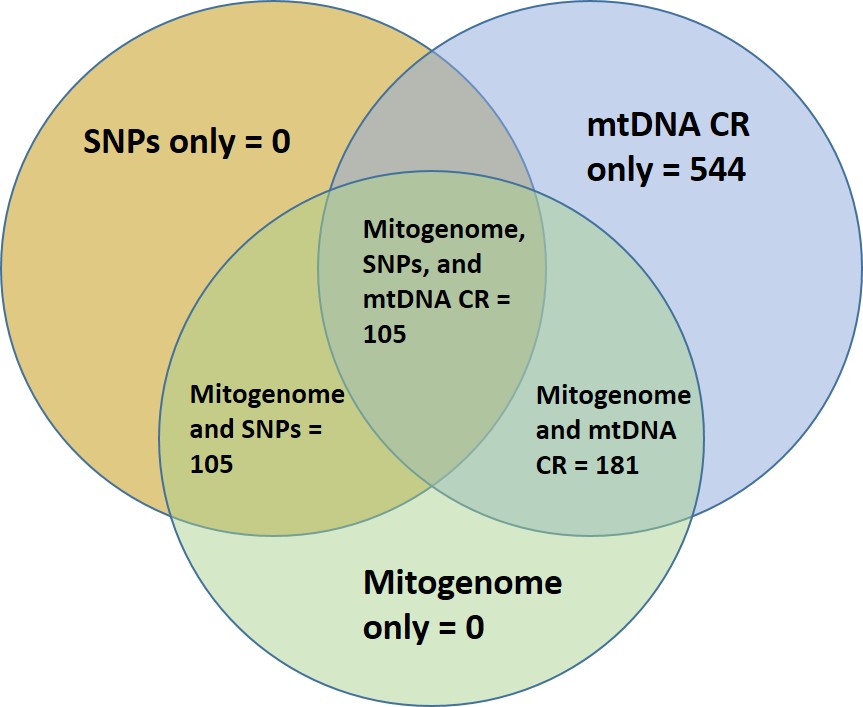
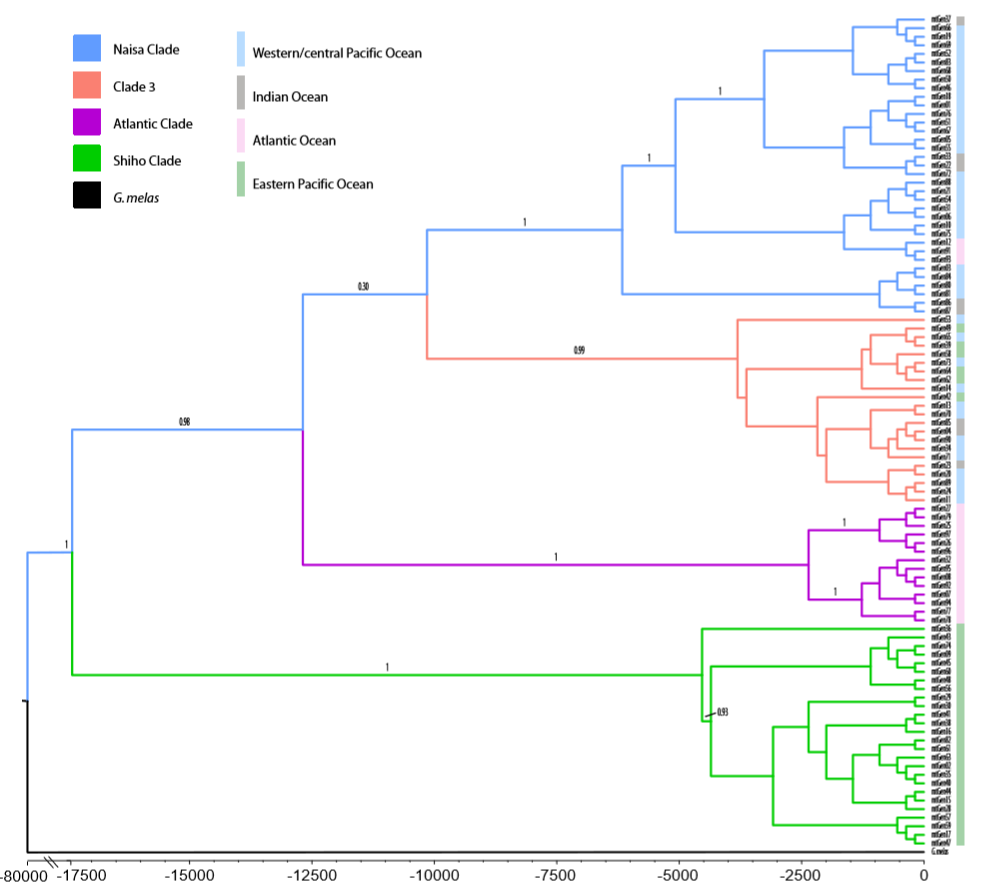
**Supplemental Figures**

