



Fig. S1. Results from a linearity assessment of a testosterone enzyme-linked immunosorbent assay (ELISA) using pooled olive ridley sea turtle (*Lepidochelys olivacea*) plasma extracts (two adult males, one adult female, and one immature male). Serial dilutions of a pool of extracted olive ridley plasma samples (hollow circles) show parallelism with serial dilutions of testosterone standards (filled squares). %B/B0 = %bound/maximum bound

R and JAGS code for the Bayesian Model used to predict olive ridley sex. Note: transfer the code to a text file to run it.

R code:

```
compute.LOOIC <- function(loglik, data.vector, MCMC.params){
  n.per.chain <- (MCMC.params$n.samples - MCMC.params$n.burnin)/MCMC.params$n.thin

  loglik.vec <- as.vector(loglik)

  # each column corresponds to a data point and rows are MCMC samples
  loglik.mat <- matrix(loglik.vec, nrow = n.per.chain * MCMC.params$n.chains)

  # take out the columns that correspond to missing data points
  loglik.mat <- loglik.mat[, !is.na(data.vector)]
  # loglik.mat <- matrix(loglik.vec[!is.na(data.vector)],
  #                       nrow = MCMC.params$n.chains * n.per.chain)

  Reff <- relative_eff(exp(loglik.mat),
                      chain_id = rep(1:MCMC.params$n.chains,
                                     each = n.per.chain),
                      cores = 4)

  #
  loo.out <- loo(loglik.mat,
                r_eff = Reff,
                cores = 4, k_threshold = 0.7)

  out.list <- list(Reff = Reff,
                  loo.out = loo.out)

  return(out.list)
}

models <- c("sex_only",
            "sex_tail", "sex_tail", "sex_temp",
            "sex_tail_SCL", "sex_tail_SCL", "sex_tail_temp", "sex_tail_temp",
            "sex_tail_SCL_temp", "sex_tail_SCL_temp",
            "sex_tail", "sex_tail",
            "sex_tail_SCL", "sex_tail_SCL", "sex_tail_temp", "sex_tail_temp",
            "sex_tail_SCL_temp", "sex_tail_SCL_temp",
            "sex_SCL_temp")

model.names <- c("sex_only",
                "sex_tail", "sex_tail2", "sex_temp",
                "sex_tail_SCL", "sex_tail2_SCL", "sex_tail_temp", "sex_tail2_temp",
                "sex_tail_SCL_temp", "sex_tail2_SCL_temp",
                "sex_ptail", "sex_ptail2",
                "sex_ptail_SCL", "sex_ptail2_SCL",
                "sex_ptail_temp", "sex_ptail2_temp",
                "sex_ptail_SCL_temp", "sex_ptail2_SCL_temp",
                "sex_SCL_temp")

# use mean for missing SST
dat.2.3$TempC[is.na(dat.2.3$TempC)] <- mean(dat.2.3$TempC, na.rm = T)

jags.data.list <- list(list(H1 = dat.2.3[, c("log_Testo1_mean0",      #sex_only
                                           "log_Testo2_mean0")],
                          Sex1 = dat.2.3$sex01,
                          N1 = nrow(dat.2.3)),

                      list(H1 = dat.2.3[, c("log_Testo1_mean0",      #sex_tail
                                           "log_Testo2_mean0")],
```

```
d      Sex1 = dat.2.3$sex01,
      N1 = nrow(dat.2.3),
      tail = dat.2.3$tail_length),

list(H1 = dat.2.3[, c("log_Testo1_mean0",      # sex_tail2
