Intron Evolution and Fragile Sites

Niels Richard Hansen

November 10, 2017

Contents

About this document

This document constitutes the full analysis of the intron size distribution for ortholog genes across 203 vertebrates. The main results are reported in the manuscript

FANCD2 binding identifies conserved fragile sites at large transcribed genes in avian cells

Constanze Pentzold, Shiraz Ali Shah, Niels Richard Hansen, Benoît Le Tallec, Andaine Seguin-Orlando, Michelle Debatisse, Michael Lisby, Vibe H. Oestergaard.

Data

The `vertGenomes` data set loaded below contains data on ortholog genes from 204 vertebrates. The fish Latimeria chalumnae is removed before the original data set, and the remaining 203 species were used in the analysis.

```
vertGenomes <- read_delim(
  "data/vert.flt.lengths.OG.genes.tax.tsv",
  "\t", escape_double = FALSE,
  col_names = c(
    "ensembl_id",   ## Protein ID
    "glength",      ## Gene length
    "mRNAlength",   ## Length of mature (spliced) mRNA
    "plength",      ## Protein length
    "orthId",       ## Ortholog ID
    "gName",        ## Gene name
    "species",      ## Species name
    "class"         ## Class name (Mammal, reptile (includes three amphibiens), bird, fish)
  ),
  trim_ws = TRUE,
  progress = FALSE)
```

```
## Parsed with column specification:
## cols(
```

```
##   ensembl_id = col_character(),
##   glength = col_integer(),
##   mRNAlength = col_integer(),
##   plength = col_integer(),
##   orthId = col_character(),
##   gName = col_character(),
##   species = col_character(),
##   class = col_character()
## )
```

```
## The total intron length is computed by subtracting the length of (mature) mRNA
## from the length of the gene, and correction of a class label mistake in the data.
vertGenomes <- filter(vertGenomes, species != "Latimeria_chalumnae") %>%
  mutate(ilength = pmax(glength - mRNAlength, 0),
         class = ifelse(species == "Manacus_vitellinus", "bird", class),
         class = ifelse(species == "Callorhinchus_milii", "fish", class))
```

The `galgal` data set below contains the experimental data from Gallus gallus (chicken).

```
galgal <- read_delim(
  "data/galgal5n.flt.chr.features.tsv",
  "\t",
  escape_double = FALSE,
  trim_ws = TRUE)
```

```
## Parsed with column specification:
## cols(
##   ensembl_id = col_character(),
##   size = col_integer(),
##   timing = col_double(),
##   RNAseq = col_integer(),
##   ChIP1 = col_integer(),
##   ChIP2 = col_integer(),
##   within_peak = col_integer(),
##   name = col_character()
## )
```

The data set has 15900 rows, each corresponding to a Gallus gallus gene. The `within_peak` column is an indicator of whether the gene is located within the peak from the chip-chip experiment.

For later use we first extract the ortholog id for those genes that are within the peak.

```
peaks <- filter(galgal, within_peak == 1) %>%
  inner_join(vertGenomes, by = "ensembl_id")
peaksId <- peaks$orthId
```

Initial analysis

This section presents a descriptive analysis of the mean-variance relation of introns per gene. This analysis ignores any systematic differences between species in intron length.

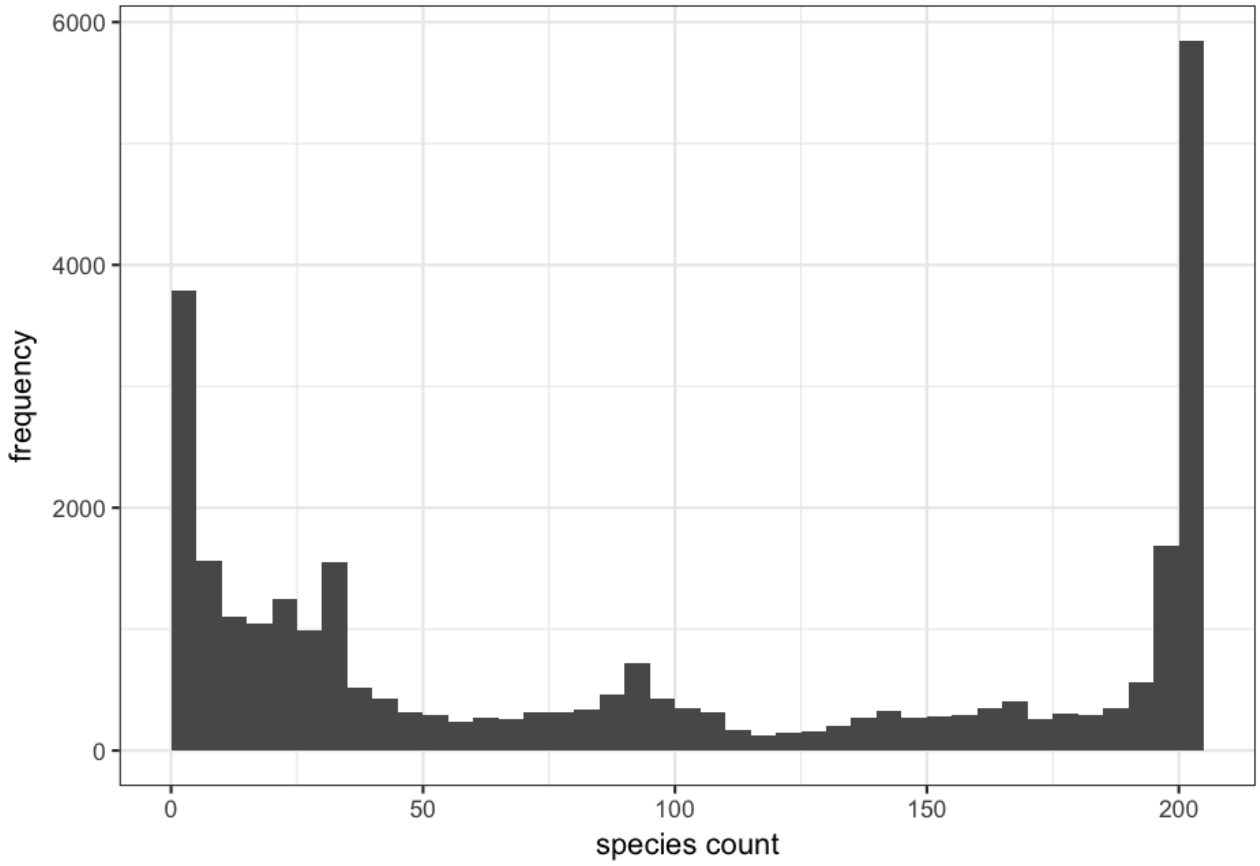
We first summarize the lengths of genes by ortholog id. Note that the empirical standard deviation (or empirical variance for that matter) is most likely a biased estimate of the standard deviation, since the lengths are dependent due to shared ancestry. A general positive correlation is expected, which will make the variance estimator downward biased.

```
vertSum <- vertGenomes %>%
  group_by(orthId) %>%
  summarize(mglength = mean(glength, na.rm = TRUE),
            sglength = sd(glength, na.rm = TRUE),
            mmRNAlength = mean(mRNAlength, na.rm = TRUE),
            smRNAlength = sd(mRNAlength, na.rm = TRUE),
            milength = mean(ilength, na.rm = TRUE),
            silength = sd(ilength, na.rm = TRUE),
            count = n()) %>%
  arrange(count)
## Here we join the summary statistics back into the vert data
## for later usage
vertGenomesSum <- inner_join(vertGenomes, vertSum, by = "orthId")
```

There are 29150 different ortholog ids in the data set.

The following figure shows a histogram of the number of occurrences of each ortholog id.

