Predicting evolution using frequency-dependent selection in bacterial populations

Data analysis and simulations

Taj Azarian and Pamela P. Martinez

Contents

This is the code for reproducing the figures from the paper "Predicting evolution using frequency-dependent selection in bacterial populations" ## add DOI

```
require(ape)
require(car)
require(readr)
require(vegan)
require(gtools)
require(ggtree)
require(readxl)
require(gridSVG)
require(cowplot)
require(deSolve)
require(Metrics)
require(ggrepel)
require(svglite)
require(quadprog)
require(phytools)
require(devtools)
require(tidyverse)
require(wesanderson)
require(RColorBrewer)

#### Quadratic Programming function ####
## X is a matrix with rows = COGs and columns = SCs
## Y is a matrix with rows = COGs and columns = 1
QP <- function(X, Y){
  rinv <- solve(chol(t(X) %*% X)) # M to be minimized in quad. function (Choleski decomp)
```

```
  C <- cbind(rep(1,ncol(X)), diag(ncol(X))) #The contraints to minimize the quandratic function
  b <- c(1,rep(0,ncol(X)))
  d <- t(Y) %*% X  #Vector in the quadratic function to be minimized
  output <- solve.QP(Dmat = rinv, factorized = TRUE, dvec = d,
                     Amat = C, bvec = b, meq = 1)$solution
  output <- round(output, digits = 5)
  return(output)
}
```

Figure 1: Southwest US Data

```
#### Data mining ####
set.seed(9340)
df <- read_csv("data_southwestUS.csv") ## presence absence dataframe with metadata

dfE1 <- df %>% subset(Epoch1 == "E1") ## E1 prevaccine
vaccineT <- df %>% distinct(BAPS2, PCV7.actual)
vaccineT <- vaccineT %>% subset(PCV7.actual == "VT") %>%
  dplyr::rename(W = PCV7.actual) %>%
  full_join(subset(vaccineT,PCV7.actual == "NVT")) %>%
  unite("vaccine", W:PCV7.actual, na.rm = TRUE)

dfF <- df %>% select(BAPS2, Epoch1) %>% group_by(Epoch1) %>%
  count(BAPS2) %>% mutate(freq = prop.table(n)) %>% ungroup() %>%
  select(Epoch1, BAPS2, freq) %>%
  spread(Epoch1, freq, fill = 0)

zero_E1 <- dfE1 %>% distinct(BAPS2) %>%
  mutate(n = 0, freq = 0)
```

```
#### Replicates  - null expectation Pro rata ####
dfE1_NVT_all <- data.frame(NULL)
replicates <- 10000

for(i in 1:replicates){
  #sub-sampling from each epoch independently with replacemnt - best subsampling strategy
  dfE1_NVT_i <- dfE1 %>% sample_frac(1, replace = TRUE) %>%
    subset(PCV7.actual == "NVT") %>% count(BAPS2) %>%
    mutate(freq = prop.table(n)) %>% bind_rows(zero_E1) %>%
    group_by(BAPS2) %>% summarise(n = sum(n), freq = sum(freq)) %>%
    arrange(BAPS2) %>% ungroup %>% mutate(iter = i)

  dfE1_NVT_all <- bind_rows(dfE1_NVT_all, dfE1_NVT_i)
}

dfE1_NVT_all <- dfE1_NVT_all %>% group_by(BAPS2) %>%
  summarise(expected_E3 = quantile(freq, 0.5), cil = quantile(freq, 0.025),
            ciu = quantile(freq, 0.975)) %>% ungroup

dfF <- left_join(dfF,dfE1_NVT_all) %>% replace(., is.na(.), 0) %>%
  mutate(delta = E3 - E1, deltaExp = expected_E3-E1,
         CI_low = cil-E1, CI_up = ciu-E1) %>%
```

```
  mutate(SC=BAPS2) %>%
  mutate(expectation = "Null exp.", signif = ifelse(delta > CI_up, "pos",NA)) %>%
  mutate(signif = ifelse(delta < CI_low, "neg", signif)) %>%
  mutate(signif = ifelse(E1 == 0 | E3 == 0, NA, signif))

dfF <- left_join(dfF,vaccineT)

#### Plot A: Prevalence by sequence cluster ####
dat1 <- mutate(dfF, signif = ifelse(E3 > CI_up | E3 < CI_low, -0.005,NA), Epoch = "E1")
dat1 <- subset(dat1, select = c(SC, Epoch, signif))

datPlotA <- dfF %>% select(SC, E1, E3) %>%
  pivot_longer(-SC, names_to =  "Epoch", values_to = "Prevalence") %>%
  left_join(dat1) %>% left_join(select(dfF, SC, vaccine)) %>%
  mutate(SC = as.character(SC)) %>% arrange(Epoch,-Prevalence)

rank <- as.character(unique(datPlotA$SC))

datPlotA <- datPlotA %>% mutate(SC = factor(SC, levels = rank))

plot1A <- ggplot(datPlotA, aes(x=SC, y=Prevalence, alpha=Epoch, fill=vaccine)) +
  geom_bar(stat='identity', position='dodge') +
  scale_alpha_manual(values = c(1,0.4),
                     labels = c("Pre-vaccine", "Post-vaccine")) +
  scale_fill_manual(values = c("#143c77","darkred","mediumpurple4"),
                    labels = c("Non-vaccine type", "Vaccine type", "Mixed"),
                    name = "Composition") +
  xlab("Strain (SC)") + theme_classic() +
  scale_y_continuous(expand = c(0, 0), limits = c(0,0.15)) +
  theme(legend.title = element_text(face="bold", size = 10),
        legend.justification = c(1, 1), legend.box = "horizontal",
        legend.position = c(1, 1), # legend.position = c(0.725, 0.85),
        legend.spacing.y = unit(0.2, "cm"),
        legend.text=element_text(size=9),
        legend.background = element_blank(),
        legend.box.background = element_rect(fill = gray(0.96), color = NA))

#### Plot B: Change in prevalence ####
dfF <- dfF %>% mutate(SC = factor(SC, levels = rank))

plot1B <- ggplot() + ylim(c(-0.09,0.08)) +
  geom_hline(yintercept=0, lty="dashed",size=0.65) +
  scale_colour_manual(values = c("#143c77","darkred","mediumpurple4"),
                      labels = c("Non-vaccine type", "Vaccine type", "Mixed"),
                      name = "SC Composition") +
  scale_linetype_manual(values = c(1), name = NULL) +
  scale_shape_manual(values = c("-", "+")) + theme_classic() +
  geom_point(data=dfF, aes(SC, delta, col = vaccine),
             size =4, alpha  = 0.85, show.legend = F) +
  geom_pointrange(data=dfF, aes(x=SC, y=deltaExp,
                                ymin=CI_low, ymax=CI_up, lty=expectation),
                  size=.15, fatten = 6, show.legend = F) +
  geom_point(data=dfF, aes(x=SC, y=-0.09, shape = signif), size=6,
```

```
            col = "lightsteelblue4", show.legend = F) +
  labs(x="Strain (SC)") + labs(y="Change in prevalence") +
  theme(legend.title = element_text(face="bold", size = 10),
        legend.text=element_text(size=9),
        legend.position = c(0.01, 1),
        legend.justification = c(0.01, 1),
        legend.box = "horizontal",
        legend.background = element_blank(),
        legend.box.background = element_rect(fill = gray(0.95), color = NA))

#### Final figure 1 ####
figure1 <- plot_grid(plot1A, plot1B, labels = c("A","B"), ncol = 1)
ggsave("figure1.png", figure1, width = 14,height = 7)
```

Figure 2: Simulations

```
#### Funct. rootfun ####
rfun <- function(t, state, pars){
  dstate <- unlist(repEq(t, state, pars)) # rate of change vector
  return(sum(abs(dstate)) - 1e-4)
}

#### Funct. checkFeas ####
checkFeas <- function(e, g){
  g <- as.data.frame(g)
  temp <- sapply(g, function(x)(length(unique(x))))
  id <- which(temp==1)
  e[id] <- as.numeric(unique(g[,id]))
  return(e)
}

#### Funct. replicator ####
repEq <- function(t, Nf, pars){
  with(as.list(c(Nf, pars)), {
    f <- NULL
    dfdt <- rep(0, nSC)
    xifi <- rep(0, nSC)

    ## loci frequencies ##
    for(k in 1:nCOG){ f[k] <- sum(Nf*genot[,k]) }

    for(k in 1:nSC){
      for(l in 1:nCOG){
        xifi[k] <- xifi[k] + Nf[k]*(genot[k,l]*(eqbm[l] - f[l]))
      }
    }

    ## dfdt ##
    for(k in 1:nSC){ dfdt[k] <- xifi[k] - Nf[k]*sum(xifi) }

    return(list(dfdt))
```

```
  })
}


####################################

#### Pre-intervation ####
#### Parameters ####
nCOG <- 10
nSC <- 8
eqbm <- c(0.5677,0.5138,0.4050,0.4388,0.4981,
          0.5065,0.5725,0.4513,0.5811,0.4034)
vacT <- c(2,3,5)
timeSteps <- 0.5
posCom <- 2^nCOG
genot <- data.frame(permutations(n=2,r=nCOG,v=c(0,1), repeats.allowed = T))
colnames(genot) <- as.character(1:nCOG)
pres <- c(7,193,320,337,340,621,674,842)
genotF <- c(0.2342,0.1511,0.0033,0.1248, 0.1750,0.0067,0.2219,0.083)
genotP <- genot[pres,]
times <- seq(from=0, to=1000, by=timeSteps)
pars <- list(eqbm = eqbm, nCOG = nCOG, nSC = nSC, genot = genotP)

#### Simulations ####
out1 <- out1P <- as.data.frame(lsodar(func=repEq,y=genotF,times=times,parms=pars,rootfun=rfun))
out1 <- out1 %>% pivot_longer(-time, names_to = "genotype", values_to = "frequency") %>%
  mutate(genotype = paste('G', genotype, sep = "")) %>% subset(frequency > 0)

out1_pre <- out1 %>% mutate(time = time/3)
E1 <- round(as.numeric(out1P[nrow(out1P),-1]), digits = 5) ### last time in the data frame

#### Intervation ####
#### Parameters ####
E2 <- E1
E2[vacT] <- 0 ### remove genotypes G2 (001) and G6 (101) 'vaccine types'
E2 <- E2/sum(E2)
idZ <- which(E2 > 0)
timeS <- 40
genotPV <- genotP[idZ,]
eqbmPV <- checkFeas(eqbm, genotPV)
pars$eqbm <- eqbmPV
pars <- list(eqbm = eqbm, nCOG = nCOG, nSC = nSC, genot = genotP)

rfun <- function(t, state, pars){
  dstate <- unlist(repEq(t, state, pars)) # rate of change vector
  return(sum(abs(dstate)) - 1e-6)
}

### Simulations ###
out2 <- out2P <- as.data.frame(lsodar(func=repEq,y=E2,times=times,parms=pars,rootfun=rfun))
out2 <- out2 %>% mutate(time = time + timeS) %>%
  pivot_longer(-time, names_to = "genotype", values_to = "frequency") %>%
  mutate(genotype = paste('G', genotype, sep = "")) %>% subset(time <= 80)
```

```
outF <- rbind(out1_pre, out2)
E3 <- round(as.numeric(out2P[nrow(out2P),-1]), digits = 5)

lineT <- data.frame(linetype = "Non-vaccine type",
                    genotype = paste('G', 1:8, sep = ""), index = 1:8)
lineT <- lineT %>% mutate(linetype = ifelse(index %in% vacT, "Vaccine type",
                                            "Non-vaccine type"), index = NULL)

outF <- outF %>% full_join(lineT) %>%
  mutate(frequency = ifelse(frequency == 0,
                            runif(1, min = 0.00001, max = 0.00002), frequency))

E3 <- outF %>% subset(time == max(outF$time)) %>% arrange(-frequency)
alphaT <- data.frame(genotype = E3$genotype,
                     alpha = c(seq(from=1,to=0.9,length.out =5),1,1,1))
outF <- full_join(outF, alphaT)

#### Figure A ####
plot2A <- ggplot(outF, aes(time, frequency, group = genotype, alpha = alpha,
                           colour = factor(linetype), linetype = factor(linetype))) +
  annotate("rect", xmin = 31.5, xmax = 37.5, ymin = -Inf,
           ymax = 0.28, fill = "gray93", colour = NA) +
  annotate("rect", xmin = 69, xmax = 75, ymin = -Inf,
           ymax = 0.28, fill = "gray93", colour = NA) +
  annotate("rect", xmin = 40, xmax = 42, ymin = -Inf,
           ymax = 0.28, fill = "darkslategray4", colour = NA, alpha = 0.3) +
  geom_line(size = 1) + theme_classic() +
  scale_colour_manual(values = c('#143c77', "darkred")) + xlim(0,75) + #899DA4 #C93312
  #scale_linetype_manual(values = c("solid", "dashed")) +
  geom_segment(x = 38.2, y = 0.3175, xend = 38.2, yend = 0.265,
               arrow = arrow(length = unit(0.25, "cm")),
               colour = 'black', show.legend = F) +
  annotate("text", x = 38.2, y = 0.33, label="Vaccine introduction",
           fontface="bold", color="black", size = 3.7) +
  #geom_segment(aes(x = 65, y = 0.29, xend = 74, yend = 0.29), colour = 'black') +
  annotate("text", x = 34.5, y = 0.295, label="Pre-vaccine\nequilibrium",
           color="dimgrey", fontface="bold", size = 2.65) +
  annotate("text", x = 72, y = 0.295, label="Post-vaccine\nequilibrium",
           color="dimgrey", fontface="bold", size = 2.65) +
  annotate("text", x = 41, y = 0.295, label="Predicted\nFitness",
           color="darkslategray", fontface="bold", size = 2.5) +
  ylab("Strain Prevalence") + xlab("Time") +
  scale_alpha(range = c(0.15, 1), guide = 'none') +
  theme(legend.justification=c(0,0),legend.position=c(0.02,0.85),
        legend.title=element_blank(),
        axis.text = element_text(colour = "black"),
        legend.text=element_text(size=8),
        plot.title = element_text(hjust = 0, size=10, face="bold"))

###################################
COGs <- 2371 ## 2371
SCs <- 35
dataOutF <- data.frame(NULL)
```

```r
replicates <- 10
VTselect <- 3

for(int in 1:replicates){
  el0 <- runif(COGs, min = 0.05, max = 0.95)
  dat <- as.matrix(replicate(SCs, sample(c(0,1), COGs, replace = TRUE)))
  dat <- unique(dat, MARGIN = 2)
  VT <- sample(1:SCs, VTselect)
  NVT <- (1:SCs)[-VT]

  if(ncol(dat) < SCs){
    cat("run again\n")
    break
  }

  #### E1 = pre-vaccine frequencies ####
  x1 <- QP(dat, as.matrix(el0))

  ### re-calculate el ###
  el <- dat %*% as.matrix(x1)

  #### E2 - frequencies just after vaccine intro ####
  x2 <- x1[-VT]
  x2 <- round(x2/sum(x2), digits = 5)
  dat2 <- dat[,-VT]
  fl <- dat2 %*% as.matrix(x2)

  #### E3 - frequencies long-term post-vaccine ####
  x3 <- QP(dat2, el)
  el3 <- dat2 %*% as.matrix(x3)

  #### fitness function "omega" just after vaccine intro ####
  whole <- as.numeric(el - fl) ### this is similar to (el - fl) and thus fitness.
  deltaE <- x3 - x2
  omega <- as.vector(t(dat2) %*% whole) ## length SCs, similar to FFS

  phi <- sum(x2 * omega) ## average fitness
  rateOfChange <- omega - phi ## "rate of change": omega_g - phi

  ################### Relative fitness ###################
  dataOut <- data.frame(riskDif = deltaE, r = rateOfChange)
  dataOut <- subset(dataOut, riskDif != 0)
  dataOut <- mutate(dataOut, change = ifelse(riskDif < 0, "Decreased", "Increased"), rep = int)
  dataOutF <- rbind(dataOutF, dataOut)
}

dataOutF <- dataOutF %>% mutate(col = ifelse(sign(riskDif) == sign(r), "same", "diff"))

#### Figure B ####
plot2B <- ggplot(dataOutF, aes(x=change, y=r,
                               group=factor(rep), colour = col)) +
  geom_point(position = position_dodge(width = 1), size = 2.5, alpha  = 0.6) +
  geom_hline(yintercept = 0, linetype = 3) + theme_classic() +
```

```
  xlab("Observed prevalence change") +
  ylab("Predicted Fitness") +
  theme(legend.position="none", axis.text = element_text(colour = "black"),
        plot.title = element_text(hjust = 0, size=10, face="bold")) +
  scale_colour_manual(values = c("gray70", "darkslategray4"))

figure2 <- plot_grid(plot2A, plot2B, labels = c("A", "B"), rel_widths = c(1.75,1))
ggsave("figure2.png", figure2, width = 14, height = 4.5)
```

