sd_p2.Q <- sqrt(delta_12^2*chi_1^2 + delta_22^2*chi_2^2 +
  2*delta_12*delta_22*chi_1*chi_2)
sd_p2.G <- sqrt(delta_12^2*eta_1^2 + delta_21^2*eta_2^2)
sd_p2.U <- epsilon_2

sd(P_1)

## [1] 0.5018429

sqrt(sd_p1.Q^2 + sd_p1.G^2 + sd_p1.U^2)

## [1] 0.5036864

sd_p1

## [1] 0.5036864

sd(P_2)

## [1] 0.6393697

sqrt(sd_p2.Q^2 + sd_p2.G^2 + sd_p2.U^2)

## [1] 0.6363175

sd_p2

## [1] 0.6397656

# full correlation path model
delta_s_11 <- delta_11*sd_m1/sd_p1 # standardized
delta_s_12 <- delta_12*sd_m1/sd_p2 # standardized
delta_s_21 <- delta_21*sd_m2/sd_p1 # standardized
delta_s_22 <- delta_22*sd_m2/sd_p2 # standardized
chi_s_1 <- chi_1/sd_m1 # standardized
chi_s_2 <- chi_2/sd_m2 # standardized

# empirical correlations
cor(P_1,P_2)

## [1] 0.3344963

# expected correlation
delta_s_11*delta_s_12 + delta_s_21*delta_s_22 +
  delta_s_11*chi_s_1*chi_s_2*delta_s_22 + delta_s_12*chi_s_1*chi_s_2*delta_s_21

```

```
## [1] 0.3360836

# decomposition into intrinsic and extrinsic components
# beta is the component due to G
# alpha is the component due to Q
beta_s_11 <- delta_11*eta_1/sd_p1 # standardized
beta_s_12 <- delta_12*eta_1/sd_p2 # standardized
beta_s_21 <- delta_21*eta_2/sd_p1 # standardized
beta_s_22 <- delta_22*eta_2/sd_p2 # standardized

alpha_s_11 <- delta_11*chi_1/sd_p1 # standardized
alpha_s_12 <- delta_12*chi_1/sd_p2 # standardized
alpha_s_21 <- delta_21*chi_2/sd_p1 # standardized
alpha_s_22 <- delta_22*chi_2/sd_p2 # standardized

# check to see if beta and alpha components sum equal to delta_s_11
sqrt(beta_s_11^2 + alpha_s_11^2)

## [1] 0.4830595

delta_s_11

## [1] -0.4830595

# expected correlations
r_beta <- beta_s_11*beta_s_12 + beta_s_21*beta_s_22 # due to beta
# checking the math, should be equivalent
r_beta <- delta_11*delta_12*eta_1^2/(sd_p1*sd_p2) +
  delta_21*delta_22*eta_2^2/(sd_p1*sd_p2)
# the 3rd and 4th parts use delta instead of alpha because the correlation
# between M_1 and M_2 requires all of the variance and not just that due to Q
r_alpha <- alpha_s_11*alpha_s_12 + alpha_s_21*alpha_s_22 +
  delta_s_11*chi_s_1*chi_s_2*delta_s_22 +
  delta_s_12*chi_s_1*chi_s_2*delta_s_21
r_alpha

## [1] 0.4329055

# computing r_alpha without the standardized coefficients shows that sd_m1 and
# sd_m2 are in both num and denom and cancel
# = r_alpha
delta_11*delta_12*chi_1^2/(sd_p1*sd_p2) +
  delta_21*delta_22*chi_2^2/(sd_p1*sd_p2) +
  delta_11*sd_m1/sd_p1 * chi_1/sd_m1 * chi_2/sd_m2 * delta_22*sd_m2/sd_p2 +
  delta_12*sd_m1/sd_p2 * chi_1/sd_m1 * chi_2/sd_m2 * delta_21*sd_m2/sd_p1

## [1] 0.4329055

# simplifying by removing M and directly linking Q to P_1 and P_2 with
# coefficients alpha_1 and alpha_2 as sum of products along paths
alpha_1 <- chi_1*delta_11/sd_p1 + chi_2*delta_21/sd_p1
alpha_2 <- chi_1*delta_12/sd_p2 + chi_2*delta_22/sd_p2
alpha_1*alpha_2 # = r_alpha

## [1] 0.4329055
```

```

# expected correlation not decomposed (left side)
delta_s_11*delta_s_12 + delta_s_21*delta_s_22 +
  delta_s_11*chi_s_1*chi_s_2*delta_s_22 + delta_s_12*chi_s_1*chi_s_2*delta_s_21

## [1] 0.3360836

#expected correlation decomposed (right side) - should be equivalent to above
r_beta + r_alpha

## [1] 0.3360836

#empirical correlation
cor(P_1,P_2)

## [1] 0.3344963

#expected functional correlation
r_beta

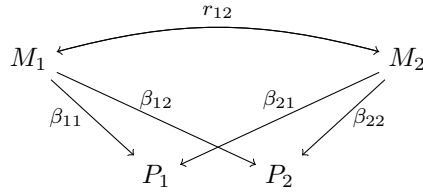
## [1] -0.09682187

# empirical functional correlation
cor(P_1,P_2) - r_alpha

## [1] -0.09840923

```

1.4 Effect of inter-trait correlation on functional correlation



(5)

Demonstration that the correlation between M_1 and M_2 can reverse the sign of the functional correlation. The correlation between M_1 and M_2 is set by the effects (γ_1 and γ_2 of their common cause Z on each.