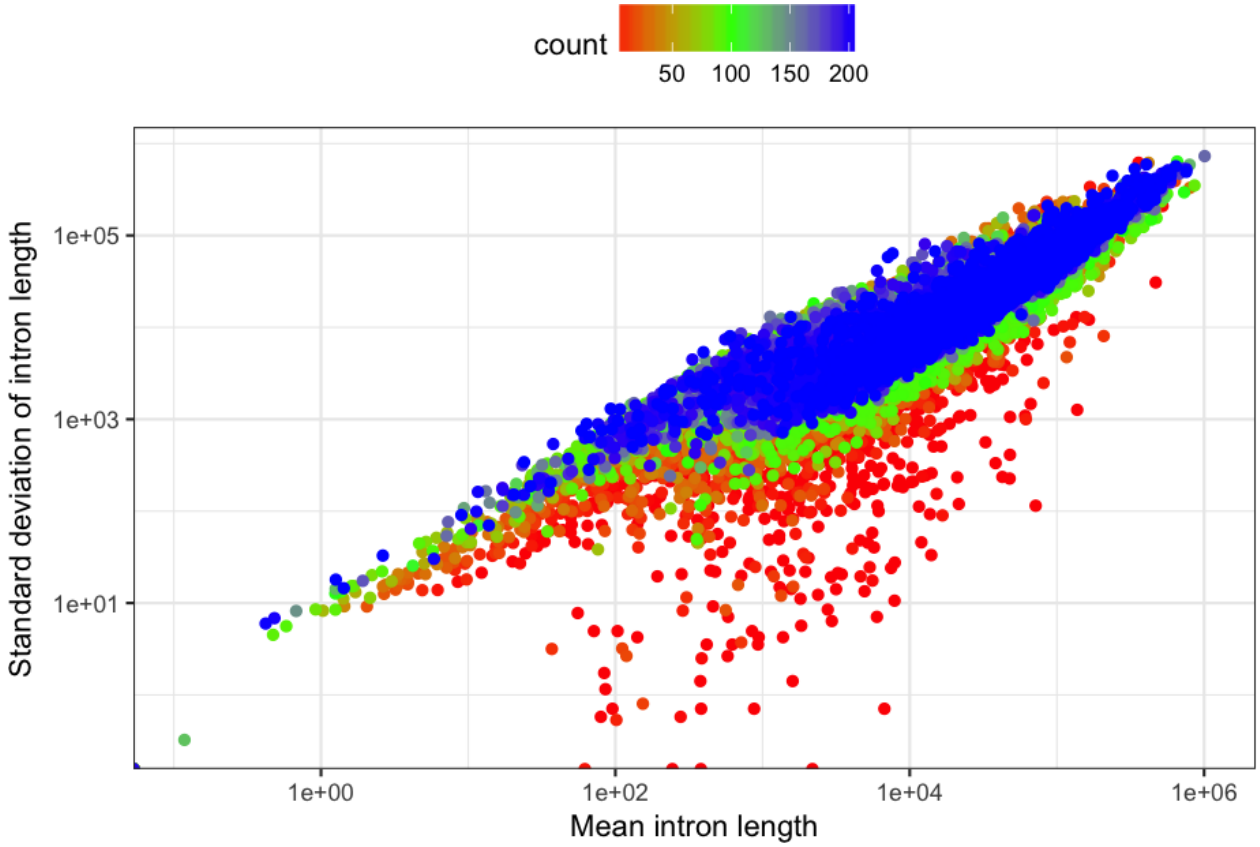
```
ggplot(vertSum, aes(count)) + geom_histogram(binwidth = 5, boundary = 0) +
  xlab("species count") +
  ylab("frequency")
```



We observe a U-shaped distribution with a large fraction of genes present in all or almost all species, and a large fraction present in fewer than 50 species. There are 2161 genes that are present in only one or two species.

The following figure shows the standard deviation of intron length against mean intron length. The points are color coded according to the number of species with the corresponding gene.

```
ggplot(data = vertSum, aes(milength, silength)) +
  scale_x_log10() + scale_y_log10() +
  xlab("Mean intron length") +
  ylab("Standard deviation of intron length") +
  geom_point(aes(color = count)) +
  scale_colour_gradientn(colors = rainbow(3))
```

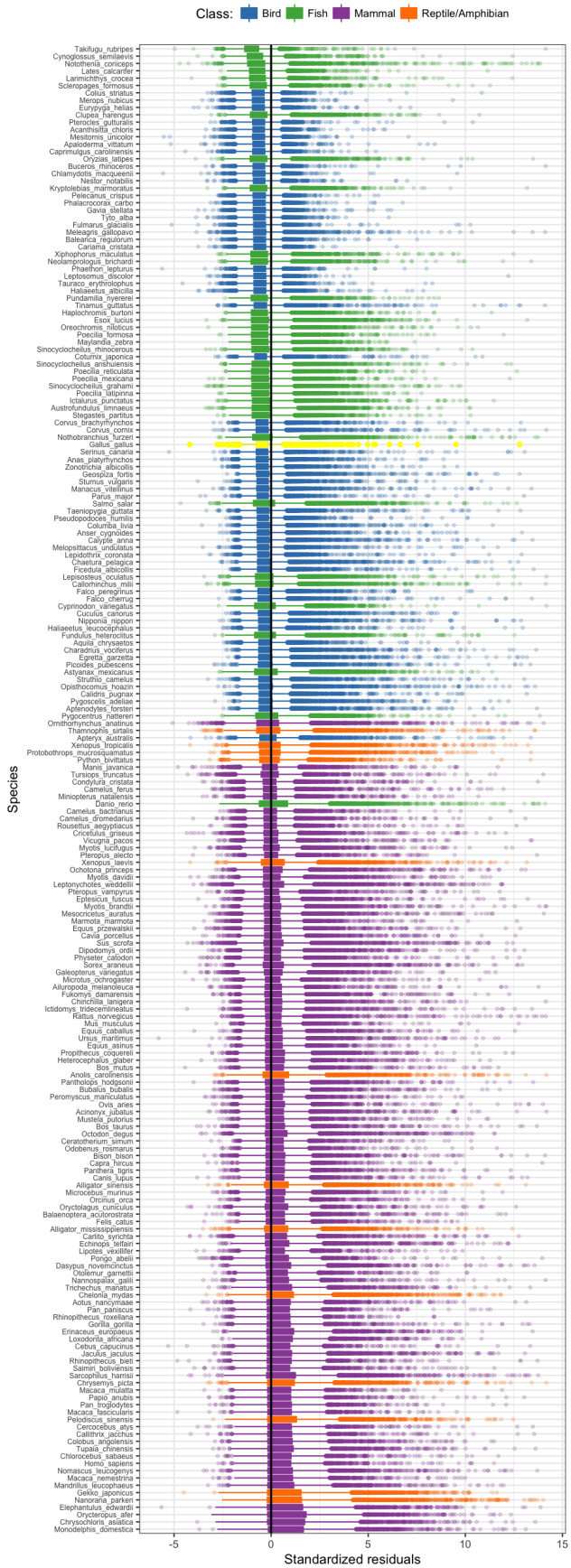


There is a relatively good linear relation between mean and variance. The main extreme deviations correspond to genes present in few (mostly only two corresponding to the red points) species, for which the estimate of variance is expected to be severely downward biased (this is particularly so if those species are evolutionary closely related).