Figure 3: Prediction (Southwest US data)

```
#### Data ####
dfFVT <- df %>% select(BAPS2,PCV7.actual, Epoch1) %>%
  group_by(Epoch1) %>%  count(BAPS2,PCV7.actual) %>%
  mutate(freq = round(prop.table(n), digits = 3)) %>% ungroup() %>%
  select(Epoch1, BAPS2, PCV7.actual, freq) %>%
  spread(Epoch1, freq, fill = 0) %>% arrange(BAPS2,PCV7.actual)

#### Present at E1 ####
SCE1 <- dfFVT %>% subset(E1 > 0) %>% ## & BAPS2 !="27"
  select(BAPS2, PCV7.actual) %>%
  mutate(Epoch1 = "E1")

#### NVTs Present at E1 ####
SCE2 <- SCE1 %>% subset(PCV7.actual == "NVT"  & BAPS2 !="27")


SC_freq_df <- df %>% select(BAPS2, PCV7.actual, Epoch1,
                            HMPREF0837_12128:HMPREF0837_10616) %>%
  arrange(BAPS2) %>% group_by(BAPS2,PCV7.actual,Epoch1) %>%
  mutate(SC_n = n()) %>% ungroup() %>%
  group_by(BAPS2,PCV7.actual,Epoch1,SC_n) %>%
  summarise_at(vars(HMPREF0837_12128:HMPREF0837_10616),mean) %>%
  ungroup()

### Get the matrix and the SC for the pre-vaccine epch "E1"
df_preV <- SCE1 %>% left_join(SC_freq_df)
SC_freq_preV <- as.matrix(df_preV %>% mutate(SC_freq=SC_n/sum(SC_n)) %>% select(SC_freq))
SC_COG_preV <- as.matrix(t(df_preV %>% select(HMPREF0837_12128:HMPREF0837_10616)))

#### Get e_l ####
el <- SC_COG_preV %*% SC_freq_preV

#### Figure A ####
dfImputed <- dfFVT %>% subset(PCV7.actual == "NVT" & E1 == 0) %>%
  select(BAPS2,PCV7.actual) %>% mutate(Epoch1 = "E1", n=1) %>%
  select(Epoch1,BAPS2,PCV7.actual,n)

dfFVTImputed <- df %>% select(BAPS2,PCV7.actual, Epoch1) %>%
  group_by(Epoch1) %>%  count(BAPS2,PCV7.actual) %>%
  ungroup() %>% bind_rows(dfImputed) %>% group_by(Epoch1) %>%
```

```
  mutate(freq = round(prop.table(n), digits = 3)) %>% ungroup() %>%
  select(Epoch1, BAPS2, PCV7.actual, freq) %>%
  spread(Epoch1, freq, fill = 0) %>% arrange(BAPS2,PCV7.actual)

dfFNVTImputed <- dfFVTImputed %>%
  subset(PCV7.actual == "NVT" & BAPS2 !="27") %>%
  mutate(deltaE = E3 - E1) %>% arrange(BAPS2)

#### E2 - frequencies just after vaccine intro ####
x_imputed <- dfFNVTImputed$E1
x_imputed <- as.matrix(round(x_imputed/sum(x_imputed), digits = 5))

dat2_imputed <- dfImputed %>% mutate(Epoch1 = "E2") %>%
  select(BAPS2,PCV7.actual,Epoch1) %>% bind_rows(SCE2) %>%
  arrange(BAPS2) %>% left_join(SC_freq_df) %>%
  select(HMPREF0837_12128:HMPREF0837_10616)
dat2_imputed <- as.matrix(dat2_imputed)

fl_imp <- t(dat2_imputed) %*% x_imputed

#### fitness function "omega" just after vaccine intro ####
whole <- as.numeric(el - fl_imp) ### this is similar to (el - fl) and thus fitness.
deltaE <- dfFNVTImputed$deltaE
omega <- as.vector(dat2_imputed %*% whole) ## length SCs, similar to FFS

phi <- sum(x_imputed * omega) ## average fitness
rateOfChange <- omega - phi ## "rate of change": omega_g - phi

dat3AB <- dfFNVTImputed %>% select(BAPS2, deltaE) %>%
  mutate(r = rateOfChange) %>% subset(deltaE != 0) %>%
  mutate(change = ifelse(deltaE < 0, "Decreased", "Increased")) %>%
  mutate(col = ifelse(sign(deltaE) == sign(r), "same", "diff")) %>%
  left_join(vaccineT)

highlight <- dat3AB %>% filter(col == "diff")

lm3 <- lm(r~deltaE,dat3AB)
summary(lm3)
```

Call: lm(formula = r ~ deltaE, data = dat3AB)

Residuals: Min 1Q Median 3Q Max -12.7964 -1.8504 -0.1459 2.3969 10.6031

Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.4151 1.0220 0.406 0.688
deltaE 182.1310 38.6050 4.718 5.55e-05 *** — Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ’’ 0.05 ‘.’ 0.1 ’ ’ 1

Residual standard error: 4.995 on 29 degrees of freedom Multiple R-squared: 0.4342, Adjusted R-squared: 0.4147 F-statistic: 22.26 on 1 and 29 DF, p-value: 5.555e-05

```
#### Figure A ####
plot3A <- ggplot(dat3AB, aes(x=deltaE, y=r, colour = vaccine)) +
  theme_classic() + xlab("Observed Prevalence Change") +
  ylab("Predicted Fitness") + xlim(-0.03, 0.065) + ylim(-12,20) +
  geom_hline(yintercept=0, lty="dotted", alpha=.6) +
  geom_vline(xintercept=0, lty="dotted", alpha=.6) +
```

```
  annotate("rect", xmin=-0.03,xmax=0,ymin=-12,ymax=0,
           fill="darkslategray4", alpha= 0.15) +
  annotate("rect", xmin=0,xmax=0.065,ymin=0,ymax=20,
           fill="darkslategray4", alpha= 0.15) +
  geom_smooth(aes(group = 1), color="grey70", method=lm,
              formula = y~x,  show.legend=FALSE, se=FALSE) +
  geom_point(size = 3) + theme(legend.position = "none",
                               axis.text = element_text(colour = "black"),) +
  scale_colour_manual(values = c("#143c77","mediumorchid4")) +
  geom_text_repel(aes(label = paste("SC", BAPS2, sep="-")), data = highlight, size = 3.5)


#### Figure B ####
df_postV <- SCE2 %>% left_join(SC_freq_df)
SC_COG_postV <- as.matrix(t(df_postV %>% select(HMPREF0837_12128:HMPREF0837_10616)))

SC_freq_postV_obs <- SCE2 %>% mutate(Epoch1 = "E3") %>%
  left_join(SC_freq_df) %>%
  mutate(SC_freq=SC_n/sum(SC_n, na.rm = T)) %>%
  select(BAPS2, PCV7.actual, SC_freq) %>%
  mutate(SC_freq = replace_na(SC_freq, 0))

## Predict postV frequencies
SC_freq_postV_pred <- QP(SC_COG_postV, el) #Matrix: rows = COGs, columns = (SCs - VT)
SC_freq_postV_obs <- SC_freq_postV_obs %>%
  mutate(SC_pred = SC_freq_postV_pred) %>%
  left_join(vaccineT)

outlier3B <- SC_freq_postV_obs %>% mutate(diff = abs(SC_freq - SC_pred))
outlier3B <- outlier3B %>% filter(diff %in% boxplot(outlier3B$diff, plot = FALSE)$out)

W_model3B <- lm(data=SC_freq_postV_obs,SC_freq~SC_pred);
summary(W_model3B)
```

Call: lm(formula = SC_freq ~ SC_pred, data = SC_freq_postV_obs)

Residuals: Min 1Q Median 3Q Max -0.059467 -0.014478 -0.002457 0.015118 0.053442

Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.012366 0.008616 1.435 0.16364
SC_pred 0.666107 0.198686 3.353 0.00255 ** — Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ’’ 0.05 ‘.’ 0.1 ’ ’ 1

Residual standard error: 0.02329 on 25 degrees of freedom Multiple R-squared: 0.3101, Adjusted R-squared: 0.2826 F-statistic: 11.24 on 1 and 25 DF, p-value: 0.002551

```
confint(W_model3B)
```

```
                2.5 %     97.5 %
```

(Intercept) -0.005380061 0.03011185 SC_pred 0.256905970 1.07530899

```
linearHypothesis(W_model3B, c("(Intercept) = 0", "SC_pred = 1"), test = "Chisq")
```

