Selection shapes synonymous stop codon use in mammals

R code and data required to reproduce figures and supplementary figures for our analysis of orthologue families from the OrthoMaM (v.8) database using the stop-extended codon model.

Figures

The following section describes: * “R” packages * Description * Code used to generate each figure * Plot generated

R packages and pre-required data

- Require ggplot2

```
# Load in library
require(ggplot2)

## Loading required package: ggplot2

# Read the data table
t = read.table("gene_data")
```

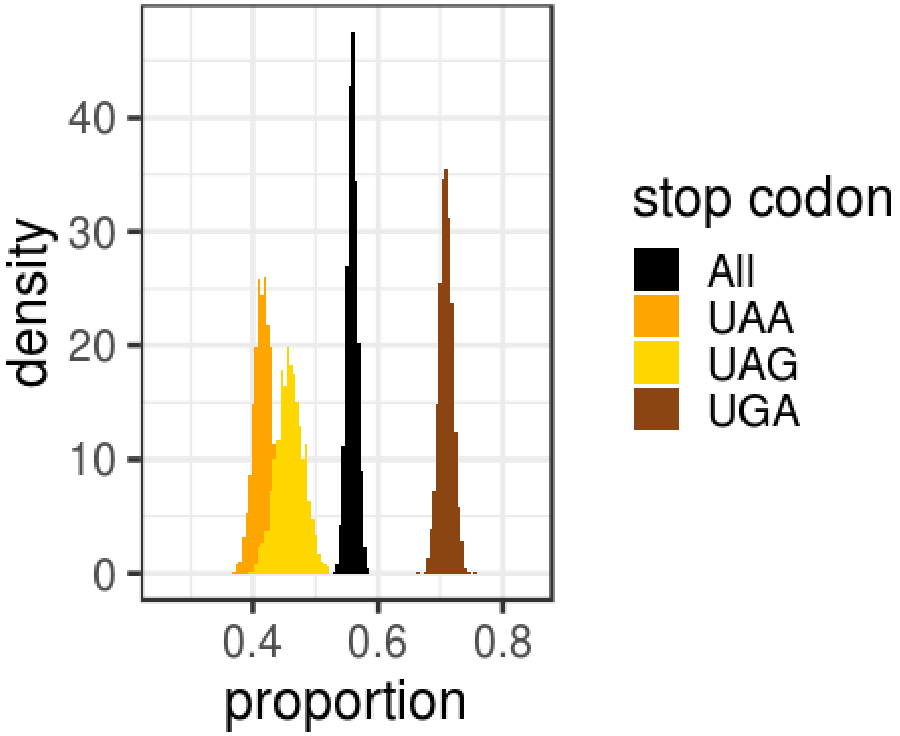
Description

Fig 1

Estimated proportions of stop codons under purifying selection for all genes (black) and for genes with UGA, UAG and UGA shown in brown, yellow and orange, respectively (or umber, amber and opal, according to the colour nomenclature for stop codons). Each histograms was derived from 1000 bootstrap replicates of the data.

Code

```
####Figure 1: mixture_histograms
figure1 = function() {
bootstraps = 1000
Ws = read.table("Bootstraps.weights")
Phis = read.table("Bootstraps.phis")
Ws$UAG[Phis$UAG > 0.99] = NA  ##Weight not estimable for phi close to
one (which can occur for UAG, due to low proportion under selection)
Phis$UAG[Ws$UAG < 0.01] = NA  ##Phi not estimable for weight close to
zero
proportion = c(Ws$UGA,Ws$UAG,Ws$UAA,Ws$All)
stop_codon =
c(rep('UGA',bootstraps),rep('UAG',bootstraps),rep('UAA',bootstraps),rep('All',b
ootstraps))
df = data.frame(proportion=proportion,stop_codon = stop_codon)
colnames(df) = c("proportion","stop codon")
ggplot(df, aes(proportion, fill = `stop codon`)) + theme_bw(base_size = 20)
+ geom_histogram( aes(y = ..density..), position = 'identity', binwidth=0.005)
+ scale_fill_manual(values=c("black", "orange", "gold","chocolate4")) +
xlim(0.25,0.85)
}
################

figure1()
```