The following is a residual plot in form of boxplots for each species ordered according to the species median residual. Gallus gallus is yellow.

```
specOrd <- vertGenomesSum %>%
  group_by(species) %>%
  summarize(medres = median((ilength - milength) / silength, na.rm = TRUE)) %>%
  arrange(desc(medres)) %>%
  .$species

ggplot(vertGenomesSum, aes(x = factor(species, levels = specOrd),
                           y =  (ilength - milength) / silength,
                           color = class, fill = class)) +
  geom_boxplot(outlier.alpha = 0.2, outlier.size = 1) +
  geom_boxplot(data = filter(vertGenomesSum, species == "Gallus_gallus"),
               fill = "yellow", color = "yellow") +
  geom_abline(intercept = 0, slope = 0, size = 0.8) +
```

```
  coord_flip() +
  xlab("Species") +
  ylab("Standardized residuals") +
  scale_color_manual("Class:", labels = c("Bird", "Fish", "Mammal", "Reptile/Amphibian"),
                     values = classPalette[-1]) +
  scale_fill_manual("Class:", labels = c("Bird", "Fish", "Mammal", "Reptile/Amphibian"),
                     values = classPalette[-1]) +
  theme(axis.text.y = element_text(size = 6),
        legend.position = "top")
```


Class: Bird Fish Mammal Reptile/Amphibian
Species
Standardized residuals
-5
0
5
10
15

We see a clear difference in residual distribution across the different species. In conclusion, introns are generally shorter in some species across genes than in other species.

An additive model

Due to the observed species effect, we construct a model that describes the intron length as a function of both gene (ortholog id) and species.

Some theory

We want to construct an additive effects model:

$$E(t(L_{g,s})) = \alpha_g + \beta_s$$

where $L_{g,s}$ is the intron length of gene g (identified by ortholog id) and species s, and t is a transformation. In the analysis below, the log-transformation ($t(x) = \log(x)$) is used. The model is fitted by least-squares, and the purpose of the transformation is variance stabilization. One alternative could be the square-root transformation.

Fitting this model is a non-trivial problem as the number of levels for the gene factor (the number, 29150, of unique gene ortholog ids) is very large.

We let A and B_0 denote the design matrices with dummy variable encoding of the two factors. With n observations, p genes and q species these matrices are $n \times p$ and $n \times (q-1)$, respectively. The reason that the latter has only $q-1$ columns is that we have to remove one column (use one species as reference species) to have an identifiable parametrization of the additive model. We note that each of these matrices have orthogonal columns, and the projection onto the column space of A is

$$P = A(D_A)^{-1}A^T,$$

where $D_A = A^T A$ is diagonal ($(D_A)_{gg}$ is the number of occurrences of gene g).

It follows that

$$B = (I - P)B_0 = B_0 - A(D_A)^{-1}A^T B_0 = B_0 - A(D_A)^{-1}C,$$

with $C = A^T B_0$, has columns orthogonal to those in A, and that

$$X = \begin{bmatrix} A & B \end{bmatrix}$$

spans the same column space as $\begin{bmatrix} A & B_0 \end{bmatrix}$. The projection onto this column space can therefore be computed as

$$\begin{aligned} Q &= X(X^T X)^{-1}X^T \\ &= \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} D_A & 0 \\ 0 & B^T B \end{bmatrix}^{-1} \begin{bmatrix} A^T \\ B^T \end{bmatrix} \\ &= P + B(B^T B)^{-1}B^T, \end{aligned}$$

which is the sum of projections onto the two orthogonal spaces.

The cross product, $B^T B$, can be computed as follows

$$\begin{aligned} B^T B &= (B_0 - A(D_A)^{-1}C)^T(B_0 - A(D_A)^{-1}C) \\ &= (B_0)^T B_0 - C^T(D_A)^{-1}A^T B_0 - (B_0)^T A(D_A)^{-1}C + C^T(D_A)^{-1}A^T A(D_A)^{-1}C \\ &= D_{B_0} - C^T(D_A)^{-1}C. \end{aligned}$$

The matrix C is computed as a cross-tabulation of genes and species. The diagonal entries in D_A and D_B are the row sums and column sums, respectively, of C.

Multiplication with A^T and B^T can be computed by groupwise summation, and multiplication by A and B can be computed by join operations. The function implemented below uses these operations, cross-tabulation, and linear algebra for $p \times (q-1)$ matrices to compute the projections. The dummy variable encoding is avoided.

Note that

$$\begin{aligned} Qy &= A \underbrace{\tilde{\beta}_A}_{=(D_A)^{-1}A^T y} + B^T \underbrace{\hat{\beta}_B}_{=(B^TB)^{-1}B^T y} \\ &= A\tilde{\beta}_A + B_0^T\hat{\beta}_B - A(D_A)^{-1}C\hat{\beta}_B \\ &= A\underbrace{(\tilde{\beta}_A - (D_A)^{-1}C\hat{\beta}_B)}_{=\hat{\beta}_A} + B_0\hat{\beta}_B \\ &= A\hat{\beta}_A + B\hat{\beta}_B, \end{aligned}$$

which gives the formulas for computing the coefficients in the (A, B_0)-parametrization.

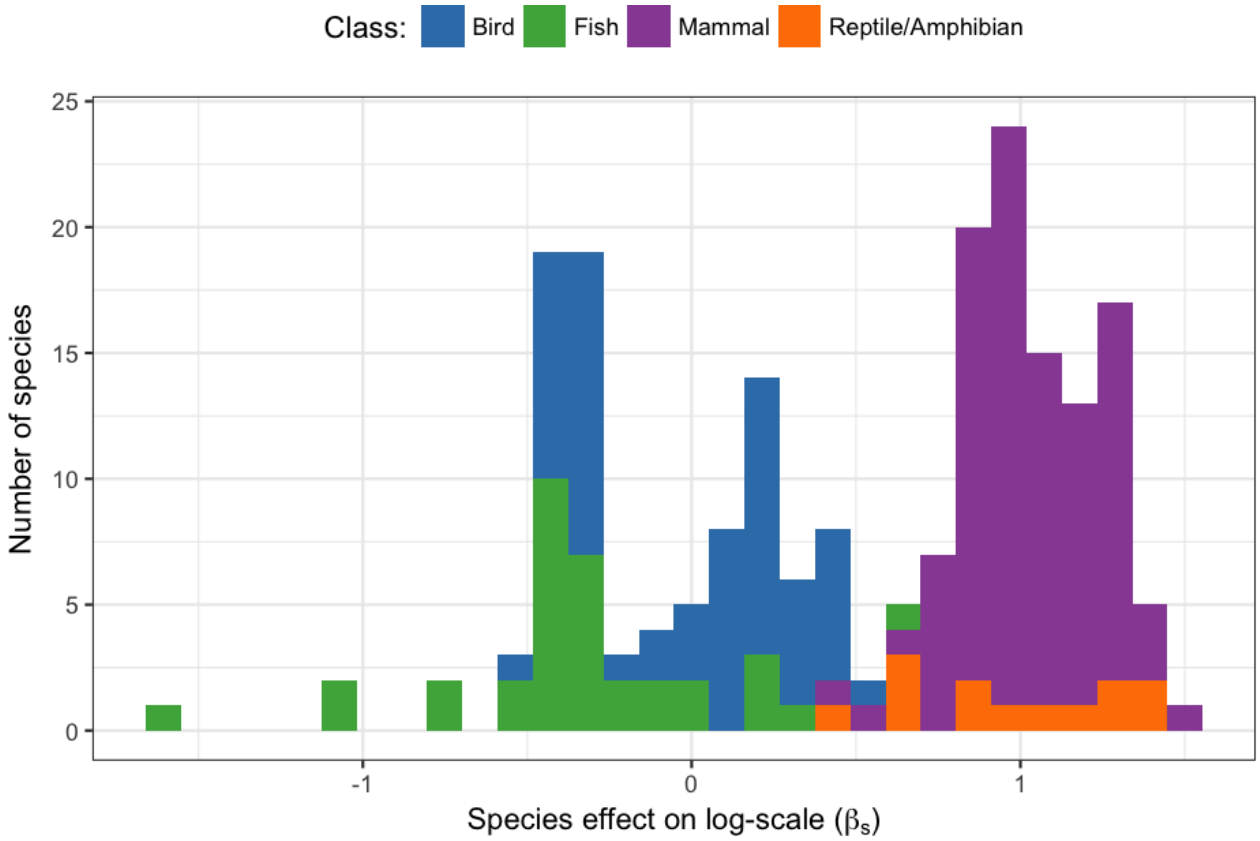
```
## Fits the additive effects model for data in a data frame 'x',
## with 'y' the name of the response column, and 'a' and 'b' the
## names of the two factors.

addBig <- function(x, y, a, b, ref_level = 1, fun = identity, funinv = identity, eps = 0) {
  vars <- lapply(c(a, b), as.symbol)
  funstring <- deparse(substitute(fun))
  sums <- paste("sum(", funstring, "(", y, ")", ")", sep = "")
  means <- list(.betaA = lazyeval::interp(~mean(fun(var)), var = as.name(y)))
  crosstab <- group_by_(x, .dots = vars) %>%
    summarize(count = n()) %>%
    spread_(key_col = b, value_col = "count", fill = 0)
  C <- as.matrix(crosstab[, -1])
  p <- nrow(C)
  q <- ncol(C)
  DA <- rowSums(C)
  C <- C[, - ref_level]
  DB <- colSums(C)
  Aty <- group_by_(x, .dots = vars[1]) %>%
    summarize_(.dots = sums)
  BOty <- group_by_(x, .dots = vars[2]) %>%
    summarize_(.dots = sums)
  beta <- numeric(q)
  beta[-ref_level] <- BOty[[2]][-ref_level] - crossprod(C, DA^(-1) * Aty[[2]])
  beta[-ref_level] <- solve(diag(DB) - crossprod(C, DA^(-1) * C) + diag(eps, length(DB)), beta[-ref_lev
  BOty$.betaB <- beta
  Aty$.betaBA <- as.vector(DA^(-1) * C %*% beta[-ref_level])
  group_by_(x, .dots = vars[1]) %>%
    summarize_(.dots = means) %>%
    inner_join(x, ., by = a) %>%
    inner_join(BOty[, c(1, 3)], by = b) %>%
    inner_join(Aty[, c(1, 3)], by = a) %>%
    mutate(.betaA = .betaA - .betaBA, .hat = funinv(.betaA + .betaB))
}
```