Linear hypothesis test

Hypothesis: (Intercept) = 0 SC_pred = 1

Model 1: restricted model Model 2: SC_freq ~ SC_pred

Res.Df RSS Df Sum of Sq Chisq Pr(>Chisq) 1 27 0.015093
2 25 0.013561 2 0.0015319 2.8241 0.2436

```
linearHypothesis(W_model3B, c("(Intercept) = 0", "SC_pred = 1"))
```

Linear hypothesis test

Hypothesis: (Intercept) = 0 SC_pred = 1

Model 1: restricted model Model 2: SC_freq ~ SC_pred

Res.Df RSS Df Sum of Sq F Pr(>F) 1 27 0.015093
2 25 0.013561 2 0.0015319 1.412 0.2624

```
plot3B <- ggplot(SC_freq_postV_obs,
                 aes(x = SC_pred, y = SC_freq, colour = vaccine)) +
  geom_segment(aes(x=0,xend=0.12,y=0,yend=0.12),
               color="black",alpha=.7,lwd=0.5,lty=3) +
  theme(legend.position = "none") + theme_classic() +
  geom_smooth(method='lm', color="#899DA4" ,
              formula=y~x, alpha=0.3, lwd=.6,
              fullrange=T, linetype="blank", show.legend=F) +
  annotate(geom = "text", x=0.11, y =0.114,
           label = "1:1 line", angle = 45, size = 3) +
  geom_point(size=3) +  ##
  scale_x_continuous("Predicted Prevalence (NFDS)")+
  scale_y_continuous("Observed Prevalence") +
  coord_fixed(ratio = 1, xlim=c(0,0.12), ylim=c(0,0.12)) +
  scale_colour_manual(values = c("#143c77","mediumorchid4")) +
  theme(legend.position = "none",
        axis.text = element_text(colour = "black"),) +
  geom_text_repel(aes(label = paste("SC", BAPS2, sep = "-")), data = outlier3B, size = 3.5)


#### Figure C ####

SC_freq_E1 <- df_preV %>%
  mutate(SC_freq_E1=SC_n/sum(SC_n)) %>%
  select(BAPS2,PCV7.actual,SC_freq_E1)

SC_freq_postV_diff <- SC_freq_postV_obs %>%
  left_join(SC_freq_E1) %>%
  mutate(diff_pred = SC_freq_E1 - SC_pred,
         diff_obs = SC_freq_E1 - SC_freq) %>%
  left_join(vaccineT)

SC_freq_postV_diff <- SC_freq_postV_diff %>%
  mutate(diff = abs(diff_pred - diff_obs)) %>%
  arrange(diff)

stats <- summary(lm(SC_freq_postV_diff$diff_pred~SC_freq_postV_diff$diff_obs))
ars <- round(stats$adj.r.squared, digits = 3)
sseE <- round(sse(SC_freq_postV_diff$diff_obs, SC_freq_postV_diff$diff_pred), digits = 3)
maeE <- round(mae(SC_freq_postV_diff$diff_obs, SC_freq_postV_diff$diff_pred),digits = 3)##Mean Absolute
rmseE <- round(rmse(SC_freq_postV_diff$diff_obs, SC_freq_postV_diff$diff_pred),digits = 3) ##Root Mean

accNFDS <- data.frame(Model = "Accesory genome (NFDS)", nloci = length(el),
```

```
                      adj.r.squared = ars, SSE = sseE, RMSE = rmseE)


W_model3C <- lm(data=SC_freq_postV_diff,diff_obs~diff_pred);
summary(W_model3C)
```

Call: lm(formula = diff_obs ~ diff_pred, data = SC_freq_postV_diff)

Residuals: Min 1Q Median 3Q Max -0.057236 -0.011661 0.002054 0.011921 0.067643

Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.003684 0.006783 -0.543 0.5919
diff_pred 0.795065 0.273307 2.909 0.0075 ** — Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ’’ 0.05 ‘.’ 0.1 ’ ’ 1

Residual standard error: 0.0243 on 25 degrees of freedom Multiple R-squared: 0.2529, Adjusted R-squared: 0.223 F-statistic: 8.463 on 1 and 25 DF, p-value: 0.007504

```
confint(W_model3C)
```

```
               2.5 %     97.5 %
```

(Intercept) -0.01765401 0.01028661 diff_pred 0.23217925 1.35795061

```
linearHypothesis(W_model3C, c("(Intercept) = 0", "diff_pred = 1"), test = "Chisq")
```

Linear hypothesis test

Hypothesis: (Intercept) = 0 diff_pred = 1

Model 1: restricted model Model 2: diff_obs ~ diff_pred

Res.Df RSS Df Sum of Sq Chisq Pr(>Chisq) 1 27 0.015093
2 25 0.014761 2 0.00033197 0.5623 0.7549

```
outlier3C <- SC_freq_postV_diff %>%
  filter(diff %in% boxplot(SC_freq_postV_diff$diff, plot = FALSE)$out)

#### Figure C ####
plot3C <- ggplot(SC_freq_postV_diff, aes(x = diff_pred,
                                         y = diff_obs, colour = vaccine)) +
  geom_segment(aes(x=-0.1,xend=0.05,y=-0.1,yend=0.05),
               color="black",alpha=.7,lwd=0.5,lty=3) +
  geom_smooth(method='lm', color="gray80" ,formula=y~x,
              alpha=0.3, lwd=.6, fullrange=T,
              linetype="blank", show.legend=F) +
  annotate(geom = "text", x=0.0375, y =0.0425,
           label = "1:1 line", angle = 45, size = 3) +
  geom_point(size=3) + theme_classic() +
  scale_colour_manual(values = c("#143c77","mediumorchid4"),
                      labels = c("Non-vaccine type", "Mixed"),
                      name = "SC Composition") +
  theme(legend.position = "none",
        axis.text = element_text(colour = "black"),) +
  xlab("Predicted Prevalence Change (NFDS)")+
  ylab("Observed Prevalence Change") +
  coord_fixed(ratio = 1, xlim=c(-0.1,0.05), ylim=c(-0.1,0.05)) +
  annotate("text",x=-0.097,y=0.04, size=2.5,hjust = 0,
           label=paste("SSE = ", sseE, "\nRMSE = ",
                       rmseE, "\nAdj. R2 = ", ars)) +
```

```
  geom_text_repel(aes(label = paste("SC", BAPS2, sep = "-")), data = outlier3C, size = 3.5)

#### Figure D ####
SC_freq_postV_diff <- SC_freq_postV_diff %>%
  mutate(prorata = SC_freq_E1/sum(SC_freq_E1)) %>%
  mutate(diff_predPro = SC_freq_E1 - prorata) %>%
  mutate(diffP = abs(diff_predPro - diff_obs))

stats <- summary(lm(SC_freq_postV_diff$diff_predPro~SC_freq_postV_diff$diff_obs))
ars <- round(stats$adj.r.squared, digits = 3)
sseE <- round(sse(SC_freq_postV_diff$diff_obs, SC_freq_postV_diff$diff_predPro), digits = 3)
maeE <- round(mae(SC_freq_postV_diff$diff_obs, SC_freq_postV_diff$diff_predPro),digits = 3)##Mean Absol
rmseE <- round(rmse(SC_freq_postV_diff$diff_obs, SC_freq_postV_diff$diff_predPro),digits = 3) ##Root Me

accProrata <- data.frame(Model = "Accesory genome (Pro rata)", nloci = length(el),
                         adj.r.squared = ars, SSE = sseE, RMSE = rmseE)

W_model3D <- lm(data=SC_freq_postV_diff,diff_obs~diff_predPro);
summary(W_model3D)
```

Call: lm(formula = diff_obs ~ diff_predPro, data = SC_freq_postV_diff)

Residuals: Min 1Q Median 3Q Max -0.053397 -0.014551 -0.001313 0.021763 0.038569

Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.027719 0.009368 -2.959 0.00666 ** diff_predPro -0.541862 0.431660 -1.255 0.22098
— Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ’’ 0.05 ‘.’ 0.1 ’ ’ 1

Residual standard error: 0.02727 on 25 degrees of freedom Multiple R-squared: 0.05929, Adjusted R-squared: 0.02167 F-statistic: 1.576 on 1 and 25 DF, p-value: 0.221

```
confint(W_model3D)
```

```
                2.5 %       97.5 %
```

(Intercept) -0.04701286 -0.008425814 diff_predPro -1.43088194 0.347158089

```
linearHypothesis(W_model3D, c("(Intercept) = 0", "diff_predPro = 1"), test = "Chisq")
```

Linear hypothesis test

Hypothesis: (Intercept) = 0 diff_predPro = 1

Model 1: restricted model Model 2: diff_obs ~ diff_predPro

Res.Df RSS Df Sum of Sq Chisq Pr(>Chisq)
1 27 0.028071
2 25 0.018586 2 0.0094854 12.759 0.001696 ** — Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ’’ 0.05 ‘.’ 0.1 ’ ’ 1

```
plot3D <- ggplot(SC_freq_postV_diff, aes(x = diff_predPro,
                                         y = diff_obs, colour = vaccine)) +
  geom_segment(aes(x=-0.1,xend=0.05,y=-0.1,yend=0.05),
               color="black",alpha=.7,lwd=0.5,lty=3) +
  geom_smooth(method='lm', color="gray80" ,formula=y~x,
              alpha=0.3, lwd=.6, fullrange=T,
              linetype="blank", show.legend=F) +
  annotate(geom = "text", x=0.0375, y =0.0425,
           label = "1:1 line", angle = 45, size = 3) +
  geom_point(size=3) + theme_classic() +
```

```
  scale_colour_manual(values = c("#143c77","mediumorchid4"),
                      labels = c("Non-vaccine type", "Mixed"),
                      name = "SC Composition") +
  theme(legend.position = "none",
        axis.text = element_text(colour = "black"),) +
  xlab("Predicted Prevalence Change (Pro rata)")+
  coord_fixed(ratio = 1, ylim=c(-0.1,0.05), xlim=c(-0.1,0.05)) +
  ylab("Observed Prevalence Change") +
  annotate("text",x=-0.097,y=0.04, size=2.5,hjust = 0,
           label=paste("SSE = ", sseE, "\nRMSE = ",
                       rmseE, "\nAdj. R2 = ", ars))


#### Combine figure 3 ####
ptitle1 <- ggplot() + theme_void() +
  annotate("rect", fill = "darkslategray4", alpha = 0.3,
           xmin = 0, xmax = 1, ymin = 0, ymax = 1)
ptitle1 <- ggdraw(ptitle1) + draw_label("Predicted Fitness",
                                        fontface='bold', colour = "darkslategrey") ## ,
pA <- plot_grid(plot3A, labels = c("A"))
pA <- plot_grid(ptitle1, pA, ncol = 1, scale = c(1,0.9), rel_heights=c(0.075, 1))


ptitle2 <- ggplot() + theme_void() +
  annotate("rect", fill = "gray93", xmin = 0, xmax = 1, ymin = 0, ymax = 1)
ptitle2 <- ggdraw(ptitle2) + draw_label("Post-vaccine Equilibrium Frequencies",
                                        fontface='bold', colour = "gray30") ## ,
pBCD <- plot_grid(plot3B, plot3C, plot3D, ncol = 3, labels = c("B","C","D"))
pBCD <- plot_grid(ptitle2, pBCD, ncol = 1, scale = c(1,0.9), rel_heights=c(0.075, 1))
plot3 <- plot_grid(pA, pBCD, rel_widths = c(1,3), rel_heights = c(1,1))

ggsave("figure3.png", plot3, width = 16, height = 4)
```