```

n <- 10^5 # n is the sample size
gamma_1 <- .9 # this is the effect of the common cause Z on M_1
gamma_2 <- -.9 # this is the effect of the common cause Z on M_2
r_12 <- gamma_1*gamma_2 # the expected correlation between M_1 and M_2
Z <- rnorm(n) # M is normally distributed with mean zero and sd = 1
# M_1 and M_2 are normally distributed with mean zero and sd = 1
M_1 <- gamma_1*Z + sqrt(1-gamma_1^2)*rnorm(n)
M_2 <- gamma_2*Z + sqrt(1-gamma_2^2)*rnorm(n)
beta_11 <- 0.6 # beta_1 is the path coefficient from M to P_1
beta_12 <- -0.1 # beta_2 is the path coefficient from M to P_2
beta_21 <- 0.1 # alpha_1 is the path coefficient from Q to P_1
beta_22 <- -0.6 # alpha_2 is the path coefficient from Q to P_2
# P_1 and P_2 are normally distributed, centered at 0, and has unit sd.
P_1 <- beta_11*M_1 + beta_21*M_2 +
  sqrt(1-beta_11^2 - beta_21^2 - 2*beta_11*beta_21*r_12)*rnorm(n)
P_2 <- beta_12*M_1 + beta_22*M_2 +
  sqrt(1-beta_12^2 - beta_22^2 - 2*beta_12*beta_22*r_12)*rnorm(n)

```

```

#results
apply(data.frame(M_1,M_2,P_1,P_2),2,var) # show variances ~ 1

##      M_1      M_2      P_1      P_2
## 1.006254 1.006203 1.007059 1.002162

r_12 # the expected correlation between M_1 and M_2

## [1] -0.81

cor(M_1,M_2) # the sample correlation between M_1 and M_2

## [1] -0.8104344

# the expected performance correlation
beta_11*beta_12 + beta_21*beta_22 + r_12*beta_11*beta_22 + r_12*beta_12*beta_21

## [1] 0.1797

cor(P_1,P_2) # sample performance correlation

## [1] 0.1787275

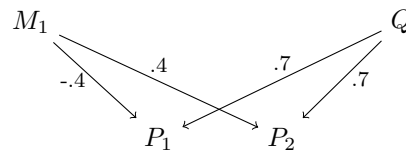
beta_11*beta_12 + beta_21*beta_22 # the functional correlation component

## [1] -0.12

```

1.5 How quality or a facilitation does not mask trade-offs at the whole animal level

This is a script to explore the section "What facilitations or quality do not mask" in the Discussion of the paper.



(6)

```

seed <- 10566
set.seed(seed)
M <- rnorm(n)
Q <- rnorm(n)
beta_1 <- -.4
beta_2 <- .4
alpha_1 <- .7
alpha_2 <- .7
n <- 10^4
E_1 <- rnorm(n)
E_2 <- rnorm(n)
P_1 <- alpha_1*Q + beta_1*M + sqrt(1 - alpha_1^2 - beta_1^2)*E_1
P_2 <- alpha_2*Q + beta_2*M + sqrt(1 - alpha_2^2 - beta_2^2)*E_2
alpha_1*alpha_2 + beta_1*beta_2 # expected r

## [1] 0.33

```



```

cor(P_1,P_2) # actual r
## [1] 0.3286328

# add training. Two notes
# 1) The response of the M trait to training is precisely the same. There will obviously
# be variation in real humans
# 2) The Error in the performance is precisely that before training (that is the error
# has correlation = 1), which means that an athlete that was .37 units above the mean in
# P_1 prior to training was also .37 units above the mean post training. In reality this
# error will likely be correlated (even high correlated) but not 1. Even if the error
# post training is independent of the error pre-training, the expected mean response is
# delta_M*beta
delta_M <- .5
M_trained <- M + delta_M
E1_post <- E_1 # make this rnorm(n) to make it independent of E_1
E2_post <- E_2
P1_trained <- alpha_1*Q + beta_1*M_trained + sqrt(1 - alpha_1^2 - beta_1^2)*E1_post
P2_trained <- alpha_2*Q + beta_2*M_trained + sqrt(1 - alpha_2^2 - beta_2^2)*E2_post
delta_P1 <- (P1_trained - P_1)
delta_P2 <- (P2_trained - P_2)
beta_1*delta_M # expected mean response in P1
## [1] -0.2

beta_2*delta_M #expected mean response in P2
## [1] 0.2

mean(delta_P1) # sample mean response to training for P1
## [1] -0.2

mean(delta_P2) # sample mean response to training for P2
## [1] 0.2

data.frame(delta_P1=delta_P1,delta_P2=delta_P2)[1:5,]

##   delta_P1 delta_P2
## 1    -0.2     0.2
## 2    -0.2     0.2
## 3    -0.2     0.2
## 4    -0.2     0.2
## 5    -0.2     0.2

## pdf
## 2

```