Analysis

The additive model is fitted using the `addBig` function as implemented above. It's impossible to fit the model using standard regression techniques in R due to the large number of genes. The log-transformation (base 2) is used, but other transformations were tried, e.g. no transformation and the square-root-transformation. The log-transformation gave by far the best model fit.

```
thres <- 1 ## Excluding genes with intron length 0
vertGenomes2 <- filter(vertGenomes, ilength >= thres) %>%
  addBig("ilength", "orthId", "species", ref_level = 80, fun = log2, funinv = function(x) 2^x)
```

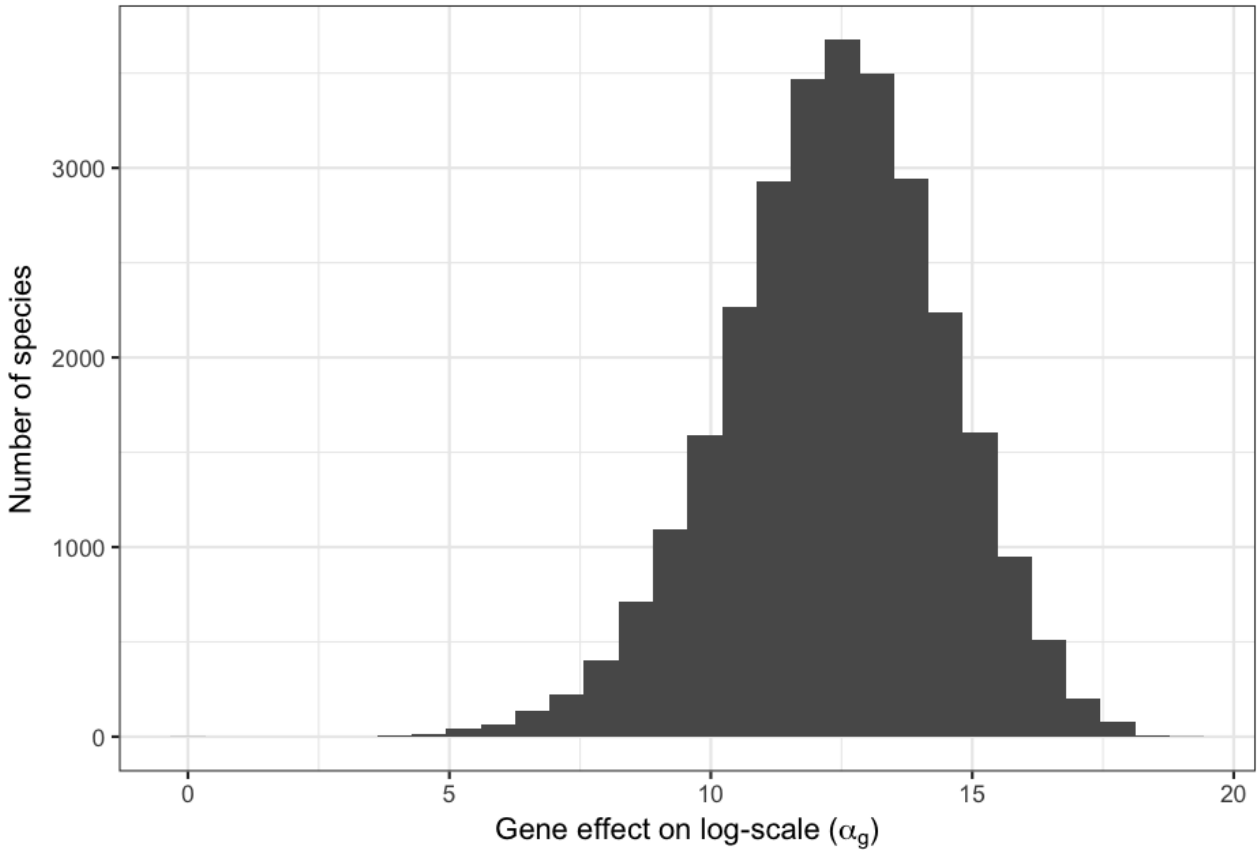
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The histogram above shows the distribution of the species effects on a log-scale. We see a similar stratification of coefficients as observed previously according to species and class. The reference species with species effect 0 is Gallus_gallus.