Fig. Suppl 1: Tree

```
## The tree was modified after being produced (style purposes)

readMatrix<-function(heatmapData){
  if (is.matrix(heatmapData)) {
    x = data.frame(heatmapData)
  }
  else if (is.data.frame(heatmapData)) {
    x = heatmapData
  }
  else {
    x<-read.csv(heatmapData,row.names=1)
  }
  x
}
getLayout<-function(infoFile,infoCols,heatmapData,barData,doBlocks,
                    treeWidth=10,infoWidth=10,dataWidth=30,edgeWidth=1,
```

```
                    labelHeight=10,mainHeight=100,barDataWidth=10,
                    blockPlotWidth=10) {
# m = layout matrix
# w = layout widths vector
# h = layout height vector

# tree
w = c(edgeWidth,treeWidth)
m<-cbind(c(0,0,0),c(0,1,0)) # first two columns, edge + tree
x = 1

# info
if (!is.null(infoFile)) { # info is provided

  printCols = TRUE
  if (!is.null(infoCols)) {
    if (is.na(infoCols)) {
      printCols = FALSE
    }}

  if (printCols) {
    x = x + 1
    m<-cbind(m,c(0,x,0))
    w = c(w,infoWidth)
  }
}

# heatmap
if (!is.null(heatmapData)) {
  x = x + 1
  m<-cbind(m,c(x+1,x,0)) # add heatmap & labels
  x = x + 2
  m[1,2] = x # add heatmap scale above tree
  w = c(w,dataWidth)
}

# barplot
if (!is.null(barData)) {
  x = x + 1
  m<-cbind(m,c(0,x,x+1)) # barplot and scale bar
  x = x + 1
  w = c(w,barDataWidth)
}

if (doBlocks) {
  x = x + 1
  m<-cbind(m,c(0,x,0)) # recomb blocks
  w = c(w,blockPlotWidth)
}

# empty edge column
m<-cbind(m,c(0,0,0))
w = c(w,edgeWidth)
```

```
  if (!is.null(heatmapData) | !is.null(barData)) { h = c(labelHeight,mainHeight,labelHeight) }
  else { h = c(edgeWidth,mainHeight,edgeWidth) }

  return(list(m=as.matrix(m),w=w,h=h))
}

plotTree<-function(tree,ladderise=NULL,heatmapData=NULL,barData=NULL,infoFile=NULL,
                   blockFile=NULL,snpFile=NULL,gapChar="?",genome_size=5E6,blwd=5,
                   block_colour="black",snp_colour="red",genome_offset=0,
                   colourNodesBy=NULL,infoCols=NULL,outputPDF=NULL,outputPNG=NULL,
                   w,h,heatmap.colours=rev(gray(seq(0,1,0.1))),tip.labels=F,
                   tipLabelSize=1,offset=0,tip.colour.cex=0.5,legend=T,
                   legend.pos="bottomleft",ancestral.reconstruction=F,cluster=NULL,
                   tipColours=NULL,lwd=1.5,axis=F,axisPos=3,edge.color="black",
                   infoCex=0.8,colLabelCex=0.8,treeWidth=10,infoWidth=10,dataWidth=30,
                   edgeWidth=1,labelHeight=10,mainHeight=100,barDataWidth=10,
                   blockPlotWidth=10,barDataCol=2,heatmapBreaks=NULL,
                   heatmapDecimalPlaces=1,vlines.heatmap=NULL,vlines.heatmap.col=2,
                   heatmap.blocks=NULL,pie.cex=0.5) {

  require(ape)

  # PREPARE TREE, CHOOSE LADDERISATION OR NOT, AND GET TIP ORDER
  if (is.character(tree)){
    t<-read.tree(tree)
  }
  else t<-tree
  if (is.null(ladderise))
  {
    tl<-t
  }
  else if (ladderise=="descending")
  {
    tl<-ladderize(t, T)
  }
  else if (ladderise=="ascending")
  {
    tl<-ladderize(t, F)
  }
  else if (!is.null(ladderise))
  {
    print("Ladderise option should be exactly 'ascending' or 'descending'.  Any other command will raise
  }
  tips<-tl$edge[,2]
  tip.order<-tips[tips<=length(tl$tip.label)]
  tip.label.order<-tl$tip.label[tip.order] # for ordering data. note that for tiplabel(), the order is


  # PREPARE HEATMAP DATA
  if (!is.null(heatmapData)) {

    # read heatmap data and convert to data frame
    x<-readMatrix(heatmapData)
```

```
  # order rows of heatmap matrix to match tree
  y.ordered<-x[tip.label.order,]

  # reorder columns?
  if (!is.null(cluster)) {
    if (!(cluster==FALSE)) {

      if (cluster=="square" & ncol(y.ordered)==nrow(y.ordered)) {
        # order columns to match row order
        original_order<-1:nrow(x)
        names(original_order)<-rownames(x)
        reordered<-original_order[tip.label.order]
        y.ordered<-y.ordered[,rev(as.numeric(reordered))]
      }

      else {
        # cluster columns
        if (cluster==TRUE) {cluster="ward"} # set default clustering algorithm
        h<-hclust(dist(t(na.omit(y.ordered))),cluster)
        y.ordered<-y.ordered[,h$order]
      }

    }} # finished reordering columns

} # finished setting up heatmap data


# PREPARE BAR PLOT
if (!is.null(barData)) {
  b<-readMatrix(barData)
  barData<-b[,1]
  names(barData)<-rownames(b)
}

# PREPARE INFO TO PRINT
if (!is.null(infoFile)) {
  info<-readMatrix(infoFile)
  info.ordered<-info[rev(tip.label.order),]
}
else {info.ordered=NULL}


# PREPARE DISCRETE TRAIT FOR COLOURING NODES AND INFERRING ANCESTRAL STATES
ancestral=NULL
nodeColourSuccess=NULL
if (!is.null(colourNodesBy) & !is.null(infoFile)) {

  if (colourNodesBy %in% colnames(info.ordered)) {
    nodeColourSuccess = TRUE
    loc1<-info.ordered[,which(colnames(info.ordered)==colourNodesBy)]

    # assign values
    tipLabelSet <- character(length(loc1))
```

```
    names(tipLabelSet) <- rownames(info.ordered)
    groups<-table(loc1,exclude="")
    n<-length(groups)
    groupNames<-names(groups)

    # set colours
    if (is.null(tipColours)){ colours<-rainbow(n) }
    else{ colours<-tipColours }

    # assign colours based on values
    for (i in 1:n) {
      g<-groupNames[i]
      tipLabelSet[loc1==g]<-colours[i]
    }
    tipLabelSet <- tipLabelSet[tl$tip]

    # ancestral reconstruction
    if (ancestral.reconstruction) { ancestral<-ace(loc1,tl,type="discrete") }

  }}
# finished with trait labels and ancestral reconstruction


# OPEN EXTERNAL DEVICE FOR DRAWING
# open PDF for drawing
if (!is.null(outputPDF)) {
  pdf(width=w,height=h,file=outputPDF)
}
# open PNG for drawing
if (!is.null(outputPNG)) {
  png(width=w,height=h,file=outputPNG)
}


# SET UP LAYOUT FOR PLOTTING
doBlocks <- (!is.null(blockFile) | !is.null(snpFile))
l <- getLayout(infoFile,infoCols,heatmapData,barData,doBlocks,
               treeWidth=treeWidth,infoWidth=infoWidth,dataWidth=dataWidth,
               edgeWidth=edgeWidth,labelHeight=labelHeight,mainHeight=mainHeight,
               barDataWidth=barDataWidth,blockPlotWidth=blockPlotWidth)
layout(l$m, widths=l$w, heights=l$h)


# PLOT TREE
par(mar=rep(0,4))
tlp<-plot.phylo(tl,no.margin=T,show.tip.label=tip.labels,label.offset=offset,
                edge.width=lwd,edge.color=edge.color,xaxs="i", yaxs="i",
                y.lim=c(0.5,length(tl$tip)+0.5),cex=tipLabelSize)

# colour by trait
if (!is.null(nodeColourSuccess)) {
  tiplabels(col= tipLabelSet,pch=16,cex=tip.colour.cex)
  if (ancestral.reconstruction) { nodelabels(pie=ancestral$lik.anc,
```

```
                                                cex=pie.cex, piecol=colours) }
  if (legend) { legend(legend.pos,legend=groupNames,fill=colours) }
}

if (axis) { axisPhylo(axisPos) }

# PLOT INFO
if (!is.null(infoFile)) { # info is provided

  printCols = TRUE
  if (!is.null(infoCols)) {
    if (is.na(infoCols)) {
      printCols = FALSE
    }}

  if (printCols) {

    par(mar=rep(0,4))

    if (!is.null(infoCols)) {infoColNumbers = which(colnames(info.ordered) %in% infoCols)}
    else { infoColNumbers = 1:ncol(info.ordered)}

    plot(NA,axes=F,pch="",xlim=c(0,length(infoColNumbers)+1.5),
         ylim=c(0.5,length(tl$tip)+0.5),xaxs="i",yaxs="i")

    # plot all info columns
    for (i in 1:length(infoColNumbers)) {
      j<-infoColNumbers[i]
      text(x=rep(i+1,nrow(info.ordered)+1),y=c((nrow(info.ordered)):1),
           info.ordered[,j],cex=infoCex)
    }

  }
}


# PLOT HEATMAP
if (!is.null(heatmapData)) {

  if (is.null(heatmapBreaks)) { heatmapBreaks = seq(min(y.ordered,na.rm=T),max(y.ordered,na.rm=T),leng

  # plot heatmap
  par(mar=rep(0,4), xpd=TRUE)
  image((1:ncol(y.ordered))-0.5,(1:nrow(y.ordered))-0.5, as.matrix(t(y.ordered)),col=heatmap.colours,b
        axes=F,xaxs="i", yaxs="i", xlab="",ylab="")

  # draw vertical lines over heatmap
  if (!is.null(vlines.heatmap)) {
    for (v in vlines.heatmap) {abline(v=v, col=vlines.heatmap.col)}
  }

  # overlay blocks on heatmap
  if (!is.null(heatmap.blocks)) {
```

```
    for (coords in heatmap.blocks) {rect(xleft=coords[1], 0, coords[2], ncol(y.ordered), col=vlines.h
  }


  # data labels for heatmap
  par(mar=rep(0,4))
  plot(NA, axes=F, xaxs="i", yaxs="i", ylim=c(0,2), xlim=c(0.5,ncol(y.ordered)+0.5))
  text(1:ncol(y.ordered)-0.5,rep(0,ncol(x)),colnames(y.ordered),
       srt=90, cex=colLabelCex, pos=4)

  # scale for heatmap
  par(mar=c(2,0,0,2))
  image(as.matrix(seq(min(y.ordered,na.rm=T),max(y.ordered,na.rm=T),
                      length.out=length(heatmap.colours)+1)),
        col=heatmap.colours,yaxt="n",breaks=heatmapBreaks,axes=F)
  axis(1,at=heatmapBreaks[-length(heatmapBreaks)]/max(y.ordered,na.rm=T),
       labels=round(heatmapBreaks[-length(heatmapBreaks)],heatmapDecimalPlaces))
}

# BARPLOT
if (!is.null(barData)) {
  par(mar=rep(0,4))
  barplot(barData[tip.label.order], horiz=T, axes=F, xaxs="i", yaxs="i", xlab="", ylab="", ylim=c(0.2
          col=barDataCol,border=0,width=0.5,space=1,names.arg=NA)

  # scale for barData plot
  par(mar=c(2,0,0,0))
  plot(NA, yaxt="n", xaxs="i", yaxs="i", xlab="", ylab="", ylim=c(0,2), xlim=c((-1)*max(barData,na.rm
}

# SNPS AND RECOMBINATION BLOCKS
if (doBlocks) {
  par(mar=rep(0,4))
  plot(NA,axes=F,pch="",xlim=c(genome_offset,genome_offset+genome_size+1.5),
       ylim=c(0.5,length(tl$tip)+0.5),xaxs="i",yaxs="i") # blank plotting area

  # plot snps
  if (!is.null(snpFile)) {
    snps<-read.csv(snpFile,header=F,row.names=1) # in case colnames start with numbers or contain das
    snps_strainCols <- snps[1,] # column names = strain names
    snps<-snps[-1,] # drop strain names

    for (strain in tip.label.order){
      # print SNPs compared to ancestral alleles in column 1
      s<-rownames(snps)[(as.character(snps[,1]) != as.character(snps[,which(snps_strainCols==strain)]
      y <- which(tip.label.order==strain)
      if (length(s)>0) {
        for (x in s) {
          points(x,y,pch="|",col=snp_colour,cex=0.25)
        }
      }
    }
  }
```

```
    # plot blocks
    if (!is.null(blockFile)){
      blocks<-read.delim(blockFile,header=F)
      for (i in 1:nrow(blocks)) {
        if (as.character(blocks[i,1]) %in% tip.label.order) {
          y <- which(tip.label.order==as.character(blocks[i,1]))
          x1 <- blocks[i,2]
          x2 <- blocks[i,3]
          lines(c(x1,x2),c(y,y),lwd=blwd,lend=2,col=block_colour)
        }
      }
    }

  } # finished with SNPs and recomb blocks

  # CLOSE EXTERNAL DRAWING DEVICE
  if (!is.null(outputPDF) | !is.null(outputPNG)) {
    dev.off()
  }

  # RETURN ordered info and ancestral reconstruction object
  if (!is.null(heatmapData)){mat=as.matrix(t(y.ordered))}
  else {mat=NULL}
  return(list(info=info.ordered,anc=ancestral,mat=mat,strain_order=tip.label.order))
}


tree<-read.tree("RAxML_bestTree.All.Core.tre") #Core genome tree
#tree<-read.tree("RAxML_bestTree.All.Binary.tre") #Accessory genome tree
#tree <- ladderize(midpoint.root(tree), right = FALSE) #Ladderized the tree and midpoint root


heatmap_colors <- c("#ECCBAE", "#A42820", "#5F5647", "#9A8822", "#74A089", "#D8B70A",
                    "#046C9A", "#3F5151", "#4E2A1E", "#F2300F", "#FF0000", "#02401B",
                    "#D69C4E", "#FAD510", "#CB2314", "#273046", "#00A08A", "#A2A475",
                    "#ABDDDE", "#550307", "#354823", "#F5CDB4", "#F2AD00", "#972D15",
                    "#000000", "#E1BD6D", "#EABE94", "#F8AFA8", "#F98400", "#F1BB7B",
                    "#81A88D", "#0B775E", "#35274A", "#FDDDA0", "#f0f0f0", "#377eb8",
                    "#e41a1c","#899DA4", "#D67236")

plotTree(tree,
         heatmapData="NWMA_metadata_numerated_revised.csv",
         heatmap.colours=heatmap_colors,
         legend=T,
         tip.labels = FALSE,
         #tipColours=, tip.colour.cex=.8, legend.pos="bottomright",
         lwd=.85, treeWidth=10, dataWidth=3
         #infoCols="baps", infoCex=0
)
```