                    "log_Testo2_mean0")],
     Sex1 = dat.2.3$sex01,
     N1 = nrow(dat.2.3),
     tail = dat.2.3$tail_2),

list(H1 = dat.2.3[, c("log_Testo1_mean0",      # sex_temp
                    "log_Testo2_mean0")],
     Sex1 = dat.2.3$sex01,
     N1 = nrow(dat.2.3),
     temp = dat.2.3$TempC),

list(H1 = dat.2.3[, c("log_Testo1_mean0",      # sex_tail_SCL
                    "log_Testo2_mean0")],
     Sex1 = dat.2.3$sex01,
     N1 = nrow(dat.2.3),
     tail = dat.2.3$tail_length,
     SCL = dat.2.3$SCL),

list(H1 = dat.2.3[, c("log_Testo1_mean0",      #sex_tail2_SCL
                    "log_Testo2_mean0")],
     Sex1 = dat.2.3$sex01,
     N1 = nrow(dat.2.3),
     tail = dat.2.3$tail_2,
     SCL = dat.2.3$SCL),

list(H1 = dat.2.3[, c("log_Testo1_mean0",      #sex_tail_temp
                    "log_Testo2_mean0")],
     Sex1 = dat.2.3$sex01,
     N1 = nrow(dat.2.3),
     tail = dat.2.3$tail_length,
     temp = dat.2.3$TempC),

list(H1 = dat.2.3[, c("log_Testo1_mean0",      #sex_tail2_temp
                    "log_Testo2_mean0")],
     Sex1 = dat.2.3$sex01,
     N1 = nrow(dat.2.3),
     tail = dat.2.3$tail_2,
     temp = dat.2.3$TempC),

list(H1 = dat.2.3[, c("log_Testo1_mean0",      #sex_tail_SCL_temp
                    "log_Testo2_mean0")],
     Sex1 = dat.2.3$sex01,
     N1 = nrow(dat.2.3),
     tail = dat.2.3$tail_length,
     temp = dat.2.3$TempC,
     SCL = dat.2.3$SCL),

list(H1 = dat.2.3[, c("log_Testo1_mean0",      #sex_tail2_SCL_temp
                    "log_Testo2_mean0")],
     Sex1 = dat.2.3$sex01,
     N1 = nrow(dat.2.3),
     tail = dat.2.3$tail_2,
     temp = dat.2.3$TempC,
     SCL = dat.2.3$SCL),

list(H1 = dat.2.3[, c("log_Testo1_mean0",      #sex_ptail
                    "log_Testo2_mean0")],
     Sex1 = dat.2.3$sex01,
```

```
      N1 = nrow(dat.2.3),
      tail = dat.2.3$tail_length/dat.2.3$SCL),

list(H1 = dat.2.3[, c("log_Testo1_mean0", #sex_ptail2
                     "log_Testo2_mean0")],
     Sex1 = dat.2.3$sex01,
     N1 = nrow(dat.2.3),
     tail = dat.2.3$tail_2/dat.2.3$SCL),

list(H1 = dat.2.3[, c("log_Testo1_mean0", # sex_ptail_SCL
                     "log_Testo2_mean0")],
     Sex1 = dat.2.3$sex01,
     N1 = nrow(dat.2.3),
     tail = dat.2.3$tail_length/dat.2.3$SCL,
     SCL = dat.2.3$SCL),

list(H1 = dat.2.3[, c("log_Testo1_mean0", # sex_ptail2_SCL
                     "log_Testo2_mean0")],
     Sex1 = dat.2.3$sex01,
     N1 = nrow(dat.2.3),
     tail = dat.2.3$tail_2/dat.2.3$SCL,
     SCL = dat.2.3$SCL),

list(H1 = dat.2.3[, c("log_Testo1_mean0", #sex_ptail_temp
                     "log_Testo2_mean0")],
     Sex1 = dat.2.3$sex01,
     N1 = nrow(dat.2.3),
     tail = dat.2.3$tail_length/dat.2.3$SCL,
     temp = dat.2.3$TempC),

list(H1 = dat.2.3[, c("log_Testo1_mean0", #sex_ptail2_temp
                     "log_Testo2_mean0")],
     Sex1 = dat.2.3$sex01,
     N1 = nrow(dat.2.3),
     tail = dat.2.3$tail_2/dat.2.3$SCL,
     temp = dat.2.3$TempC),

list(H1 = dat.2.3[, c("log_Testo1_mean0", #sex_ptail_SCL_temp
                     "log_Testo2_mean0")],
     Sex1 = dat.2.3$sex01,