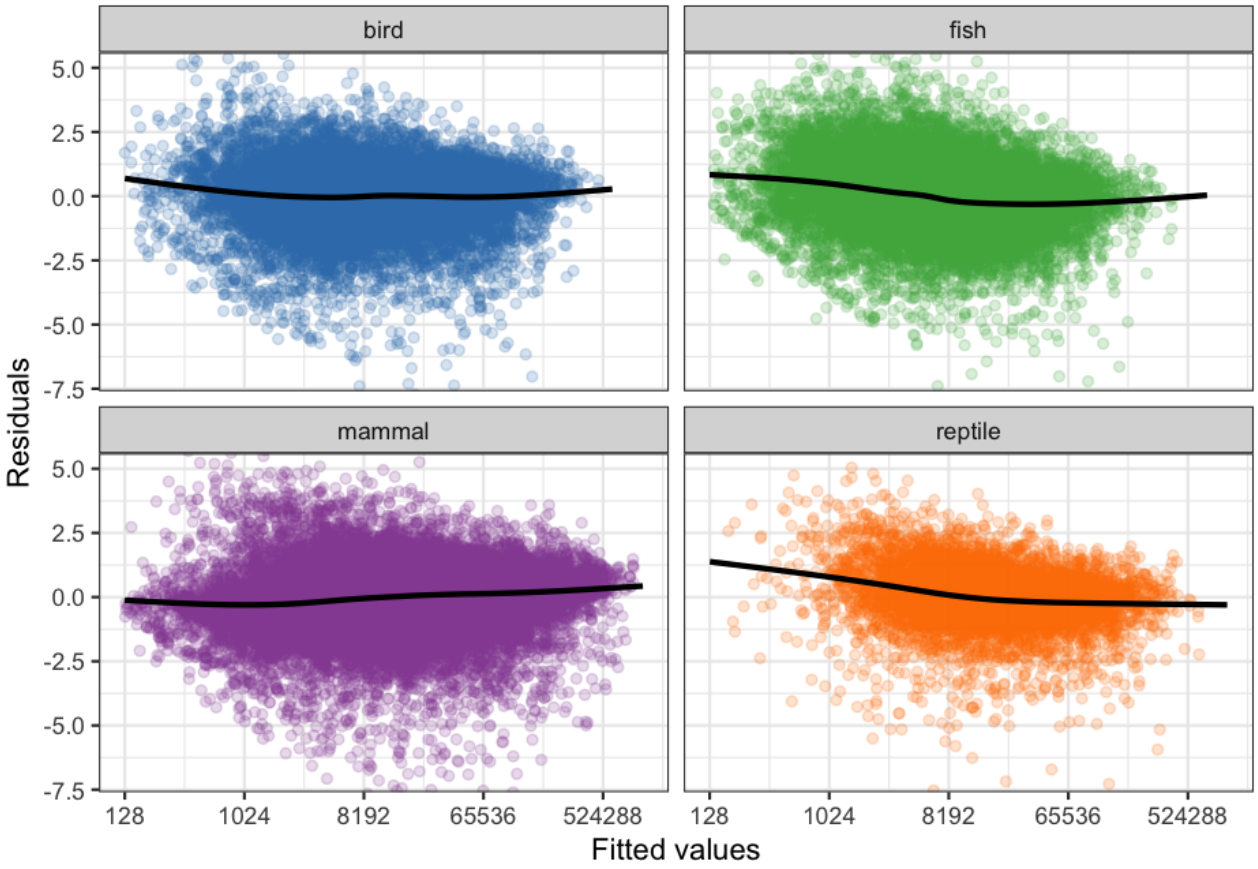
The histogram below shows the gene effects. Note the range of the log-scale with 5 corresponding to an intron length of $2^5 = 32$, 15 to an intron length of $2^{15} = 32768$.

```
vertGenomes2 %>% group_by(orthId) %>%
  summarize(betaA = .betaA[1]) %>%
  ggplot(aes(x = betaA)) +
  geom_histogram() +
  xlab(expression(paste("Gene effect on log-scale (", alpha[g], ")", sep =""))) +
  ylab("Number of species")
```



The following residual plot stratified according to class shows some lack of model fit.

```
nr <- 100000  ## A subset of data points plotted. Smooth fitted using all residuals.
ggplot(data = vertGenomes2[sample(nrow(vertGenomes2), nr), ],
       aes(x = log2(.hat), y = (log2(ilength) - log2(.hat)), color = class)) +
  geom_point(alpha = 0.2) +
  ## Extreme x-values excluded from smoothing
  scale_x_continuous("Fitted values",
                     labels = 2^seq(7, 20, 3),
                     breaks = seq(7, 20, 3),
                     limits = c(7, 20)) +
  ylab("Residuals") +
  ## Extreme y-values are included
  coord_cartesian(ylim = c(-7, 5)) +
  geom_smooth(data = vertGenomes2, color = "black") +
  facet_wrap(~ class, ncol = 2) +
  scale_color_manual(values = classPalette[-1]) +
  theme(legend.position = "none")
```



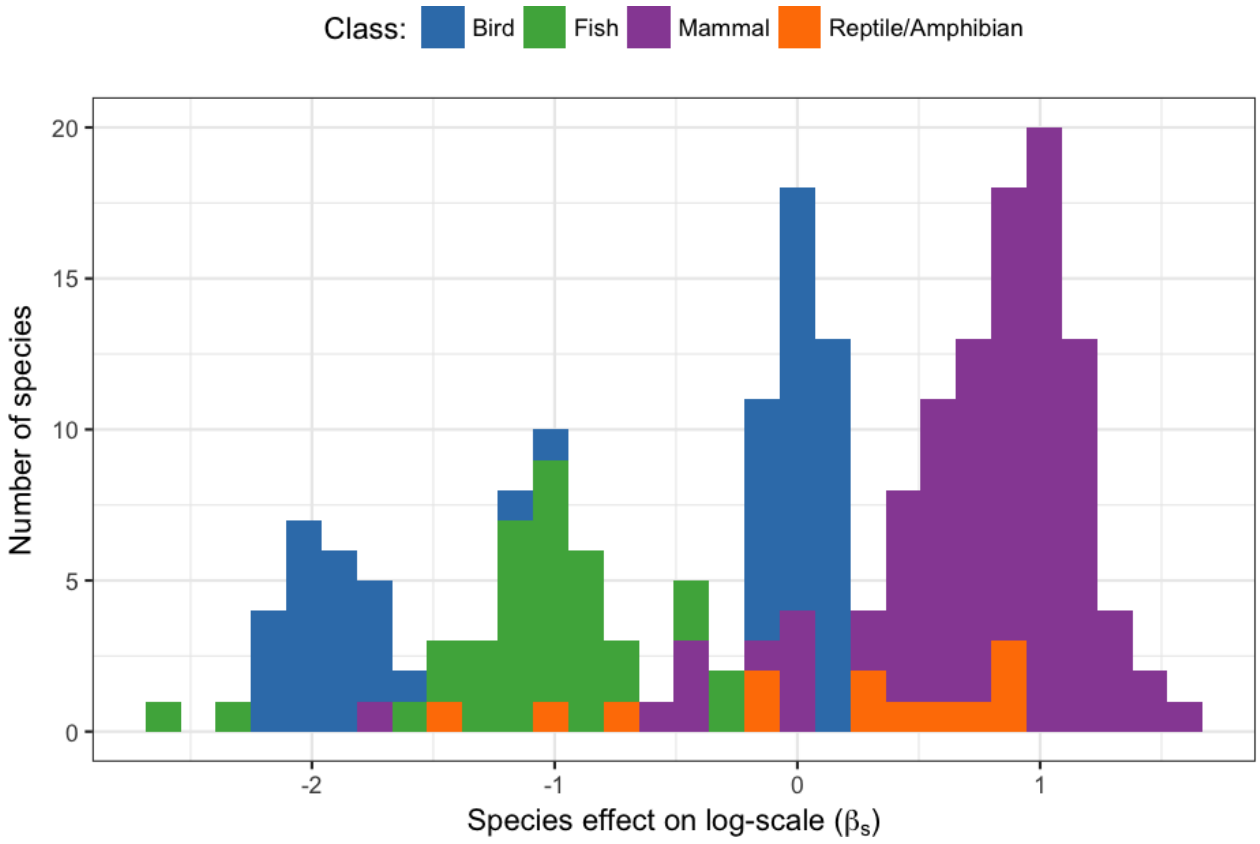
It's unsurprising that the model doesn't fit, as the species effect is unlikely to be a simple additive effect across the vastly different intron lengths.

Conditional Analysis

Due to the lack of model fit, we refit the model using only a subset of genes. These are selected using the gene effect estimate from the additive model as a proxy quantification of "typical gene length". It's clear that this decision can be affected by the composition of species in our data set. We set the threshold to a species effect of 100,000. For a small subset of observations, intron lengths are still extremely short in one or a few species, and we have discarded those extremes as well as they will affect the least squares fit a lot. We have set the threshold on the intron length to be at least 1,000.