Fig. Suppl 2: Mass US Data

```
dfMA <- read_csv("data_MassUS.csv")

vaccineMA <- dfMA %>% distinct(SC, PCV7.actual)
vaccineMA <- vaccineMA %>% subset(PCV7.actual == "VT") %>%
  dplyr::rename(W = PCV7.actual) %>%
  full_join(subset(vaccineMA,PCV7.actual == "NVT")) %>%
  unite("vaccine", W:PCV7.actual, na.rm = TRUE)

dfE1MA <- dfMA %>% subset(Epoch1 == "E1") ## E1 prevaccine

dfFMA <- dfMA %>% select(SC, Epoch1) %>% group_by(Epoch1) %>%
  count(SC) %>% mutate(freq = prop.table(n)) %>% ungroup() %>%
  select(Epoch1, SC, freq) %>%
  spread(Epoch1, freq, fill = 0)


zero_E1MA <- dfE1MA %>% distinct(SC) %>%
  mutate(n = 0, freq = 0)

#### Replicates  - null expectation Pro rata ####
dfE1_NVT_allMA <- data.frame(NULL)
replicates <- 10000

for(i in 1:replicates){
  #sub-sampling from each epoch independently with replacemnt - best subsampling strategy
  dfE1_NVTMA_i <- dfE1MA %>% sample_frac(1, replace = TRUE) %>%
    subset(PCV7.actual == "NVT") %>% count(SC) %>%
    mutate(freq = prop.table(n)) %>% bind_rows(zero_E1MA) %>%
    group_by(SC) %>% summarise(n = sum(n), freq = sum(freq)) %>%
    arrange(SC) %>% ungroup %>% mutate(iter = i)

  dfE1_NVT_allMA <- bind_rows(dfE1_NVT_allMA, dfE1_NVTMA_i)
}

dfE1_NVT_allMA <- dfE1_NVT_allMA %>% group_by(SC) %>%
  summarise(expected_E3 = quantile(freq, 0.5), cil = quantile(freq, 0.025),
            ciu = quantile(freq, 0.975)) %>% ungroup

dfFMA <- left_join(dfFMA,dfE1_NVT_allMA) %>%
  replace(., is.na(.), 0) %>%
  mutate(delta = E3 - E1, deltaExp = expected_E3-E1,
         CI_low = cil-E1, CI_up = ciu-E1) %>%
  mutate(expectation = "Null exp.",
         signif = ifelse(delta > CI_up, "pos",NA)) %>%
  mutate(signif = ifelse(delta < CI_low, "neg", signif)) %>%
  mutate(signif = ifelse(E1 == 0 | E3 == 0, NA, signif)) %>%
  as_tibble(dfFMA) %>% mutate(SC = as.double(SC))

dfFMA <- left_join(as_tibble(dfFMA),vaccineMA)

#### Plot A: Prevalence by sequence cluster ####
dat1MA <- mutate(dfFMA, signif = ifelse(E3 > CI_up | E3 < CI_low, -0.005,NA), Epoch = "E1")
```

```r
dat1MA <- subset(dat1MA, select = c(SC, Epoch, signif))

datPlotAMA <- dfFMA %>% select(SC, E1, E3) %>%
  pivot_longer(-SC, names_to =  "Epoch", values_to = "Prevalence") %>%
  left_join(dat1MA) %>% left_join(select(dfFMA, SC, vaccine)) %>%
  mutate(SC = as.character(SC)) %>% arrange(Epoch,-Prevalence)

rank <- as.character(unique(datPlotAMA$SC))

datPlotAMA <- datPlotAMA %>% mutate(SC = factor(SC, levels = rank))
dfFMA <- dfFMA %>% mutate(SC = factor(SC, levels = rank))

figureSA <- ggplot(datPlotAMA, aes(x=SC, y=Prevalence, alpha=Epoch, fill=vaccine)) +
  geom_bar(stat='identity', position='dodge') +
  scale_alpha_manual(values = c(1,0.4),
                     labels = c("Pre-vaccine", "Post-vaccine")) +
  scale_fill_manual(values = c("NVT"="#143c77","VT"="darkred",
                               "VT_NVT"="mediumpurple4"),
                    labels = c("NVT"="Non-vaccine type",
                               "VT"="Vaccine type", "VT_NVT"="Mixed"),
                    name = "Composition") +
  xlab("Strain (SC)") + theme_classic() +
  scale_y_continuous(expand = c(0, 0), limits = c(0,0.3)) +
  theme(legend.title = element_text(face="bold", size = 10),
        legend.justification = c(1, 1), legend.box = "horizontal",
        legend.position = c(1, 1), # legend.position = c(0.725, 0.85),
        legend.spacing.y = unit(0.2, "cm"),
        legend.text=element_text(size=9),
        legend.background = element_blank(),
        legend.box.background = element_rect(fill = gray(0.96), color = NA))


#### Plot B: Change in prevalence ####
figureSB <- ggplot() + ylim(c(-0.2,0.35)) +
  geom_hline(yintercept=0, lty="dashed",size=0.65) +
  scale_colour_manual(values = c("NVT"="#143c77","VT"="darkred",
                                 "VT_NVT"="mediumpurple4"),
                      labels = c("NVT"="Non-vaccine type",
                                 "VT"="Vaccine type", "VT_NVT"="Mixed"),
                      name = "SC Composition") +
  scale_linetype_manual(values = c(1), name = NULL) +
  scale_shape_manual(values = c("-", "+")) + theme_classic() +
  geom_point(data=dfFMA, aes(SC, delta, col = vaccine),
             size =4, alpha  = 0.85, show.legend = F) +
  geom_pointrange(data=dfFMA, aes(x=SC, y=deltaExp,
                                  ymin=CI_low, ymax=CI_up, lty=expectation),
                  size=.15, fatten = 6, show.legend = F) +
  geom_point(data=dfFMA, aes(x=SC, y=-0.2, shape = signif), size=6,
             col = "lightsteelblue4", show.legend = F) +
  labs(x="Strain (SC)") + labs(y="Change in prevalence") +
  theme(legend.title = element_text(face="bold", size = 10),
        legend.text=element_text(size=9),
        legend.position = c(0.01, 1),
```

```
        legend.justification = c(0.01, 1),
        legend.box = "horizontal",
        legend.background = element_blank(),
        legend.box.background = element_rect(fill = gray(0.95), color = NA))

######################

dfFVTMA <- dfMA %>% select(SC,PCV7.actual, Epoch1) %>%
  group_by(Epoch1) %>%  count(SC,PCV7.actual) %>%
  mutate(freq = round(prop.table(n), digits = 3)) %>%
  ungroup() %>% select(Epoch1, SC, PCV7.actual, freq) %>%
  spread(Epoch1, freq, fill = 0) %>% arrange(SC,PCV7.actual)

#### Present at E1 (17 SCs)####
SCE1MA <- dfFVTMA %>% subset(E1 > 0) %>% ##
  select(SC, PCV7.actual) %>%
  mutate(Epoch1 = "E1")

#### NVT present at E1 (9 SCs)####
SCE2MA <- SCE1MA %>% subset(PCV7.actual == "NVT")


SC_freq_dfMA <- dfMA %>% select(SC,PCV7.actual,Epoch1,pbp1a:CLS343169) %>%
  arrange(SC) %>% group_by(SC,PCV7.actual,Epoch1) %>%
  mutate(SC_n = n()) %>% ungroup() %>%
  group_by(SC,PCV7.actual,Epoch1,SC_n) %>%
  summarise_at(vars(pbp1a:CLS343169),mean) %>%
  ungroup()

### Get the matrix and the SC for the pre-vaccine epch "E1"
df_preVMA <- SCE1MA %>% left_join(SC_freq_dfMA)
SC_freq_preVMA <- as.matrix(df_preVMA %>%
                              mutate(SC_freq=SC_n/sum(SC_n)) %>% select(SC_freq))
SC_COG_preVMA <- as.matrix(t(df_preVMA %>% select(pbp1a:CLS343169)))

## Get e_l for the Mass data (1056 COGs)
el_MA <- SC_COG_preVMA %*% SC_freq_preVMA

#################################

df_postVMA <- SCE2MA %>% left_join(SC_freq_dfMA)
SC_COG_postVMA <- as.matrix(t(df_postVMA %>% select(pbp1a:CLS343169)))

#### Predict postV frequencies ####
SC_freq_postV_predMA <- QP(SC_COG_postVMA, el_MA) #Matrix: rows = COGs, columns = (SCs - VT)

SC_freq_postV_obsMA <- SCE2MA %>% mutate(Epoch1 = "E3") %>%
  left_join(SC_freq_dfMA) %>%
  mutate(SC_freq=SC_n/sum(SC_n, na.rm = T)) %>%
  select(SC, PCV7.actual, SC_freq) %>%
  mutate(SC_freq = replace_na(SC_freq, 0))

SC_freq_postV_obsMA <- SC_freq_postV_obsMA %>%
```

```
  mutate(SC_pred = SC_freq_postV_predMA) %>%
  left_join(vaccineMA)

W_modelSB <- lm(data=SC_freq_postV_obsMA,SC_freq~SC_pred);
summary(W_modelSB)
```

Call: lm(formula = SC_freq ~ SC_pred, data = SC_freq_postV_obsMA)

Residuals: Min 1Q Median 3Q Max -0.073407 -0.018085 -0.008067 0.027381 0.068476

Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.02328 0.02959 0.787 0.4572
SC_pred 0.79044 0.22626 3.494 0.0101 * — Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ’’ 0.05 ‘.’ 0.1 ’ ’ 1

Residual standard error: 0.04684 on 7 degrees of freedom Multiple R-squared: 0.6355, Adjusted R-squared: 0.5834 F-statistic: 12.2 on 1 and 7 DF, p-value: 0.01008

```
confint(W_modelSB)
```

```
               2.5 %     97.5 %
```

(Intercept) -0.04669562 0.09326407 SC_pred 0.25542617 1.32545776

```
linearHypothesis(W_modelSB, c("(Intercept) = 0", "SC_pred = 1"), test = "Chisq")
```

Linear hypothesis test

Hypothesis: (Intercept) = 0 SC_pred = 1

Model 1: restricted model Model 2: SC_freq ~ SC_pred

Res.Df RSS Df Sum of Sq Chisq Pr(>Chisq) 1 9 0.017243
2 7 0.015361 2 0.0018824 0.8578 0.6512

```
## no outliers
outlierSB <- SC_freq_postV_obsMA %>% mutate(diff = abs(SC_freq - SC_pred))
outlierSB <- outlierSB %>%
  filter(diff %in% boxplot(outlierSB$diff, plot = FALSE)$out)

#### Figure C ####
plotSC <- ggplot(SC_freq_postV_obsMA,
                 aes(x = SC_pred, y = SC_freq, colour = vaccine)) +
  theme_classic() +
  geom_segment(aes(x=0,xend=0.3,y=0,yend=0.3),
               color="black",alpha=.7,lwd=0.5,lty=3) +
  geom_smooth(method='lm',color="#899DA4",
              formula=y~x, alpha=0.3, lwd=.6,
              fullrange=T, linetype="blank", show.legend=F) +
  annotate(geom = "text", x=0.28, y =0.29,
           label = "1:1 line", angle = 45, size = 3) +
  geom_point(size=3) +

  scale_x_continuous("Predicted Prevalence (NFDS)")+
  scale_y_continuous("Observed Prevalence") +
  coord_fixed(ratio = 1, ylim=c(0,0.3), xlim=c(0,0.3)) +
  scale_colour_manual(values = c("#143c77","mediumorchid4")) +
  theme(legend.position = "none",
        axis.text = element_text(colour = "black"),)
```

```
###################################

SC_freq_E1MA <- df_preVMA %>%
  mutate(SC_freq_E1=SC_n/sum(SC_n)) %>%
  select(SC,PCV7.actual,SC_freq_E1)

SC_freq_postV_diffMA <- SC_freq_postV_obsMA %>%
  left_join(SC_freq_E1MA) %>%
  mutate(diff_pred = SC_freq_E1 - SC_pred,
         diff_obs = SC_freq_E1 - SC_freq) %>%
  left_join(vaccineMA)

SC_freq_postV_diffMA <- SC_freq_postV_diffMA %>%
  mutate(diff = abs(diff_pred - diff_obs)) %>%
  arrange(diff)

stats <- summary(lm(SC_freq_postV_diffMA$diff_pred~SC_freq_postV_diffMA$diff_obs))
ars <- round(stats$adj.r.squared, digits = 3)
sseE <- round(sse(SC_freq_postV_diffMA$diff_obs,
                  SC_freq_postV_diffMA$diff_pred), digits = 3)
maeE <- round(mae(SC_freq_postV_diffMA$diff_obs,
                  SC_freq_postV_diffMA$diff_pred),digits = 3)##Mean Absolute Error
rmseE <- round(rmse(SC_freq_postV_diffMA$diff_obs,
                    SC_freq_postV_diffMA$diff_pred),digits = 3) ##Root Mean Squared Error

W_modelSC <- lm(data=SC_freq_postV_diffMA,diff_obs~diff_pred);
summary(W_modelSC)
```