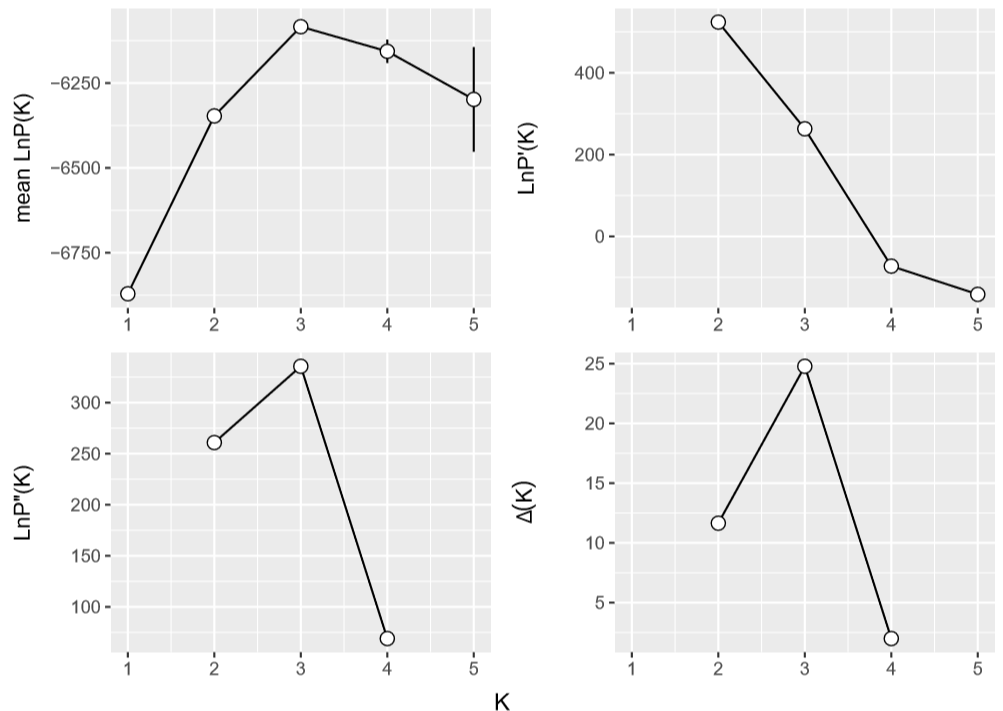
Supplemental Figure S1. Venn Diagram illustrating the overlapping sample sets used in this study.