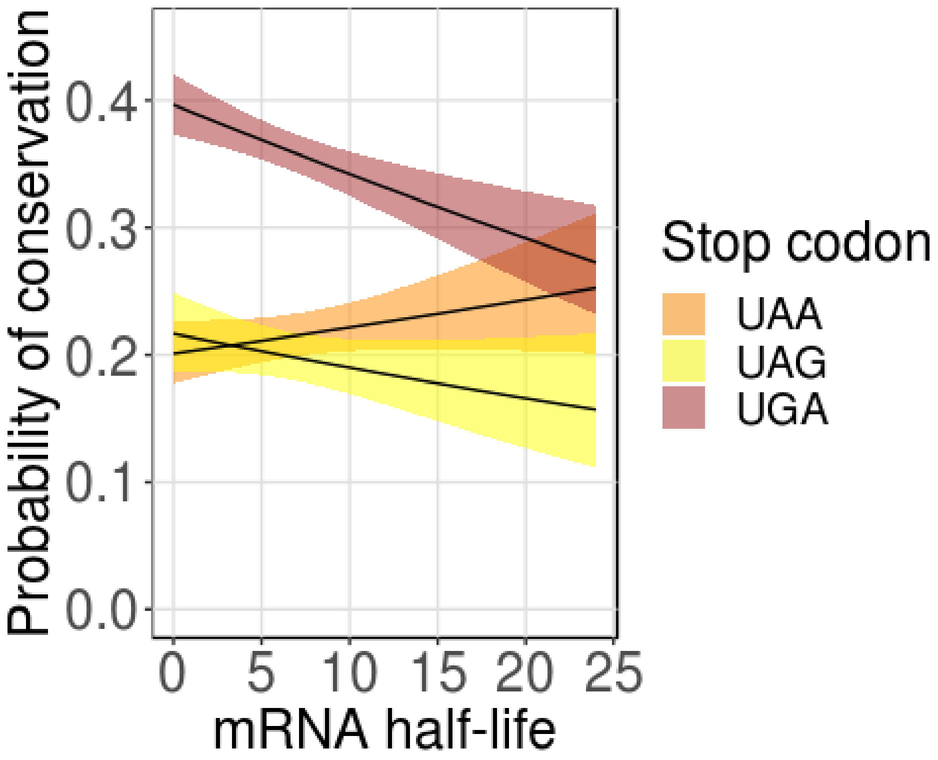
Description

Fig 2a and Fig 2b

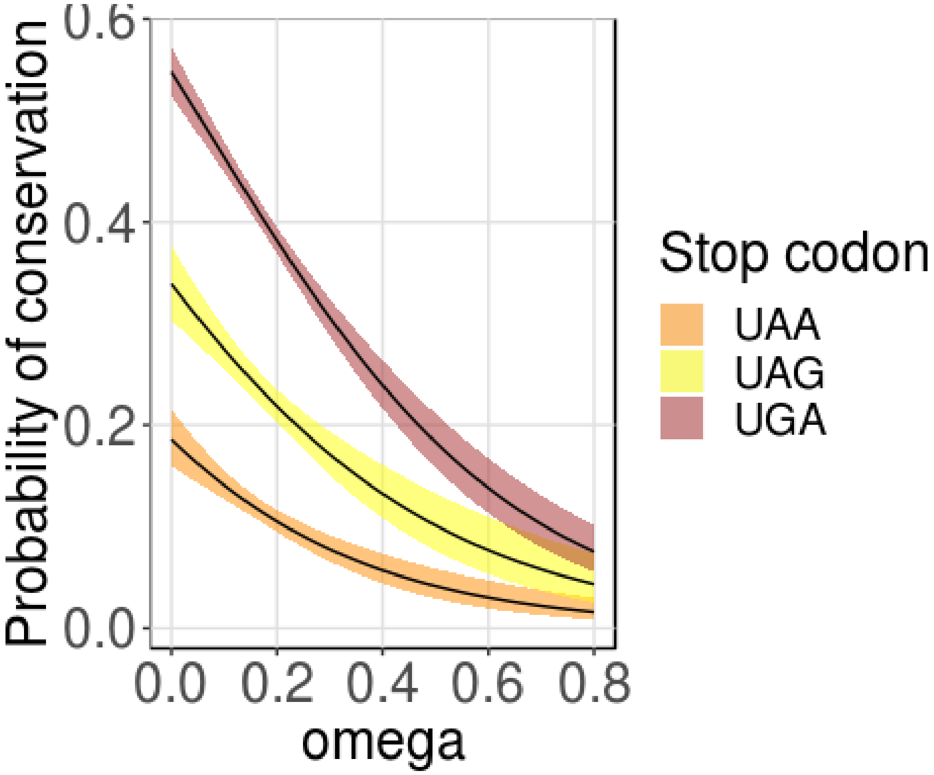
Estimates are from a logistic regression model, which included the number of taxa for which the stop codon was positionally homologous with the end of the alignment as a covariate

Code

```
# Figure 2a: Plot probability of conservation as a function of half-life, with
stop codon interaction
figure2a = function() {
  txtsize = 20
  summary(lm(hl~cons+genegc+cdslen+omega+utrlen+cons:humstop +
taxcount,data=t))
  gcglm = glm(cons ~ hl * humstop + taxcount ,family="binomial", data=t)
  newdata = data.frame( hl = rep(seq(from=0,to = 24,
length.out=100),3),humstop =
c(rep('UAG',100),rep('UGA',100),rep('UAA',100)),taxcount=rep(median(t$taxc
ount,na.rm=T),300))
  predictdata = cbind(newdata,predict(gcglm,newdata=newdata,se=T))
  predictdata = within(predictdata, { predictProb = plogis(fit)
  LL = plogis(fit-1.96*se.fit)
  UL = plogis(fit + 1.96*se.fit)
  })
  ggplot(predictdata,aes(x=hl,y=predictProb)) +
    geom_ribbon(aes(ymin=LL,ymax=UL,fill=factor(humstop)),alpha=.5) +

geom_line(data=predictdata[predictdata$humstop=='UAA',],color="black")+

geom_line(data=predictdata[predictdata$humstop=='UAG',],color="black")+

geom_line(data=predictdata[predictdata$humstop=='UGA',],color="black")+
    xlab("mRNA half-life") + ylab("Probability of conservation") +
    theme(text=element_text(size=txtsize), axis.text.x =
element_text(size=txtsize),axis.text.y = element_text(size=txtsize),
panel.background = element_rect(fill= "white", colour="black"),