Call: lm(formula = diff_obs ~ diff_pred, data = SC_freq_postV_diffMA)

Residuals: Min 1Q Median 3Q Max -0.065605 -0.017260 -0.003470 0.006301 0.071991

Coefficients: Estimate Std. Error t value Pr(>|t|) (Intercept) -0.02685 0.02422 -1.108 0.304 diff_pred 0.54087 0.33000 1.639 0.145

Residual standard error: 0.04393 on 7 degrees of freedom Multiple R-squared: 0.2773, Adjusted R-squared: 0.1741 F-statistic: 2.686 on 1 and 7 DF, p-value: 0.1452

```
confint(W_modelSC)
```

```
               2.5 %     97.5 %
```

(Intercept) -0.08413255 0.03043303 diff_pred -0.23946801 1.32120619

```
linearHypothesis(W_modelSC, c("(Intercept) = 0", "diff_pred = 1"), test = "Chisq")
```

Linear hypothesis test

Hypothesis: (Intercept) = 0 diff_pred = 1

Model 1: restricted model Model 2: diff_obs ~ diff_pred

Res.Df RSS Df Sum of Sq Chisq Pr(>Chisq) 1 9 0.017243
2 7 0.013508 2 0.0037353 1.9357 0.3799

```
#### No outlier ####
outlier3C <- SC_freq_postV_diffMA %>%
  filter(diff %in% boxplot(SC_freq_postV_diffMA$diff, plot = FALSE)$out)
```

```
#### Figure D ####
plotSD <- ggplot(SC_freq_postV_diffMA,
                 aes(x = diff_pred, y = diff_obs, colour = vaccine)) +
  geom_segment(aes(x=-0.2,xend=0.025,y=-0.2,yend=0.025),
               color="black",alpha=.7,lwd=0.5,lty=3) +
  geom_smooth(method='lm', color="gray80" ,formula=y~x,
              alpha=0.3, lwd=.6, fullrange=T,
              linetype="blank", show.legend=F) +
  annotate(geom = "text", x=0.01, y =0.0175,
           label = "1:1 line", angle = 45, size = 3) +
  geom_point(size=3) + theme_classic() +
  scale_colour_manual(values = c("#143c77","mediumorchid4"),
                      labels = c("Non-vaccine type", "Mixed"),
                      name = "SC Composition") +
  theme(legend.position = "none",
        axis.text = element_text(colour = "black"),) +
  xlab("Predicted Prevalence Change (NFDS)")+
  ylab("Observed Prevalence Change") +
  coord_fixed(ratio = 1, xlim=c(-0.2,0.025), ylim=c(-0.2,0.025)) +
  annotate("text",x=-0.19,y=0.01, size=2.5,hjust = 0,
           label=paste("SSE = ", sseE, "\nRMSE = ",
                       rmseE, "\nAdj. R2 = ", ars))

#### Figure E ####
SC_freq_postV_diffMA <- SC_freq_postV_diffMA %>%
  mutate(prorata = SC_freq_E1/sum(SC_freq_E1)) %>%
  mutate(diff_predPro = SC_freq_E1 - prorata) %>%
  mutate(diffP = abs(diff_predPro - diff_obs))

stats <- summary(lm(SC_freq_postV_diffMA$diff_predPro~SC_freq_postV_diffMA$diff_obs))
ars <- round(stats$adj.r.squared, digits = 3)
sseE <- round(sse(SC_freq_postV_diffMA$diff_obs,
                  SC_freq_postV_diffMA$diff_predPro), digits = 3)
maeE <- round(mae(SC_freq_postV_diffMA$diff_obs,
                  SC_freq_postV_diffMA$diff_predPro),digits = 3)##Mean Absolute Error
rmseE <- round(rmse(SC_freq_postV_diffMA$diff_obs,
                    SC_freq_postV_diffMA$diff_predPro),digits = 3) ##Root Mean Squared Error

W_modelSD <- lm(data=SC_freq_postV_diffMA,diff_obs~diff_predPro);
summary(W_modelSD)
```

Call: lm(formula = diff_obs ~ diff_predPro, data = SC_freq_postV_diffMA)

Residuals: Min 1Q Median 3Q Max -0.05935 -0.04736 0.00284 0.01786 0.07442

Coefficients: Estimate Std. Error t value Pr(>|t|) (Intercept) -0.04741 0.02675 -1.772 0.12 diff_predPro 0.18930 0.35483 0.533 0.61

Residual standard error: 0.05065 on 7 degrees of freedom Multiple R-squared: 0.03907, Adjusted R-squared: -0.0982 F-statistic: 0.2846 on 1 and 7 DF, p-value: 0.6102

```
confint(W_modelSD)
```

```
               2.5 %     97.5 %
```

(Intercept) -0.1106683 0.01584975 diff_predPro -0.6497446 1.02834770

```
linearHypothesis(W_modelSD, c("(Intercept) = 0", "diff_predPro = 1"), test = "Chisq")
```

Linear hypothesis test

Hypothesis: (Intercept) = 0 diff_predPro = 1

Model 1: restricted model Model 2: diff_obs ~ diff_predPro

Res.Df RSS Df Sum of Sq Chisq Pr(>Chisq)
1 9 0.031356
2 7 0.017961 2 0.013394 5.22 0.07353 . — Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ’’ 0.05 ‘.’ 0.1 ’ ’ 1

```
plotSE <- ggplot(SC_freq_postV_diffMA, aes(x = diff_predPro,
                                           y = diff_obs, colour = vaccine)) +
  geom_segment(aes(x=-0.2,xend=0.025,y=-0.2,yend=0.025),
               color="black",alpha=.7,lwd=0.5,lty=3) +
  geom_smooth(method='lm', color="gray80" ,formula=y~x,
              alpha=0.3, lwd=.6, fullrange=T,
              linetype="blank", show.legend=F) +
  annotate(geom = "text", x=0.01, y =0.0175,
           label = "1:1 line", angle = 45, size = 3) +
  geom_point(size=3) + theme_classic() +
  scale_colour_manual(values = c("#143c77","mediumorchid4"),
                      labels = c("Non-vaccine type", "Mixed"),
                      name = "SC Composition") +
  theme(legend.position = "none",
        axis.text = element_text(colour = "black"),) +
  xlab("Predicted Prevalence Change (Pro rata)")+
  ylab("Observed Prevalence Change") +
  coord_fixed(ratio = 1, xlim=c(-0.2,0.025), ylim=c(-0.2,0.025)) +
  annotate("text",x=-0.19,y=0.01, size=2.5,hjust = 0,
           label=paste("SSE = ", sseE, "\nRMSE = ",
                       rmseE, "\nAdj. R2 = ", ars))



figureSAB <- plot_grid(figureSA, figureSB, labels = c("A","B"), nrow = 1)
figureSCE <- plot_grid(plotSC, plotSD, plotSE,
                       nrow = 1, labels=c("C","D","E"))

plotSPARC <- plot_grid(figureSAB,figureSCE, ncol=1)
ggsave("figureS_MASS.png", plotSPARC, width = 14, height = 7)
```

Fig. Suppl 3: Similarity of accessory genes among strains

```
#Accessory genome phylogeny
AG.tree <- read.tree("RAxML_bestTree.All.Binary.tre") #All

#Creating distance matrix from Tee
PatristicDistMatrix <- cophenetic(AG.tree) # patristic distances
PatristicDist <- as.dist(PatristicDistMatrix, diag = TRUE, upper = TRUE)

#Setting up group/SC assignments for between SC patristic distance
seq.labels <- as.data.frame(rownames(PatristicDistMatrix)) #Obtaining ordered taxa
```

```
colnames(seq.labels) <- "taxa"
clades <- as.data.frame(cbind(df$FinalName,df$BAPS2)) #All data set
colnames(clades) <- c("taxa","clade")
seq.labels$id  <- 1:nrow(clades) #adding row number to maintain order for sorting after merge
labels.clades <- merge(seq.labels, clades, by="taxa") #merging
labels.clades <- labels.clades[order(labels.clades$id), ] #ordering

#Creating Final Matrix
md <- meandist(PatristicDist, labels.clades$clade) #calculating mean distance between clades
md.matrix <- as.dist(md,diag = FALSE, upper = TRUE)
md.matrix <- as.matrix(md.matrix)
diag(md.matrix) <- NA

#Saving Accessory geneome distances in long format - needed for Sup Figure 4
PatristicDist.long <- as.data.frame(as.table(md.matrix))
PatristicDist.long <- PatristicDist.long[! (PatristicDist.long$Var1 == PatristicDist.long$Var2),]
PatristicDist.long <- PatristicDist.long[!is.na(PatristicDist.long$Freq),]
AccGenomeDistances <- PatristicDist.long #Sup figure 4

#Heatmap of between strain accessory genome distances
my_palette <- colorRampPalette(c("#4d004b", "#8c96c6", "#e0ecf4"))(n = 100)

pdf('figureS_Heatmap.pdf', width = 10, height = 10) #Change as needed
gplots::heatmap.2(md.matrix,
                  col = my_palette,
                  scale = "none",
                  na.rm = TRUE,
                  dendrogram = "row",
                  trace="none",
                  cexCol = .7, cexRow=.7,
                  labRow = rownames(md),
                  srtCol = 70, #Changes angle of X-axis
                  key=TRUE,
                  key.title=NA,
                  key.xlab = "Patristic \nDistance"
)
dev.off()
```

```
## quartz_off_screen
##                 2
```