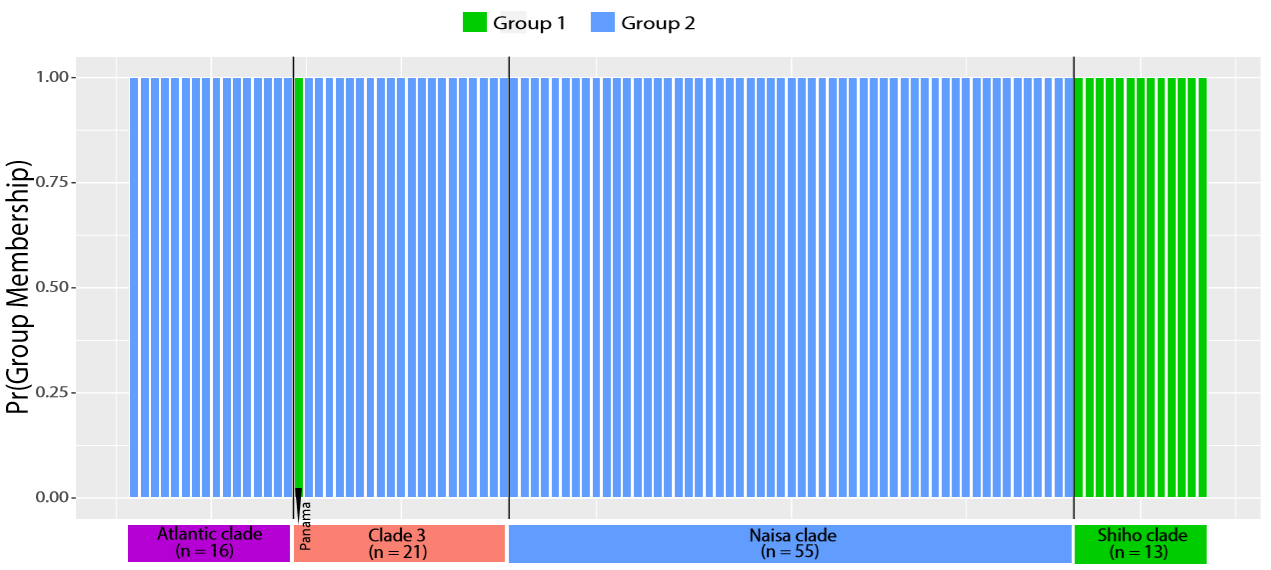
Supplemental Figure S2. BEAST phylogenetic tree including mitogenome haplotype labels.



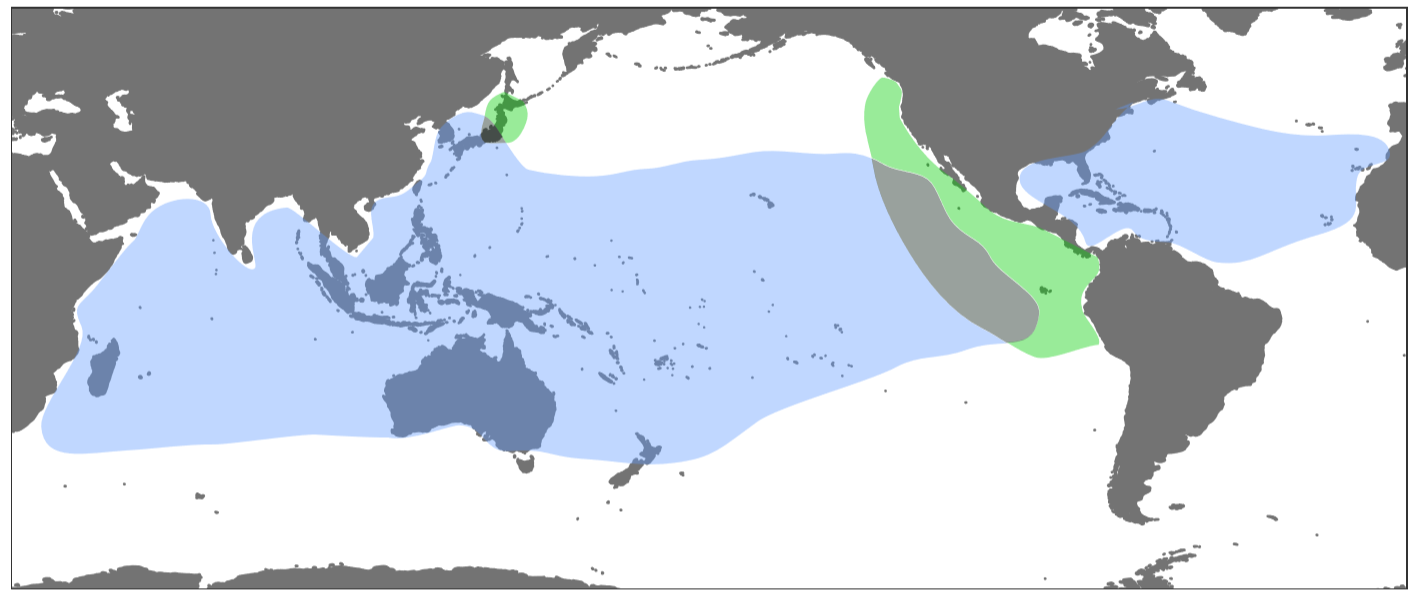
Supplemental Figure S3. Evanno metrics for STRUCTURE analysis of nuclear SNPs, indicating k = 3 as the best supported number of groups.



