

Supplement A: Simulations to Determine whether Average Familial Ancestry Provides Better Estimates of Progeny Ancestry than Direct Genotyping

Background

Accurately estimating admixture can be difficult when using only a few genetic markers, but the use of average familial admixture provides a promising alternative. Recombination in hybrid individuals can produce genomic regions that are not representative of genome-wide admixture patterns. As a result, estimates of genomic ancestry may be incorrect if only a small number of loci are used to produce the estimates. An alternative approach is to use the familial admixture calculated by averaging the ancestry of both parents. By using both parents to estimate progeny admixture, twice as many genes will be used in the estimate. However, the accuracy of this technique may be influenced by the variation in admixture in the progeny produced or the effect of recombination on the variance of progeny admixture. Here, we simulated 10,000 F₂ and backcrossed progeny to identify whether using familial admixture provides better estimates of progeny admixture than genotyping progeny directly. F₂ offspring are expected to have the greatest variance in admixture, as an F₁ × F₁ cross could produce offspring with admixture levels ranging from 0 to 1.

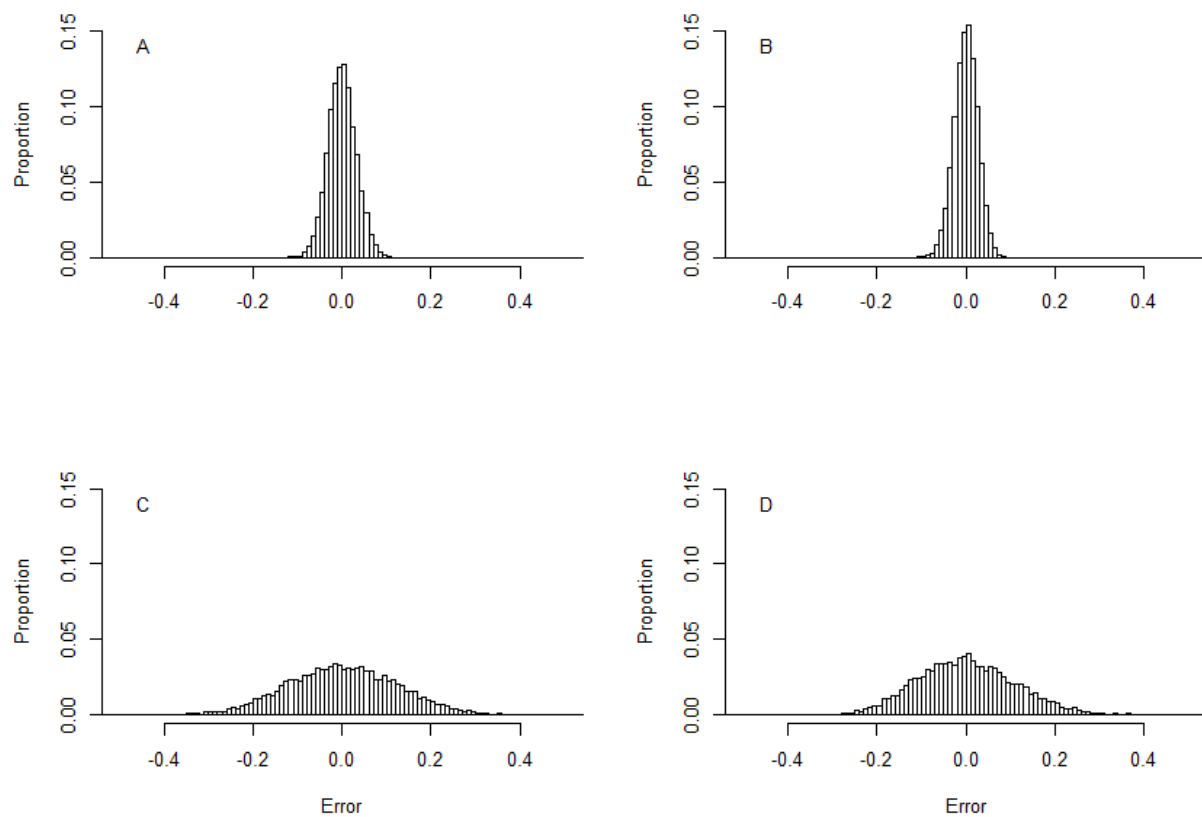
Methods

To simulate progeny, we created pure and F₁ parents with 100 diagnostic loci per chromosome arm. We used a realistic number of chromosome arms for members of the genus *Oncorhynchus*, i.e., 50 pairs (Allendorf and Thorgaard 1984). Recombination was allowed to occur at previously published rates of 0.9, 0.026, and 0.074 for one, two, and zero crossovers per arm (Brieuc et al. 2014). Recombination was allowed to occur randomly across each chromosome arm.

The true admixture was identified for each progeny and compared with two different admixture estimates: (1) the average admixture in the parents estimated using 8 randomly selected SNPs and (2) the admixture estimated from 8 randomly selected SNPs in the progeny itself.

Results

The variation in error (estimate minus true admixture) was smaller for familial admixture estimates than for admixtures estimated directly from the progeny (Figure S.A.1). The standard deviation of error for familial admixture estimates was 0.031 (F_2) and 0.026 (backcross), compared with 0.122 (F_2) and 0.108 (backcross) for admixture estimated directly from progeny.



Supplementary Figure S.A.1. Variation in error rates for different estimates of admixture in simulated progeny. Panel (A) shows the error in estimated admixture of F_2 progeny using the average admixture of both parents (familial admixture) calculated at 8 randomly selected loci, panel (B) the error in estimated admixture of backcrossed progeny using the average admixture of both parents calculated at 8 randomly selected loci, panel (C) the error in estimated admixture of F_2 progeny using the admixture calculated at 8 randomly selected loci in the individual itself,