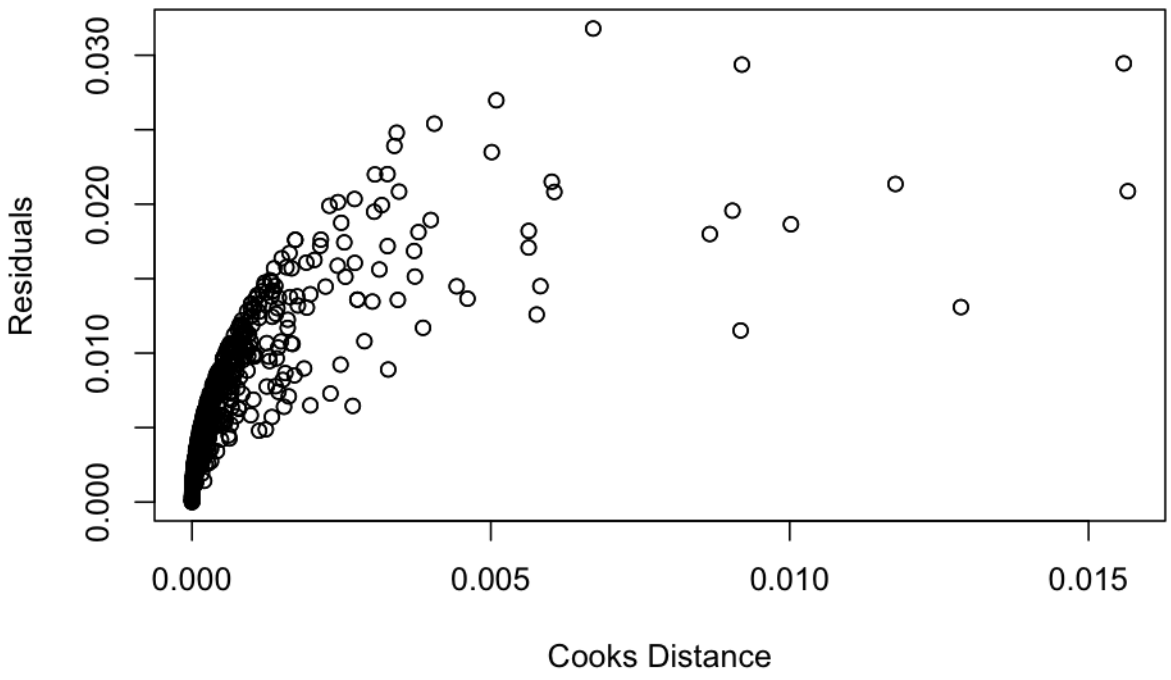
Fig. Suppl 4: Core versus Accessory Genome distances

```
##Scatterplot comparing core genome patristic
## distance and accessory genome distance

#### Figure A ####
#Core genome distances
CG.tree<-read.tree("RAxML_bestTree.All.Core.tre") #Core genome tree

PatristicDistMatrix <- cophenetic(CG.tree) #patristic distances
PatristicDist <- as.dist(PatristicDistMatrix,diag = TRUE, upper = TRUE)
```

```r
#Setting up group/SC assignments for between SC patristic distance
seq.labels <- as.data.frame(rownames(PatristicDistMatrix)) #Obtaining ordered taxa
colnames(seq.labels) <- "taxa"
clades <- as.data.frame(cbind(df$FinalName,df$BAPS2)) #All data set
colnames(clades) <- c("taxa","clade")
seq.labels$id  <- 1:nrow(clades) #adding row number to maintain order for sorting after merge
labels.clades <- merge(seq.labels, clades, by="taxa") #merging
labels.clades <- labels.clades[order(labels.clades$id), ] #ordering

md <- meandist(PatristicDist, labels.clades$clade) #calculating mean distance between clades
md.matrix <- as.dist(md,diag = FALSE, upper = TRUE)
md.matrix <- as.matrix(md.matrix)
diag(md.matrix) <- NA

#Accessory geneome distances
PatristicDist.long <- as.data.frame(as.table(md.matrix))
PatristicDist.long <- PatristicDist.long[! (PatristicDist.long$Var1 == PatristicDist.long$Var2),]
PatristicDist.long <- PatristicDist.long[!is.na(PatristicDist.long$Freq),]
CoreGenomeDistances <- PatristicDist.long

MergedDistances <- as.data.frame(cbind(CoreGenomeDistances,AccGenomeDistances))
colnames(MergedDistances) <- c("Var1a", "Var2b", "PCore", "Var1c", "Var2d","PAcc")

MergedDistances$MSE  <- (MergedDistances$PCore-MergedDistances$PAcc)^2
MergedDistancesMedian <- MergedDistances[(MergedDistances$PCore > 0.06 & MergedDistances$PCore < 0.15),]
MergedDistancesMedian <- within(MergedDistancesMedian, A.quartile <- as.integer(cut(MergedDistancesMedia
MergedDistancesMedian <- within(MergedDistancesMedian, P.quartile <- as.integer(cut(MergedDistancesMedia

model <- lm(MergedDistancesMedian$PCore~MergedDistancesMedian$PAcc)
car::outlierTest(model)
```

```
## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##      rstudent unadjusted p-value Bonferroni p
## 207 -3.554855         0.00039412      0.43826
```

```r
MergedDistancesMedian$residuals <- abs(resid(model)) #Risiduals
MergedDistancesMedian$cooks <- cooks.distance(model)
plot(MergedDistancesMedian$cooks, MergedDistancesMedian$residuals,
     xlab="Cooks Distance", ylab="Residuals")
```




```
#Make name varianble
MergedDistancesMedian$comp <- paste(MergedDistancesMedian$Var1a,
                                    MergedDistancesMedian$Var2b, sep = "-")

R2 <- round(cor(MergedDistancesMedian$PCore,
                MergedDistancesMedian$PAcc,method = "pearson"),2)

plotS4A <- ggplot(MergedDistancesMedian, aes(PCore,PAcc)) +
  geom_point(color="black",size=1) + theme_classic() +
  geom_density_2d(color="#08519c", alpha=.7, show.legend =FALSE) +
  geom_smooth(color="#636363", method=lm, alpha=.3, linetype="dashed",
              size=.6, formula = y~x,  show.legend=FALSE, se=TRUE) +
  labs(x="Core Genome Divergence (Patristic Distance)") +
  labs(y="Accessory Genome Divergence \n(Patristic Distance)") +
  annotate("text", x = .075, y = 2.3,
           label = paste("R^2 == ", R2), parse=TRUE) +
  theme(axis.text = element_text(colour = "black"))

#### Figure B ####

RelativeFitness.E1 <- dat3AB %>% filter(BAPS2!="01")

RF.E1.distmat <- as.matrix(dist(RelativeFitness.E1$r))
RF.E1.distmat[upper.tri(RF.E1.distmat)] <- NA; diag(RF.E1.distmat) <- NA
rownames(RF.E1.distmat) <- RelativeFitness.E1$BAPS2
colnames(RF.E1.distmat) <- RelativeFitness.E1$BAPS2
```

```
RF.E1.distlong <- na.omit(as.data.frame.table(RF.E1.distmat))
colnames(RF.E1.distlong) <- c("Var1a","Var2b","RelFit")

CorDist_RelFit <- merge(RF.E1.distlong, MergedDistances, by=c("Var1a", "Var2b"))
CorDist_RelFit <- CorDist_RelFit[,c("Var1a", "Var2b","RelFit","PCore","PAcc")]

####Supplemental figures - Core and accessory divergence including fitness
plotS4B <- ggplot(CorDist_RelFit, aes(PCore,RelFit)) +
  geom_point(color="black",size=1) + theme_classic() +
  geom_density_2d(color="#08519c", alpha=.7, show.legend =FALSE) +
  labs(x="Core Genome Divergence (Patristic Distance)") +
  labs(y="Absolute fitness difference") +
  theme(axis.text = element_text(colour = "black"))

plotS4C <- ggplot(CorDist_RelFit, aes(PAcc,RelFit)) +
  geom_point(color="black",size=1) + theme_classic() +
  geom_density_2d(color="#08519c", alpha=.7, show.legend =FALSE) +
  labs(x="Accessory Genome Divergence (Patristic Distance)") +
  labs(y="Absolute fitness difference") +
  theme(axis.text = element_text(colour = "black"))


#theme_set(theme_cowplot(font_size=13))
figureS4 <- plot_grid(ncol=3, nrow=1, plotS4A, plotS4B,
                      plotS4C, labels = "AUTO", align = 'h')
ggsave("figureS_distance.png", figureS4, width = 15, height = 5)
```

Fig. Suppl 5: COGs by SCs

```
SC_freq_E1 <- SC_freq_postV_obs %>% select(BAPS2, PCV7.actual) %>%
  left_join(df) %>% subset(Epoch1 == "E1") %>%
  select(BAPS2, HMPREF0837_12128:HMPREF0837_10616) %>%
  arrange(BAPS2) %>% group_by(BAPS2) %>%
  summarise_at(vars(HMPREF0837_12128:HMPREF0837_10616),mean) %>%
  ungroup() %>%
  pivot_longer(cols=HMPREF0837_12128:HMPREF0837_10616,
               names_to = "COG", values_to = "f") %>%
  mutate(Epoch = "Pre-vaccine")

SC_freq_E3 <- SC_freq_postV_obs %>% select(BAPS2, PCV7.actual) %>%
  left_join(df) %>% subset(Epoch1 == "E3") %>%
  select(BAPS2, HMPREF0837_12128:HMPREF0837_10616) %>%
  arrange(BAPS2) %>% group_by(BAPS2) %>%
  summarise_at(vars(HMPREF0837_12128:HMPREF0837_10616),mean) %>%
  ungroup() %>%
  pivot_longer(cols=HMPREF0837_12128:HMPREF0837_10616,
               names_to = "COG", values_to = "f") %>%
  mutate(Epoch = "Post-vaccine")

SC_freq_E13 <- bind_rows(SC_freq_E1, SC_freq_E3)
```

```
plotCOGs_SC <- ggplot(SC_freq_E13, aes(f, fill = Epoch)) +
  geom_histogram(aes(y = ..count..), position = "dodge", bins = 10) +
  facet_wrap(~BAPS2, nrow = 3) +
  xlab("COG frequency") + theme_minimal() +
  scale_fill_manual("", values = c("#D3723D", "#8A9DA4")) +
  annotate("segment", x=-Inf, xend=Inf, y=-Inf, yend=-Inf)+
  annotate("segment", x=-Inf, xend=-Inf, y=-Inf, yend=Inf) +
  theme(legend.position = "bottom")

ggsave("figureS_COGsSCs.png", plotCOGs_SC, width = 15, height = 5)
```