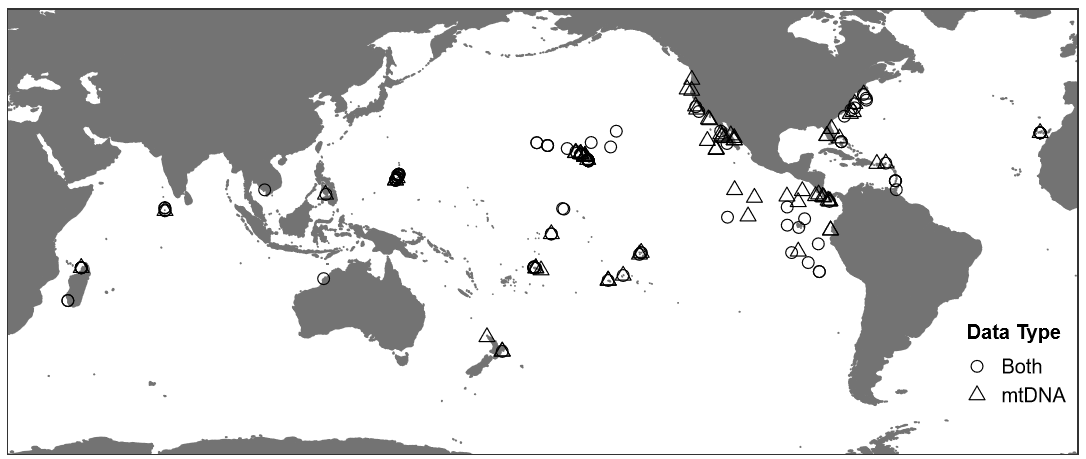
Supplemental Figure S4. Assignment plot based on STRUCTURE analyses, with K = 2. Mitogenome clade stratification is on the x axis, and probability of assignment is on the y axis.



Supplemental Figure S5. Distribution of the two proposed subspecies of short-finned pilot whale. The Naisa short-finned pilot whale is shown in blue. The Shiho short-finned pilot whale is shown in green. Grey areas represent potential mixing or sympatry between the two subspecies.



Supplemental Figure S6. Sampling locations of samples used in the present study.