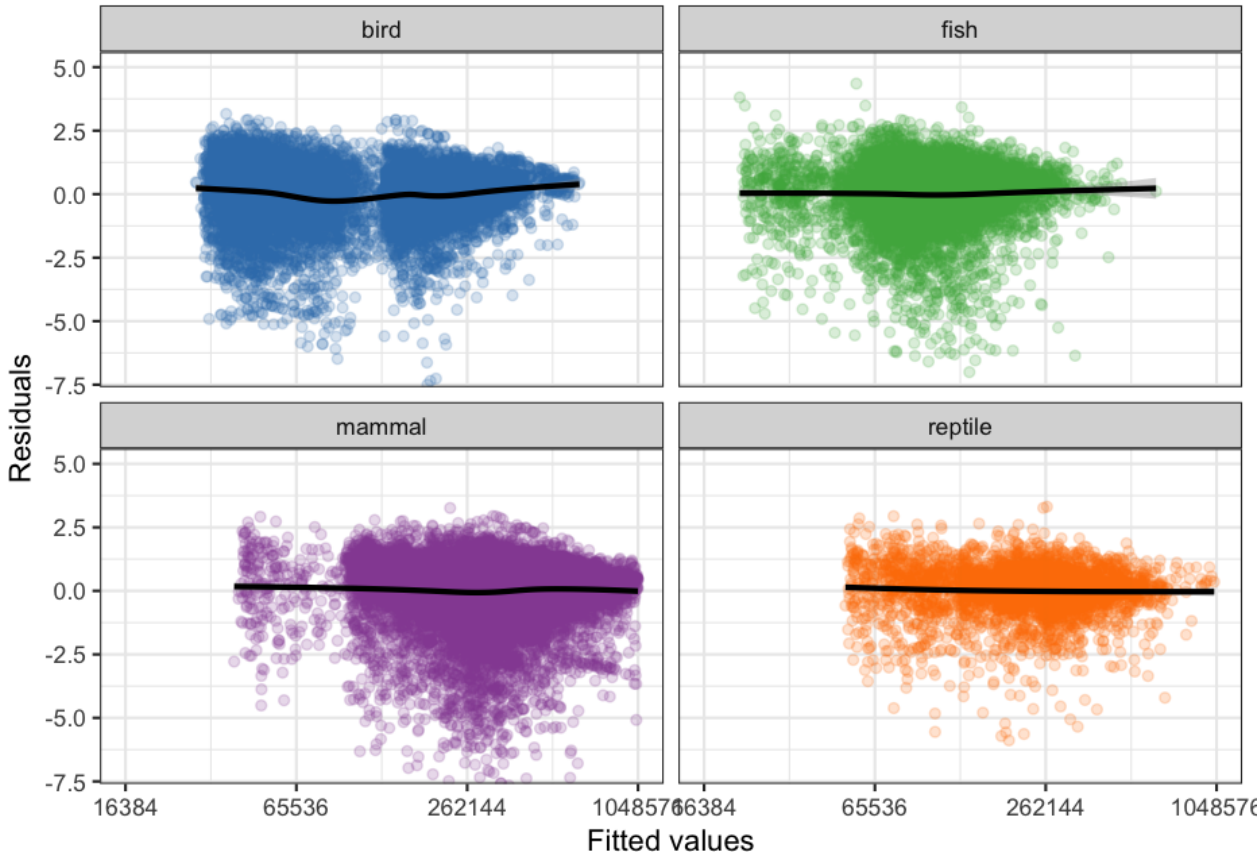
```
vertGenomesLarge <- filter(vertGenomes2, .betaA > log2(100000) & ilength >= 1000) %>%
  select(-c(.betaA, .betaB, .betaBA, .hat))
```

```
vertGenomesLarge <- addBig(vertGenomesLarge,
                           "ilength", "orthId", "species",
                           ref_level = 80, fun = log2, funinv = function(x) 2^x)
group_by(vertGenomesLarge, by = species) %>%
  summarize(betaB = .betaB[1], class = class[1]) %>%
  ggplot(aes(betaB, fill = class)) +
  geom_histogram() +
  scale_fill_manual("Class:", labels = c("Bird", "Fish", "Mammal", "Reptile/Amphibian"),
                    values = classPalette[-1]) +
  xlab(expression(paste("Species effect on log-scale (", beta[s], ")", sep = ""))) +
  ylab("Number of species")
```



The histogram above shows the distribution of the species effects on a log-scale for the analysis conditional on long genes. The reference species with species effect 0 is Gallus_gallus.
The residual plots below (stratified according to class) shows a rather good model fit except for bird with a skewed residual distribution. The central 98% of the intron lengths are within factors 1/16 and 4 of the model prediction. The skewed distribution can, for instance, be explained by some genes having lost a large fraction of its intron in a species due to a large deletion event.

```
ggplot(data = vertGenomesLarge,
       aes(x = log2(.hat), y = (log2(ilength) - log2(.hat)), color = class)) +
  geom_point(alpha = 0.2) +
  ## Extreme x-values excluded from smoothing
  scale_x_continuous("Fitted values",
                     labels = 2^seq(14, 20, 2),
                     breaks = seq(14, 20, 2),
                     limits = c(14, 20)) +
  ylab("Residuals") +
  ## Extreme y-values are included
  coord_cartesian(ylim = c(-7, 5)) +
  geom_smooth(color = "black") +
  facet_wrap(~ class, ncol = 2) +
  scale_color_manual(values = classPalette[-1]) +
  theme(legend.position = "none")
```