     N1 = nrow(dat.2.3),
     tail = dat.2.3$tail_length/dat.2.3$SCL,
     temp = dat.2.3$TempC,
     SCL = dat.2.3$SCL),

list(H1 = dat.2.3[, c("log_Testo1_mean0", #sex_ptail2_SCL_temp
                     "log_Testo2_mean0")],
     Sex1 = dat.2.3$sex01,
     N1 = nrow(dat.2.3),
     tail = dat.2.3$tail_2/dat.2.3$SCL,
     temp = dat.2.3$TempC,
     SCL = dat.2.3$SCL),

list(H1 = dat.2.3[, c("log_Testo1_mean0", #sex_SCL_temp
                     "log_Testo2_mean0")],
     Sex1 = dat.2.3$sex01,
     N1 = nrow(dat.2.3),
     temp = dat.2.3$TempC,
     SCL = dat.2.3$SCL)

parameters <- c("Sex1", "tail",
               "beta_0", "beta_Sex", "beta_T", "beta_L", "beta_temp",
               "mu_H1", "sigma_H1",
```

```

        "deviance", "loglik")

LOOIC <- DIC <- vector(mode = "numeric", length = length(models))

for (k in 1:length(models)){

  jags.data <- jags.data.list[[k]]
  MCMC.params$parameters <- parameters
  MCMC.params$model.file <- paste0("models/Model_", models[[k]], ".txt")
  start.time <- Sys.time()
  jm.sex <- jags(data = jags.data,
                #inits = inits,
                parameters.to.save= MCMC.params$parameters,
                model.file = MCMC.params$model.file,
                n.chains = MCMC.params$n.chains,
                n.burnin = MCMC.params$n.burnin,
                n.thin = MCMC.params$n.thin,
                n.iter = MCMC.params$n.samples,
                DIC = T,
                parallel=T)
  elapsed.time <- Sys.time() - start.time
  jm.out <- list(data = jags.data,
                params = MCMC.params,
                jm = jm.sex,
                Run.Date = Sys.Date(),
                comp.time = elapsed.time,
                System = Sys.getenv())

  saveRDS(jm.out, file = paste0("RData/LO_", model.names[[k]],
                                "_May2024.rds"))

  DIC[k] <- jm.out$jm$DIC
  loo.out <- compute.LOOIC(loglik = jm.out$jm$sims.list$loglik,
                          data.vector = as.vector(jm.out$data$H1),
                          MCMC.params = jm.out$params)

  saveRDS(loo.out, paste0("RData/LO_", model.names[[k]],
                          "_loo_out_May2024.rds") )
  LOOIC[k] <- loo.out$loo.out$estimates["looic","Estimate"]
}

max.pareto <- n.bad.pareto <- p.bad.pareto <- vector(mode = "numeric", length =
length(model.names))
for (k in 1:length(model.names)){
  loo.out <- readRDS(paste0("RData/LO_", model.names[k], "_loo_out_May2024.rds"))
  max.pareto[k] <- max(loo.out$loo.out$diagnostics$pareto_k)
  n.bad.pareto[k] <- sum(loo.out$loo.out$diagnostics$pareto_k > 0.7)
  p.bad.pareto[k] <- sum(loo.out$loo.out$diagnostics$pareto_k >
0.7)/length(loo.out$loo.out$diagnostics$pareto_k)
}

LOOIC_DIC <- data.frame(Model = model.names,
                        dDIC = signif(DIC - min(DIC), 5),
                        dLOOIC = signif(LOOIC - min(LOOIC), 3),
                        max.pareto = signif(max.pareto, 2),
                        bad.pareto = n.bad.pareto,
                        percent.bad.pareto = signif(p.bad.pareto*100,2)) %>% arrange(by = dLOOIC)