panel.grid.minor = element_blank(), panel.grid.major =
element_line(colour="grey88"))+
    guides(fill=guide_legend(title="Stop codon")) + ylim(0,0.45) +
    scale_fill_manual(values=c("darkorange","yellow","brown"))
  }

figure2a()
```



```
#Figure 2b: Probability of conservation as a function of omega
figure2b = function() {
txtsize = 20
omegaglm = glm(conserved ~ omega * humstop +
taxcount,family="binomial", data=t)
newdata = data.frame( omega = rep(seq(from=0,to = 0.8,
length.out=100),3),humstop =
c(rep('UAG',100),rep('UGA',100),rep('UAA',100)),taxcount=rep(median(t$taxc
ount,na.rm=T),300))
predictdata = cbind(newdata,predict(omegaglm,newdata=newdata,se=T))
predictdata = within(predictdata, { predictProb = plogis(fit)
LL = plogis(fit-1.96*se.fit)
UL = plogis(fit + 1.96*se.fit)
})
ggplot(predictdata,aes(x=omega,y=predictProb)) +
geom_ribbon(alpha=0.5,aes(ymin=LL,ymax=UL,fill=factor(humstop))) +
  geom_line(data=predictdata[predictdata$humstop=='UAA',],color="black")
+
  geom_line(data=predictdata[predictdata$humstop=='UAG',],color="black")
+
  geom_line(data=predictdata[predictdata$humstop=='UGA',],color="black")
+
  xlab("omega") + ylab("Probability of conservation") +
theme(text=element_text(size=txtsize), axis.text.x =
element_text(size=txtsize),axis.text.y =
element_text(size=txtsize),panel.background = element_rect(fill= "white",
colour="black"), panel.grid.minor = element_blank(), panel.grid.major =
element_line(colour="grey88"))+
  guides(fill=guide_legend(title="Stop codon")) +
  scale_fill_manual(values=c("darkorange","yellow","brown"))
}

figure2b()
```