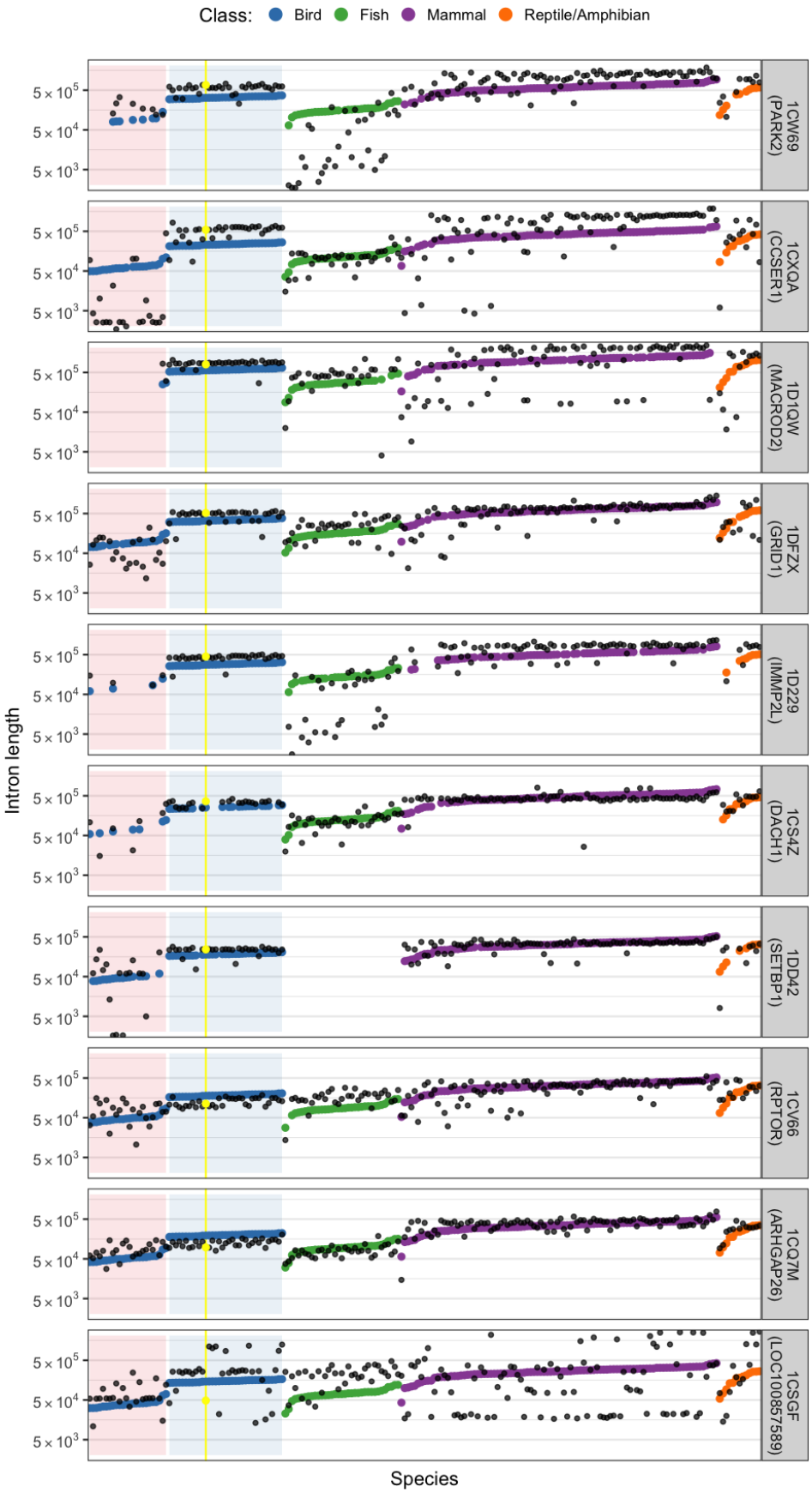
Visualization of fitted models

```
birdThres <- -0.5

vertGenomesLarge <- filter(vertGenomesLarge, species == "Gallus_gallus") %>%
  transmute(orthId = orthId, label = paste(orthId, "\n(", gName, ")", sep = "")) %>%
  inner_join(vertGenomesLarge, .)
```

```
## Joining, by = "orthId"
```

```
orthOrd <- filter(vertGenomesLarge, species == "Gallus_gallus", orthId %in% peaksId) %>%
  arrange(.hat - ilength) %>%
  .$label

specOrd <- group_by(vertGenomesLarge, species) %>%
  summarize(class = class[1], .betaB = .betaB[1]) %>%
  arrange(class, .betaB) %>%
  .$species

nrShortBird <- filter(vertGenomesLarge, class == "bird" & .betaB < birdThres) %>%
  group_by(species) %>% summarize() %>% count() %>% .$n
nrBird <- filter(vertGenomesLarge, class == "bird") %>%
  group_by(species) %>% summarize() %>% count() %>% .$n
nrGallus <- which(specOrd == "Gallus_gallus")

vertPeaks <- vertGenomesLarge %>% filter(orthId %in% peaksId) %>%
```

```
  mutate(label = factor(label, levels = orthOrd))

Gallus <- filter(vertPeaks, species == "Gallus_gallus")

p1 <- ggplot(vertPeaks, aes(species, .hat)) +
  geom_polygon(data = data.frame(species = c(1, nrShortBird, nrShortBird, 1),
                                 .hat = c(2^11, 2^11, 2^21, 2^21)),
               fill = "#e41a1c", alpha = 0.1) +
  geom_polygon(data = data.frame(species = c(nrShortBird + 1, nrBird, nrBird, nrShortBird + 1),
                                 .hat = c(2^11, 2^11, 2^21, 2^21)),
               fill = "#377eb8", alpha = 0.1) +
  geom_vline(xintercept = nrGallus, color = "yellow") +
  geom_point(aes(color = class)) +
  geom_point(aes(y = ilength), size = 1, alpha = 0.7) +
  geom_point(data = Gallus, aes(y = ilength), color = "yellow") +
  facet_grid(label ~ .) +
  scale_y_continuous("Intron length", breaks = c(5000, 50000, 500000),
                     labels = c(quote(5 %*% 10^3), quote(5 %*% 10^4), quote(5 %*% 10^5)),
                     trans = "log2") +
  coord_cartesian(ylim = c(2^11, 2^21)) +
  scale_x_discrete("Species", breaks = c(), limits = specOrd) +
  scale_color_manual("Class:", labels = c("Bird", "Fish", "Mammal", "Reptile/Amphibian"),
                     values = classPalette[-1]) +
  guides(colour = guide_legend(override.aes = list(size = 3)))
p1
```


Class: Bird Fish Mammal Reptile/Amphibian
Intron length
Species
1CW69 (PARK2)
1CXQA (CCSER1)
1D1QW (MACROD2)
1DFZX (GRID1)
1D229 (IMMP2L)
1CS4Z (DACH1)
1DD42 (SETBP1)
1CV66 (RPTOR)
1CQ7M (ARHGAP26)
1CSGF (LOC100857589)
5 × 10^5
5 × 10^4
5 × 10^3

Interactions between class and gene for large genes

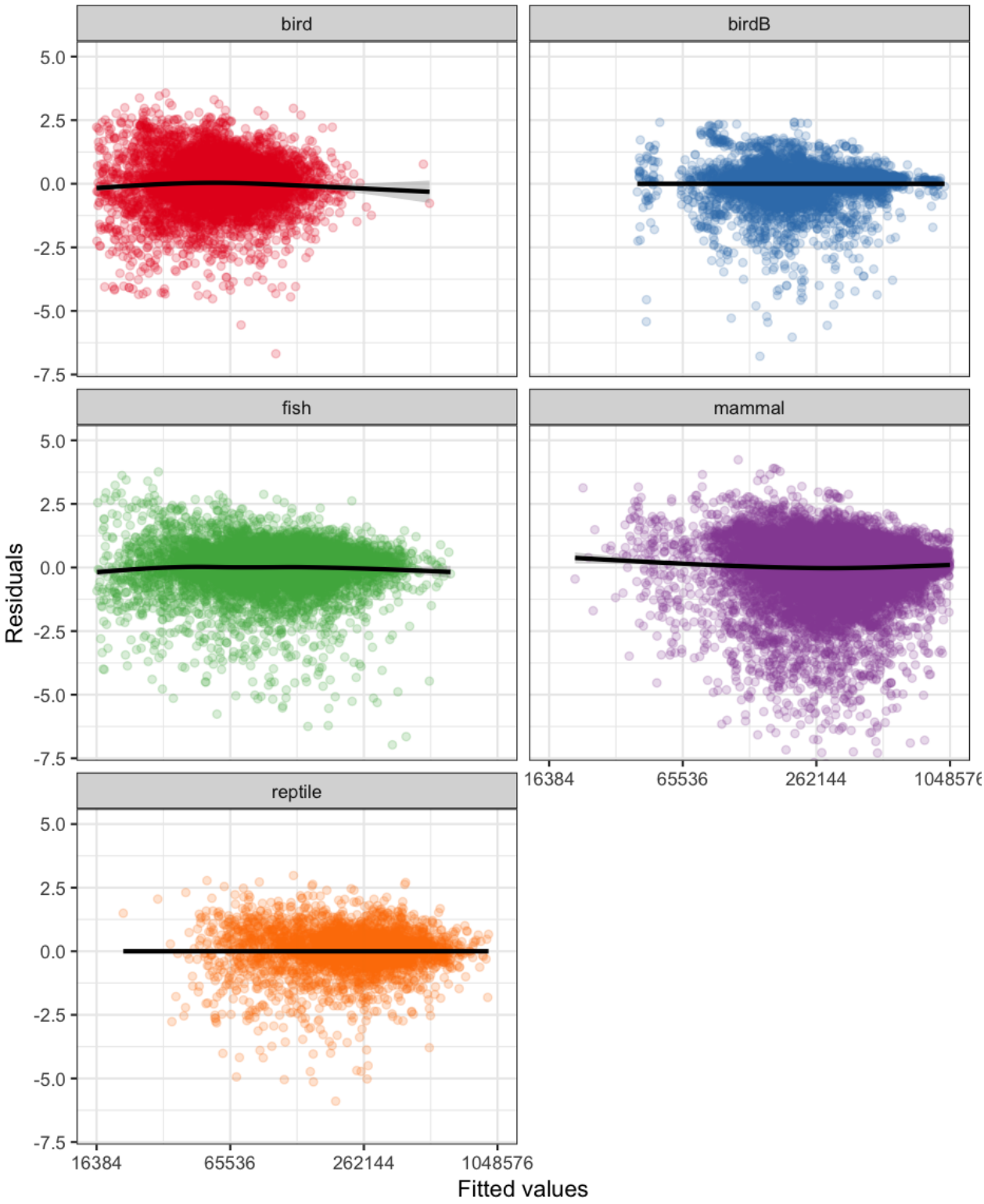
The bird class is divided into two according to the clear separation of typical intron length as depicted on the figure showing species effect in the conditional analysis of the long genes.

```
vertGenomesLargeInt <- mutate(vertGenomesLarge,
                              class2 = ifelse(class == "bird" & .betaB < birdThres, "birdA", class),
                              class2 = ifelse(class2 == "bird", "birdB", class),
                              classOrth = paste(class2, orthId, sep = "_"))
```