```

JAGS code:

Sex only:

```

model {
  for (i in 1:N1){

```

```

for (j in 1:2){
  # variability between two samples are about 0.01 (= variance)
  # so even precision = 10 (= var = 0.1) is an order of magnitude
  # greater than the observed - should be "flat" enough - see sigma_H1
  # below.

  H1[i, j] ~ dnorm(mu_H1[i], tau_H1)
  loglik[i, j] <- logdensity.norm(H1[i,j], mu_H1[i], tau_H1)

}
mu_H1[i] <- beta_0 + beta_Sex * Sex1[i]
}

# because we don't have all sex identifications
for (i in 1:N1){
  Sex1[i] ~ dbern(q)
}

q ~ dbeta(1,1)

# Setting SD of observations to be somewhere between 0 to 0.4, which equates
# to var = 0 - 0.16, which is an order of magnitude greater than the observed
# variance ~ 0.01
sigma_H1 ~ dunif(0, 0.4) #1/sqrt(tau_H1)

tau_H1 <- 1/(sigma_H1^2)

beta_0 ~ dnorm(0, 0.1)
beta_Sex ~ dnorm(0, 0.1)
}

```

Sex and tail:

```

model {
  for (i in 1:N1){
    for (j in 1:2){
      # variability between two samples are about 0.01 (= variance)
      # so even precision = 10 (= var = 0.1) is an order of magnitude
      # greater than the observed - should be "flat" enough - see sigma_H1
      # below.

      H1[i, j] ~ dnorm(mu_H1[i], tau_H1)
      loglik[i, j] <- logdensity.norm(H1[i,j], mu_H1[i], tau_H1)

    }
    mu_H1[i] <- beta_0 + beta_Sex * Sex1[i] + beta_T * tail[i]
  }

  # because we don't have all sex identifications
  for (i in 1:N1){
    Sex1[i] ~ dbern(q)
  }

  # Some tail measurements are missing
  for (i in 1:N1){
    tail[i] ~ dnorm(15, 0.01)I(0,)
  }

  q ~ dbeta(1,1)

  # Setting SD of observations to be somewhere between 0 to 0.4, which equates

```

```
# to var = 0 - 0.16, which is an order of magnitude greater than the observed
# variance ~ 0.01
sigma_H1 ~ dunif(0, 0.4) #1/sqrt(tau_H1)

tau_H1 <- 1/(sigma_H1^2)
beta_0 ~ dnorm(0, 0.1)
beta_Sex ~ dnorm(0, 0.1)
beta_T ~ dnorm(0, 0.1)
}
```

Sex and temp

```
model {
  for (i in 1:N1){
    for (j in 1:2){
      # variability between two samples are about 0.01 (= variance)
      # so even precision = 10 (= var = 0.1) is an order of magnitude
      # greater than the observed - should be "flat" enough - see sigma_H1
      # below.

      H1[i, j] ~ dnorm(mu_H1[i], tau_H1)
      loglik[i, j] <- logdensity.norm(H1[i,j], mu_H1[i], tau_H1)

    }
    mu_H1[i] <- beta_0 + beta_Sex * Sex1[i] + beta_temp * temp[i]
  }

  # because we don't have all sex identifications
  for (i in 1:N1){
    Sex1[i] ~ dbern(q)
  }

  q ~ dbeta(1,1)

  # Setting SD of observations to be somewhere between 0 to 0,4, which equates
  # to var = 0 - 0.16, which is an order of magnitude greater than the observed
  # variance ~ 0.01
  sigma_H1 ~ dunif(0, 0.4) #1/sqrt(tau_H1)

  tau_H1 <- 1/(sigma_H1^2)
  beta_0 ~ dnorm(0, 0.1)
  beta_Sex ~ dnorm(0, 0.1)
  beta_temp ~ dnorm(0, 0.1)
}
```