**Appendix S1: Supplemental File**

Supplemental File S1. Modified R-script for mitogenome genotype assembly. R-script has been modified from alignment scripts deposited in Dryad (Dryad data repository doi:10.5061/dryad.cv35b) to include a more conservative genotype-calling approach in response to reported issues with “index-hopping” by Illumina sequencing machines.

callBasesPileup <- function(plp.fname, min.cov = 7, max.prop.only = 10,

prop.thresh = 0.85, binom.pr.thresh = 0.95,

min.prop = 0.1) {

stopifnot(require(dplyr))

stopifnot(require(tidyr))

stopifnot(require(readr))

stopifnot(require(ape))

# read pileup file

plp <- read\_csv(

plp.fname,

col\_types = cols("c", "c", "i", "i", "c", "c", "l", "d", "i", "d", "d", "d", "d", "d", "d"),

progress = FALSE

) %>%

rename(position = ref.pos) %>%

as.data.frame

base.cols <- c("A", "C", "G", "T", "N", "-")

max.pos <- max(plp$position)

# extract base frequencies from proportions

base.freq <- round(plp[, base.cols] \* plp$coverage, 0)

base.freq$position <- plp$position

base.freq$id <- plp$id

base.freq$coverage <- plp$coverage

# get total proportions of each base at each position

position.base.props <- base.freq %>%

select(-id, -coverage) %>%

group\_by(position) %>%

summarize\_each(funs(sum)) %>%

mutate(total.reads = rowSums(.[, -1])) %>%

mutate\_each(funs(. / total.reads), -position, -total.reads) %>%

gather(base, pool.prop, -position, -total.reads)

# evaluate if frequency and binomial probability are good for base calling

freq.prob.check <- base.freq %>%

gather(base, freq, -id, -position, -coverage) %>%

left\_join(position.base.props, by = c("position", "base")) %>%

filter(freq > 0) %>%

mutate(

read.prop = freq / coverage,

pr.base = pbinom(freq, coverage, pool.prop),

freq.good = coverage >= min.cov & read.prop == 1 | coverage >= max.prop.only & read.prop >= prop.thresh,

prob.good = coverage >= max.prop.only & pool.prop >= 0.5 & read.prop > pool.prop

| coverage >= max.prop.only &

0.875\*read.prop - (pool.prop+0.175) > 0 &

pr.base >= binom.pr.thresh &

read.prop >= min.prop

)

# call bases at each site for each individual

baseCallFunc <- function(base, freq.good, pr.base, prob.good) {

base.call <- base[freq.good]

if(length(base.call) == 1) return(base.call)

max.pr <- which.max(pr.base)

if(prob.good[max.pr]) return(base[max.pr])

"N"

}

base.calls <- freq.prob.check %>%

group\_by(id, position) %>%

summarize(base = baseCallFunc(base, freq.good, pr.base, prob.good))

# translate to DNAbin format full sequences

dna.seqs <- as.matrix(as.DNAbin(do.call(rbind, by(base.calls, base.calls$id, function(x) {

x.seq <- rep("N", max.pos)

x.seq[x$position] <- x$base

x.seq

}))))

# identify positions with variable reads for each individual

plp$variable.bases.in.reads <- apply(plp[, base.cols], 1, function(x) sum(x == 1) == 0)

plp <- plp[, c("position", "id", "variable.bases.in.reads")]

# identify positions with variable calls

vs <- data.frame(position = 1:max.pos)

vs$variable.calls.at.position <- apply(as.character(dna.seqs), 2, function(x) {

length(unique(x)) > 1

})

# add column of variable info to base frequency data.frame

base.freq <- base.freq[, c("position", "id", "coverage", base.cols)] %>%

left\_join(base.calls, by = c("position", "id")) %>%

rename(called.base = base) %>%

left\_join(plp, by = c("position", "id")) %>%

left\_join(vs, by = "position") %>%

arrange(position, id)

# write fasta file

fname <- gsub(".csv", ".fasta", plp.fname)

write.dna(

dna.seqs, fname, format = "fasta", nbcol = -1,

colsep = "", indent = 0, blocksep = 0

)

# return data.frames of metrics for positions and ids, fasta filename, and DNAbin sequences

list(

base.freq.prob.narrow = freq.prob.check,

base.freq.wide = base.freq,

fasta.fname = fname,

dna.seqs = dna.seqs

)

}