The additive model is refitted with interactions between gene and class. Including the interactions requires reference levels for each class. These are taken as: Acanthisitta_chloris for birdB (1), Alligator_mississippiensis for reptile/Amphibian (4), Astyanax_mexicanus for fish (14), Gallus_gallus for bird (80), and Homo_sapiens for mammal (89).

```
vertGenomesLargeInt <- select(vertGenomesLargeInt,
                              orthId, ilength, classOrth, class2, species, label) %>%
  addBig("ilength", "classOrth", "species",
         ref_level = c(1, 4, 14, 80, 89), fun = log2, funinv = function(x) 2^x)
```

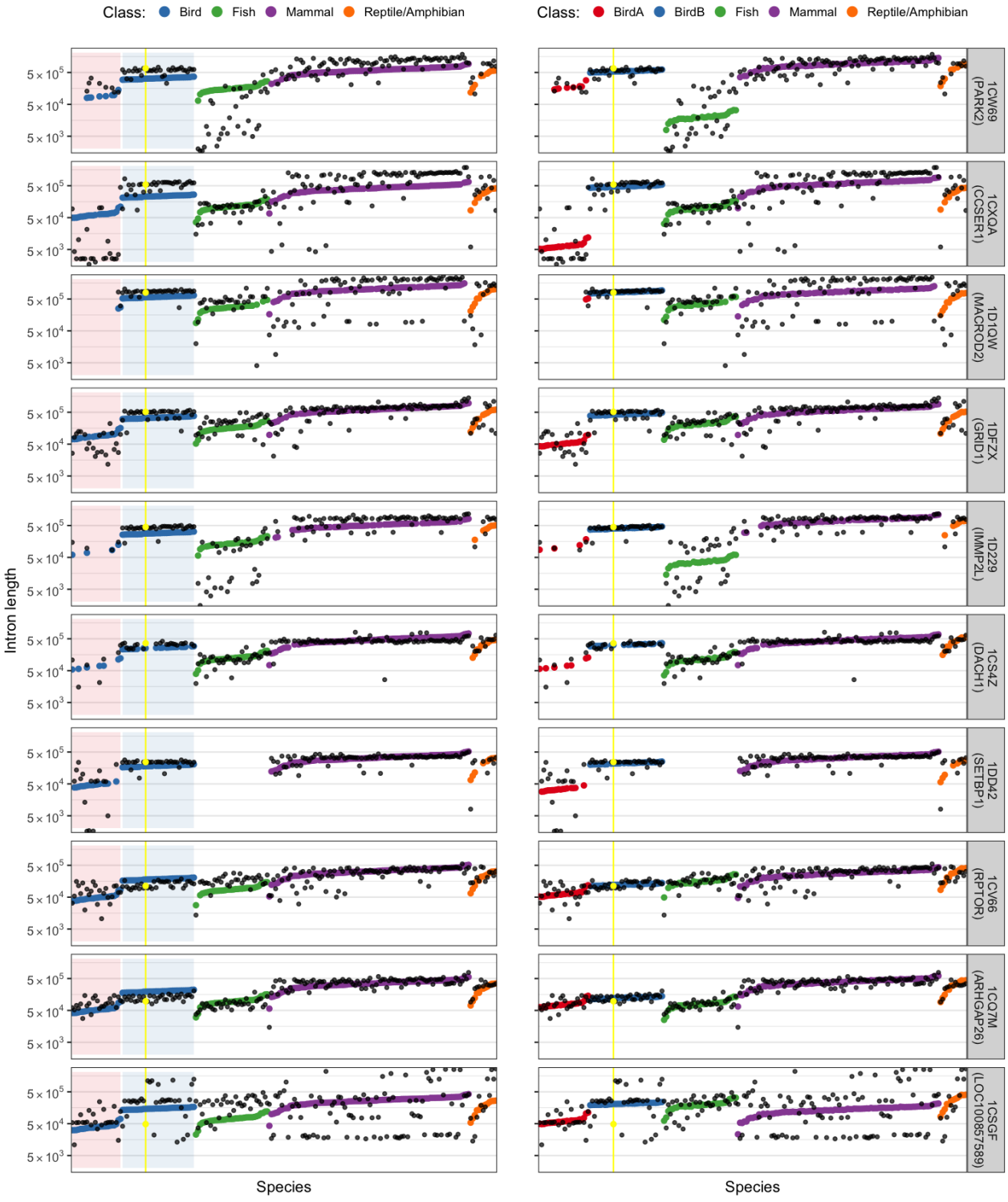
```
ggplot(data = vertGenomesLargeInt,
       aes(x = log2(.hat), y = (log2(ilength) - log2(.hat)), color = class2)) +
  geom_point(alpha = 0.2) +
  ## Extreme x-values excluded from smoothing
  scale_x_continuous("Fitted values",
                     labels = 2^seq(14, 20, 2),
                     breaks = seq(14, 20, 2),
                     limits = c(14, 20)) +
  ylab("Residuals") +
  ## Extreme y-values are included
  coord_cartesian(ylim = c(-7, 5)) +
  geom_smooth(color = "black") +
  facet_wrap(~ class2, ncol = 2) +
  scale_color_manual(values = classPalette) +
  theme(legend.position = "none")
```



Comparative visualization of fitted model

```
vertIntPeaks <- vertGenomesLargeInt %>% filter(orthId %in% peaksId) %>%
  mutate(label = factor(label, levels = orthOrd))
```

```
p2 <- ggplot(vertIntPeaks, aes(species, .hat)) +
  geom_vline(xintercept = nrGallus, color = "yellow") +
  geom_point(aes(color = class2)) +
  geom_point(aes(y = ilength), size = 1, alpha = 0.7) +
  geom_point(data = Gallus, aes(y = ilength), color = "yellow") +
  facet_grid(label ~ .) +
  scale_y_continuous("", breaks = c(5000, 50000, 500000),
                     labels = c("", "", ""),
                     trans = "log2") +
  coord_cartesian(ylim = c(2^11, 2^21)) +
  scale_x_discrete("Species", breaks = c(), limits = specOrd) +
  scale_color_manual("Class:", labels = c("BirdA", "BirdB", "Fish", "Mammal", "Reptile/Amphibian"),
                     values = classPalette) +
  guides(colour = guide_legend(override.aes = list(size = 3)))

p1 <- p1 + theme(strip.background = element_blank(),
       strip.text.y = element_blank())

gridExtra::grid.arrange(p1, p2, ncol = 2)
```


Class: Bird Fish Mammal Reptile/Amphibian
Class: BirdA BirdB Fish Mammal Reptile/Amphibian
Intron length
5×10^5
5×10^4
5×10^3
1CW69 (PARK2)
1CXQA (CCSER1)
1D1QW (MACROD2)
1DFZX (GRID1)
1D229 (IMMP2L)
1CS4Z (DACH1)
1DD42 (SETBP1)
1CV66 (RPTOR)
1CQ7M (ARHGAP26)
1CSGF (LOC100857589)
Species
Species