and panel **(D)** the error in estimated admixture of backcrossed progeny using the admixture calculated at 8 randomly selected loci in the individual itself.

Supplemental References

Allendorf, F. W., and G. H. Thorgaard. 1984. Tetraploidy and the evolution of salmonid fishes. Pages 1–27 in B. J. Turner, editor. Evolutionary genetics of fishes. Plenum, New York.

Brieuc, M. S. O., C. D. Waters, J. E. Seeb, and K. A. Naish. 2014. A dense linkage map for Chinook Salmon (*Oncorhynchus tshawytscha*) reveals variable chromosomal divergence after an ancestral whole genome duplication event. *G3: Genes, Genomes, Genetics* 4:447-460.

R script

```
# recombination
recomb <- function(arm1, arm2){
  new_arm <- c()
  recomb <- runif(1)

  if (recomb > 0.974){ # double recombination
    positions <- sort(round(runif(2)*100))
    new_arm <- c(arm1[1:positions[1]], arm2[positions[1]+1:positions[2]], arm1[positions[1]+1:length(arm1)])
    if (runif(1)>0.5){
      new_arm <- c(arm2[1:positions[1]], arm1[positions[1]+1:positions[2]], arm2[positions[1]+1:length(arm2)])
    }
  }
  else if (recomb > 0.90){ # zero recombination
    new_arm <- arm1
    if (runif(1)>0.5){
      new_arm <- arm2
    }
  }
  else { # single recombination
    position <- round(runif(1)*100)
    new_arm <- c(arm1[1:position], arm2[position+1:length(arm2)])
    if (runif(1)>0.5){
      new_arm <- c(arm2[1:position], arm1[position+1:length(arm1)])
    }
  }
  return(new_arm[1:100])
}

trueAdmixture <- function(genome){
  return(mean(genome))
}

parentalAdmixture <- function(parent1, parent2){
  positions <- round(runif(8, min=1, max=5000))
  positions <- c(positions, positions+5000)
  value <- c()
```

```

for (i in seq(1,16)){
  value <- c(value, parent1[positions[i]])
  value <- c(value, parent2[positions[i]])
}
value <- mean(value)

return(value)
}

progenyAdmixture <- function(genome){
  positions <- round(runif(8, min=1, max=5000))
  positions <- c(positions, positions+5000)
  value <- c()

  for (i in seq(1,16)){
    value <- c(value, genome[positions[i]])
  }
  value2 <- mean(value)

  return(value2)
}

##### F2 #####
# run 1000 F2s to see how well we do at estimating admixture
f1_arm1 <- rep(0,100)
f1_arm2 <- rep(1,100)
genome <- c()
parent1 <- c(rep(0,5000), rep(1,5000))
parent2 <- c(rep(0,5000), rep(1,5000))

parental_true <- c()
true_true <- c()
progeny_true <- c()
parentVsTrue_F2 <- c()
progenyVsTrue_F2 <- c()

for (j in seq(1:10000)){
  if(j%%100==0){print(j)}
  genome <- c()

  # this creates a single progeny
  for (i in seq(1:100)){ # assuming 100 chromosome arms
    genome <- c(genome, recomb(f1_arm1, f1_arm2))
  }

  truth <- trueAdmixture(genome) # calculate true admixture
  parental <- parentalAdmixture(parent1, parent2) # calculate parental admixture
  progeny <- progenyAdmixture(genome)

  parental_true[j] <- parental
  true_true[j] <- truth
  progeny_true[j] <- progeny
  parentVsTrue_F2[j] <- parental-truth
  progenyVsTrue_F2[j] <- progeny-truth
}