Sex, tail, and SCL:

```
model {
  for (i in 1:N1){
    for (j in 1:2){
      # variability between two samples are about 0.01 (= variance)
      # so even precision = 10 (= var = 0.1) is an order of magnitude
      # greater than the observed - should be "flat" enough - see sigma_H1
      # below.

      H1[i, j] ~ dnorm(mu_H1[i], tau_H1)
      loglik[i, j] <- logdensity.norm(H1[i,j], mu_H1[i], tau_H1)
    }

    mu_H1[i] <- beta_0 + beta_Sex * Sex1[i] + beta_T * tail[i] + beta_L * SCL[i]
  }
}
```

```

# because we don't have all sex identifications
for (i in 1:N1){
  Sex1[i] ~ dbern(q)
}

# Some tail measurements are missing
for (i in 1:N1){
  tail[i] ~ dnorm(15, 0.01)I(0,)
}

q ~ dbeta(1,1)

# Setting SD of observations to be somewhere between 0 to 0,4, which equates
# to var = 0 - 0.16, which is an order of magnitude greater than the observed
# variance ~ 0.01
sigma_H1 ~ dunif(0, 0.4) #1/sqrt(tau_H1)

tau_H1 <- 1/(sigma_H1^2)

beta_0 ~ dnorm(0, 0.1)
beta_Sex ~ dnorm(0, 0.1)
beta_T ~ dnorm(0, 0.1)
beta_L ~ dnorm(0, 0.1)
}

```

Sex, tail, and temperature:

```

model {
  for (i in 1:N1){
    for (j in 1:2){
      # variability between two samples are about 0.01 (= variance)
      # so even precision = 10 (= var = 0.1) is an order of magnitude
      # greater than the observed - should be "flat" enough - see sigma_H1
      # below.

      H1[i, j] ~ dnorm(mu_H1[i], tau_H1)
      loglik[i, j] <- logdensity.norm(H1[i,j], mu_H1[i], tau_H1)
    }

    mu_H1[i] <- beta_0 + beta_Sex * Sex1[i] + beta_T * tail[i] + beta_temp * temp[i]
  }

  # because we don't have all sex identifications
  for (i in 1:N1){
    Sex1[i] ~ dbern(q)
  }

  # Some tail measurements are missing
  for (i in 1:N1){
    tail[i] ~ dnorm(15, 0.01)I(0,)
  }

  q ~ dbeta(1,1)

  # Setting SD of observations to be somewhere between 0 to 0,4, which equates
  # to var = 0 - 0.16, which is an order of magnitude greater than the observed
  # variance ~ 0.01
  sigma_H1 ~ dunif(0, 0.4) #1/sqrt(tau_H1)

  tau_H1 <- 1/(sigma_H1^2)
}

```



```

beta_0 ~ dnorm(0, 0.1)
beta_Sex ~ dnorm(0, 0.1)
beta_T ~ dnorm(0, 0.1)
beta_temp ~ dnorm(0, 0.1)
}

```

Sex, tail, SCL, and temperature

```

model {
  for (i in 1:N1){
    for (j in 1:2){
      # variability between two samples are about 0.01 (= variance)
      # so even precision = 10 (= var = 0.1) is an order of magnitude
      # greater than the observed - should be "flat" enough - see sigma_H1
      # below.

      H1[i, j] ~ dnorm(mu_H1[i], tau_H1)
      loglik[i, j] <- logdensity.norm(H1[i,j], mu_H1[i], tau_H1)
    }

    mu_H1[i] <- beta_0 + beta_Sex * Sex1[i] + beta_T * tail[i] + beta_L * SCL[i] +
beta_temp * temp[i]

  }

  # because we don't have all sex identifications
  for (i in 1:N1){
    Sex1[i] ~ dbern(q)
  }

  # Some tail measurements are missing
  for (i in 1:N1){
    tail[i] ~ dnorm(15, 0.01)I(0,)
  }

  q ~ dbeta(1,1)

  # Setting SD of observations to be somewhere between 0 to 0,4, which equates
  # to var = 0 - 0.16, which is an order of magnitude greater than the observed
  # variance ~ 0.01
  sigma_H1 ~ dunif(0, 0.4) #1/sqrt(tau_H1)

  tau_H1 <- 1/(sigma_H1^2)

  beta_0 ~ dnorm(0, 0.1)
  beta_Sex ~ dnorm(0, 0.1)
  beta_T ~ dnorm(0, 0.1)
  beta_L ~ dnorm(0, 0.1)
  beta_temp ~ dnorm(0, 0.1)
}

```