Table Statistics

```
#### Accessory genome SA ####
## Sensitivity analysis using a subsample of 119 isolates
## collected in 2010 prior to the initiation of PCV13

#### Data ####
dfSenst <- df %>% separate(FinalName, c("ID1", "ID2", "ID3", "Year"))
dfSenstE12 <- dfSenst %>% filter(Epoch1 != "E3")
dfSenstE3 <- dfSenst %>% filter(Epoch1 == "E3" & Year == "2010")
dfS <- bind_rows(dfSenstE12,dfSenstE3)

dfFVTS <- dfS %>% select(BAPS2,PCV7.actual, Epoch1) %>%
  group_by(Epoch1) %>%  count(BAPS2,PCV7.actual) %>%
  mutate(freq = round(prop.table(n), digits = 3)) %>% ungroup() %>%
  select(Epoch1, BAPS2, PCV7.actual, freq) %>%
  spread(Epoch1, freq, fill = 0) %>% arrange(BAPS2,PCV7.actual)

#### Present at E1 ####
SCE1S <- dfFVTS %>% subset(E1 > 0) %>% ## & BAPS2 !="27"
  select(BAPS2, PCV7.actual) %>%
  mutate(Epoch1 = "E1")

#### NVTs Present at E1 ####
SCE2S <- SCE1S %>% subset(PCV7.actual == "NVT"  & BAPS2 !="27")

SC_freq_dfS <- dfS %>% select(BAPS2, PCV7.actual, Epoch1,
                              HMPREF0837_12128:HMPREF0837_10616) %>%
  arrange(BAPS2) %>% group_by(BAPS2,PCV7.actual,Epoch1) %>%
  mutate(SC_n = n()) %>% ungroup() %>%
  group_by(BAPS2,PCV7.actual,Epoch1,SC_n) %>%
  summarise_at(vars(HMPREF0837_12128:HMPREF0837_10616),mean) %>%
  ungroup()

### Get the matrix and the SC for the pre-vaccine epch "E1"
df_preVS <- SCE1S %>% left_join(SC_freq_dfS)
SC_freq_preVS <- as.matrix(df_preVS %>%
                mutate(SC_freq=SC_n/sum(SC_n)) %>% select(SC_freq))
SC_COG_preVS <- as.matrix(t(df_preVS %>%
                select(HMPREF0837_12128:HMPREF0837_10616)))
```

```
#### Get e_l ####
el_S <- SC_COG_preVS %*% SC_freq_preVS

df_postVS <- SCE2S %>% left_join(SC_freq_dfS)
SC_COG_postVS <- as.matrix(t(df_postVS %>%
                 select(HMPREF0837_12128:HMPREF0837_10616)))

SC_freq_postV_obsS <- SCE2S %>% mutate(Epoch1 = "E3") %>%
  left_join(SC_freq_dfS) %>%
  mutate(SC_freq=SC_n/sum(SC_n, na.rm = T)) %>%
  select(BAPS2, PCV7.actual, SC_freq) %>%
  mutate(SC_freq = replace_na(SC_freq, 0))

## Predict postV frequencies
SC_freq_postV_predS <- QP(SC_COG_postVS, el_S) #Matrix: rows = COGs, columns = (SCs - VT)
SC_freq_postV_obsS <- SC_freq_postV_obsS %>%
  mutate(SC_pred = SC_freq_postV_predS)

SC_freq_E1S <- df_preVS %>%
  mutate(SC_freq_E1=SC_n/sum(SC_n)) %>%
  select(BAPS2,PCV7.actual,SC_freq_E1)

SC_freq_postV_diffS <- SC_freq_postV_obsS %>%
  left_join(SC_freq_E1S) %>%
  mutate(diff_pred = SC_freq_E1 - SC_pred,
         diff_obs = SC_freq_E1 - SC_freq)

SC_freq_postV_diffS <- SC_freq_postV_diffS %>%
  mutate(diff = abs(diff_pred - diff_obs)) %>%
  arrange(diff)

stats <- summary(lm(SC_freq_postV_diffS$diff_pred~SC_freq_postV_diffS$diff_obs))
ars <- round(stats$adj.r.squared, digits = 3)
sseE <- round(sse(SC_freq_postV_diffS$diff_obs,
                  SC_freq_postV_diffS$diff_pred), digits = 3)
rmseE <- round(rmse(SC_freq_postV_diffS$diff_obs,
                    SC_freq_postV_diffS$diff_pred),digits = 3)

accNFDSSA <- data.frame(Model = "Accesory genome (NFDS) SA",
                        nloci = length(el_S), adj.r.squared = ars,
                        SSE = sseE, RMSE = rmseE)

SC_freq_postV_diffS <- SC_freq_postV_diffS %>%
  mutate(prorata = SC_freq_E1/sum(SC_freq_E1)) %>%
  mutate(diff_predPro = SC_freq_E1 - prorata) %>%
  mutate(diffP = abs(diff_predPro - diff_obs))

stats <- summary(lm(SC_freq_postV_diffS$diff_predPro~SC_freq_postV_diffS$diff_obs))
ars <- round(stats$adj.r.squared, digits = 3)
sseE <- round(sse(SC_freq_postV_diffS$diff_obs,
                  SC_freq_postV_diffS$diff_predPro), digits = 3)
rmseE <- round(rmse(SC_freq_postV_diffS$diff_obs,
                    SC_freq_postV_diffS$diff_predPro),digits = 3)
```

```
accProrataSA <- data.frame(Model = "Accesory genome (Prorata) SA",
                           nloci = length(el_S), adj.r.squared = ars,
                           SSE = sseE, RMSE = rmseE)

#####################

#### Core genome ####
dfcore <- read.csv("CoreSNPpresenceAbsence.txt", header = F)
dfcore <- dfcore %>% as_tibble() %>% rename(FinalName = V1)
dfcore <- df %>% select(FinalName, BAPS2, PCV7.actual, Epoch1) %>%
  left_join(dfcore)

dfFVTcore <- dfcore %>% select(BAPS2,PCV7.actual, Epoch1) %>%
  group_by(Epoch1) %>%  count(BAPS2,PCV7.actual) %>%
  mutate(freq = round(prop.table(n), digits = 3)) %>% ungroup() %>%
  select(Epoch1, BAPS2, PCV7.actual, freq) %>%
  spread(Epoch1, freq, fill = 0) %>% arrange(BAPS2,PCV7.actual)

#### Present at E1 ####
SCE1core <- dfFVTcore %>% subset(E1 > 0) %>% ## & BAPS2 !="27"
  select(BAPS2, PCV7.actual) %>%
  mutate(Epoch1 = "E1")

#### NVTs Present at E1 ####
SCE2core <- SCE1core %>% subset(PCV7.actual == "NVT"  & BAPS2 !="27")


SC_freq_dfcore <- dfcore %>% select(BAPS2, PCV7.actual, Epoch1,
                                    V2:V62654) %>%
  arrange(BAPS2) %>% group_by(BAPS2,PCV7.actual,Epoch1) %>%
  mutate(SC_n = n()) %>% ungroup() %>%
  group_by(BAPS2,PCV7.actual,Epoch1,SC_n) %>%
  summarise_at(vars(V2:V62654),mean) %>%
  ungroup()

### Get the matrix and the SC for the pre-vaccine epch "E1"
df_preVcore <- SCE1core %>% left_join(SC_freq_dfcore)
SC_freq_preVcore <- as.matrix(df_preVcore %>%
                     mutate(SC_freq=SC_n/sum(SC_n)) %>% select(SC_freq))
SC_COG_preVcore <- as.matrix(t(df_preVcore %>% select(V2:V62654)))

#### Get e_l for the core (62653 loci) ####
el_core <- SC_COG_preVcore %*% SC_freq_preVcore

#### Observed versus prediced prevalence ####
df_postVcore <- SCE2core %>% left_join(SC_freq_dfcore)
SC_COG_postVcore <- as.matrix(t(df_postVcore %>% select(V2:V62654)))

SC_freq_postV_obsCore <- SCE2core %>% mutate(Epoch1 = "E3") %>%
  left_join(SC_freq_dfcore) %>%
  mutate(SC_freq=SC_n/sum(SC_n, na.rm = T)) %>%
  select(BAPS2, PCV7.actual, SC_freq) %>%
  mutate(SC_freq = replace_na(SC_freq, 0))
```

```
## Predict postV frequencies
SC_freq_postV_predCore <- QP(SC_COG_postVcore, el_core)
SC_freq_postV_obsCore <- SC_freq_postV_obsCore %>%
  mutate(SC_pred = SC_freq_postV_predCore)

SC_freq_E1core <- df_preVcore %>%
  mutate(SC_freq_E1=SC_n/sum(SC_n)) %>%
  select(BAPS2,PCV7.actual,SC_freq_E1)

SC_freq_postV_diffcore <- SC_freq_postV_obsCore %>%
  left_join(SC_freq_E1core) %>%
  mutate(diff_pred = SC_freq_E1 - SC_pred,
         diff_obs = SC_freq_E1 - SC_freq)

SC_freq_postV_diffcore <- SC_freq_postV_diffcore %>%
  mutate(diff = abs(diff_pred - diff_obs)) %>%
  arrange(diff)

#### Pro rata ####
SC_freq_postV_diffcore <- SC_freq_postV_diffcore %>%
  mutate(prorata = SC_freq_E1/sum(SC_freq_E1)) %>%
  mutate(diff_predPro = SC_freq_E1 - prorata) %>%
  mutate(diffP = abs(diff_predPro - diff_obs))

#### Stats core NFDS ####
stats <- summary(lm(SC_freq_postV_diffcore$diff_pred~SC_freq_postV_diffcore$diff_obs))
ars <- round(stats$adj.r.squared, digits = 3)
sseE <- round(sse(SC_freq_postV_diffcore$diff_obs,
                  SC_freq_postV_diffcore$diff_pred), digits = 3)
rmseE <- round(rmse(SC_freq_postV_diffcore$diff_obs,
                    SC_freq_postV_diffcore$diff_pred),digits = 3)

coreNFDS <- data.frame(Model = "Core genome (NFDS)", nloci = length(el_core),
                       adj.r.squared = ars, SSE = sseE, RMSE = rmseE)

#### Stats core Pro rata ####
stats <- summary(lm(SC_freq_postV_diffcore$diff_predPro~SC_freq_postV_diffcore$diff_obs))
ars <- round(stats$adj.r.squared, digits = 3)
sseE <- round(sse(SC_freq_postV_diffcore$diff_obs,
                  SC_freq_postV_diffcore$diff_predPro), digits = 3)
rmseE <- round(rmse(SC_freq_postV_diffcore$diff_obs,
                    SC_freq_postV_diffcore$diff_predPro),digits = 3)

coreProrata <- data.frame(Model = "Core genome (Pro rata)", nloci = length(el_core),
                          adj.r.squared = ars, SSE = sseE, RMSE = rmseE)

##############################

#### Metabolic loci ####
dfmeta <- read.csv("Core_Metabolic_SNPpresenceAbsence.txt", header = F)
dfmeta <- dfmeta %>% as_tibble() %>% rename(FinalName = V1)
dfmeta <- df %>% select(FinalName, BAPS2, PCV7.actual, Epoch1) %>%
  left_join(dfmeta)
```

```
dfFVTmeta <- dfmeta %>% select(BAPS2,PCV7.actual, Epoch1) %>%
  group_by(Epoch1) %>%  count(BAPS2,PCV7.actual) %>%
  mutate(freq = round(prop.table(n), digits = 3)) %>% ungroup() %>%
  select(Epoch1, BAPS2, PCV7.actual, freq) %>%
  spread(Epoch1, freq, fill = 0) %>% arrange(BAPS2,PCV7.actual)

#### Present at E1 ####
SCE1meta <- dfFVTmeta %>% subset(E1 > 0) %>% ## & BAPS2 !="27"
  select(BAPS2, PCV7.actual) %>%
  mutate(Epoch1 = "E1")

#### NVTs Present at E1 ####
SCE2meta <- SCE1meta %>% subset(PCV7.actual == "NVT"  & BAPS2 !="27")


SC_freq_dfmeta <- dfmeta %>% select(BAPS2, PCV7.actual, Epoch1,
                                    V2:V22434) %>%
  arrange(BAPS2) %>% group_by(BAPS2,PCV7.actual,Epoch1) %>%
  mutate(SC_n = n()) %>% ungroup() %>%
  group_by(BAPS2,PCV7.actual,Epoch1,SC_n) %>%
  summarise_at(vars(V2:V22434),mean) %>%
  ungroup()

### Get the matrix and the SC for the pre-vaccine epch "E1"
df_preVmeta <- SCE1meta %>% left_join(SC_freq_dfmeta)
SC_freq_preVmeta <- as.matrix(df_preVmeta %>%
                     mutate(SC_freq=SC_n/sum(SC_n)) %>% select(SC_freq))
SC_COG_preVmeta <- as.matrix(t(df_preVmeta %>% select(V2:V22434)))

#### Get e_l for the meta (62653 loci) ####
el_meta <- SC_COG_preVmeta %*% SC_freq_preVmeta

#### Observed versus prediced prevalence ####
df_postVmeta <- SCE2meta %>% left_join(SC_freq_dfmeta)
SC_COG_postVmeta <- as.matrix(t(df_postVmeta %>% select(V2:V22434)))

SC_freq_postV_obsMeta <- SCE2meta %>% mutate(Epoch1 = "E3") %>%
  left_join(SC_freq_dfmeta) %>%
  mutate(SC_freq=SC_n/sum(SC_n, na.rm = T)) %>%
  select(BAPS2, PCV7.actual, SC_freq) %>%
  mutate(SC_freq = replace_na(SC_freq, 0))

## Predict postV frequencies
SC_freq_postV_predMeta <- QP(SC_COG_postVmeta, el_meta)
SC_freq_postV_obsMeta <- SC_freq_postV_obsMeta %>%
  mutate(SC_pred = SC_freq_postV_predMeta)

SC_freq_E1meta <- df_preVmeta %>%
  mutate(SC_freq_E1=SC_n/sum(SC_n)) %>%
  select(BAPS2,PCV7.actual,SC_freq_E1)

SC_freq_postV_diffmeta <- SC_freq_postV_obsMeta %>%
  left_join(SC_freq_E1meta) %>%
```

```
  mutate(diff_pred = SC_freq_E1 - SC_pred,
         diff_obs = SC_freq_E1 - SC_freq)

SC_freq_postV_diffmeta <- SC_freq_postV_diffmeta %>%
  mutate(diff = abs(diff_pred - diff_obs)) %>%
  arrange(diff)

#### Pro rata ####
SC_freq_postV_diffmeta <- SC_freq_postV_diffmeta %>%
  mutate(prorata = SC_freq_E1/sum(SC_freq_E1)) %>%
  mutate(diff_predPro = SC_freq_E1 - prorata) %>%
  mutate(diffP = abs(diff_predPro - diff_obs))

#### Stats metabolic loci NFDS ####
stats <- summary(lm(SC_freq_postV_diffmeta$diff_pred~SC_freq_postV_diffmeta$diff_obs))
ars <- round(stats$adj.r.squared, digits = 3)
sseE <- round(sse(SC_freq_postV_diffmeta$diff_obs,
                  SC_freq_postV_diffmeta$diff_pred), digits = 3)
rmseE <- round(rmse(SC_freq_postV_diffmeta$diff_obs,
                    SC_freq_postV_diffmeta$diff_pred),digits = 3)

metaNFDS <- data.frame(Model = "Metabolic loci (NFDS)", nloci = length(el_meta),
                       adj.r.squared = ars, SSE = sseE, RMSE = rmseE)

#### Stats metabolic loci Pro rata ####
stats <- summary(lm(SC_freq_postV_diffmeta$diff_predPro~SC_freq_postV_diffmeta$diff_obs))
ars <- round(stats$adj.r.squared, digits = 3)
sseE <- round(sse(SC_freq_postV_diffmeta$diff_obs,
                  SC_freq_postV_diffmeta$diff_predPro), digits = 3)
rmseE <- round(rmse(SC_freq_postV_diffmeta$diff_obs,
                    SC_freq_postV_diffmeta$diff_predPro),digits = 3)

metaProrata <- data.frame(Model = "Metabolic loci (Pro rata)", nloci = length(el_meta),
                          adj.r.squared = ars, SSE = sseE, RMSE = rmseE)

##############################

#### create table ####
dfOut <- bind_rows(accNFDS, accProrata, accNFDSSA,
                   accProrataSA, coreNFDS, coreProrata,
                   metaNFDS, metaProrata)
write.csv(dfOut, "statsTable.csv", row.names = F)
```