```

```
##### BC1 #####
# run 1000 BC1's to see how well we do at estimating admixture
f1_arm1 <- rep(0,100)
f1_arm2 <- rep(1,100)
parent1 <- c(rep(0,5000), rep(1,5000))
parent2 <- rep(0, 10000)
genome <- c()

parentVsTrue_BC1 <- c()
progenyVsTrue_BC1 <- c()

for (j in seq(1:10000)){
  if(j%%100==0){print(j)}
  genome <- c()

  # this creates a progeny
  for (i in seq(1:100)){ # assuming 100 chromosome arms
    f2_arm <- recomb(f1_arm1, f1_arm2)
    genome <- c(genome, recomb(f2_arm, pure_arm))
  }

  truth <- trueAdmixture(genome) # calculate true admixture
  parental <- parentalAdmixture(parent1, parent2) # calculate parental admixture
  progeny <- progenyAdmixture(genome)

  parentVsTrue_BC1[j] <- parental-truth
  progenyVsTrue_BC1[j] <- progeny-truth
}

par(mfrow=c(2,2))
hist(parentVsTrue_F2, xlim=c(-0.5,0.5), breaks=seq(-1, 1, 0.01), main="", xlab="", ylim=c(0,1500), ylab="Proportion", yaxt="n")
axis(2, at=c(0,500,1000,1500), labels=c("0.00", "0.05", "0.10", "0.15"))
legend("topleft", "A", bty="n")
hist(parentVsTrue_BC1, xlim=c(-0.5,0.5), breaks=seq(-1, 1, 0.01), main="", xlab="", ylim=c(0,1500), ylab="Proportion", yaxt="n")
axis(2, at=c(0,500,1000,1500), labels=c("0.00", "0.05", "0.10", "0.15"))
legend("topleft", "B", bty="n")
hist(progenyVsTrue_F2, xlim=c(-0.5,0.5), breaks=seq(-1, 1, 0.01), main="", xlab="Error", ylim=c(0,1500), ylab="Proportion", yaxt="n")
axis(2, at=c(0,500,1000,1500), labels=c("0.00", "0.05", "0.10", "0.15"))
legend("topleft", "C", bty="n")
hist(progenyVsTrue_BC1, xlim=c(-0.5,0.5), breaks=seq(-1, 1, 0.01), main="", xlab="Error", ylim=c(0,1500), ylab="Proportion", yaxt="n")
axis(2, at=c(0,500,1000,1500), labels=c("0.00", "0.05", "0.10", "0.15"))
legend("topleft", "D", bty="n")
```