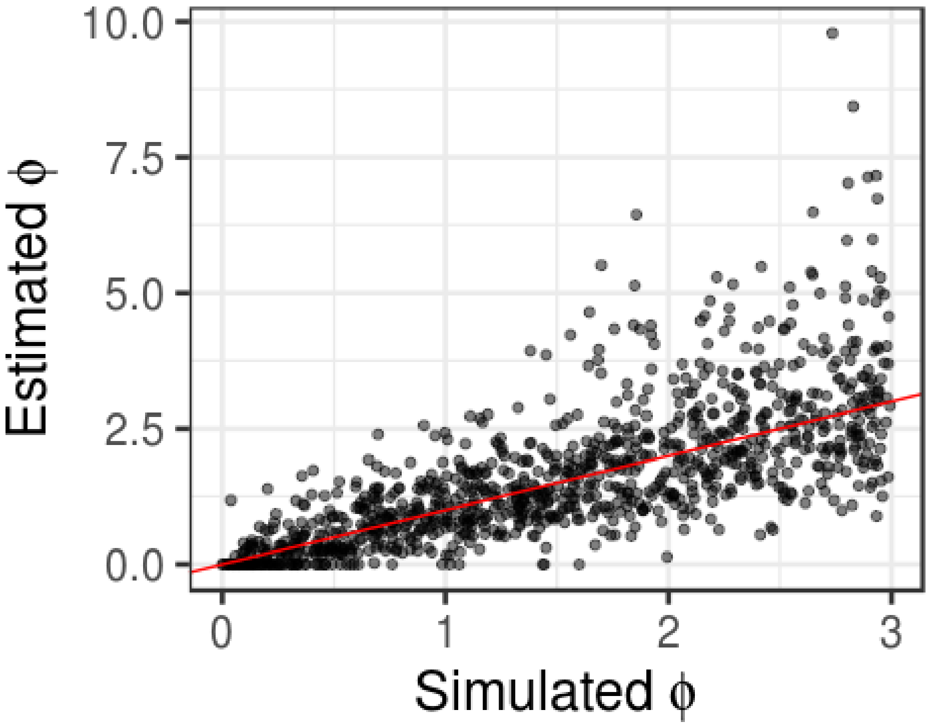
Description

Fig S1

Results of simulations based on 1,000 orthologue alignments, randomly sampled from OrthoMaM. Sequences were simulated over the phylogenetic trees corresponding to the sequence alignments, under a stop-extended codon model based on MG with the F1x4 model of codon frequencies. Simulated values of φ were sampled uniformly between 0 and 3. Branch lengths of the phylogenies were re-estimated from the simulated alignments, under the MG F1x4 model, using codonPhyml. The stop-extended codon model was then fitted to the simulated alignments. The figure shows the maximum likelihood estimates, plotted against simulated values of φ . The identity line is shown in red

Code

```
#figure_sim_var: check bias in the estimate of phi
figure_sim_var = function(file) {
  sim = read.table(file,h=T)
  ggplot(sim,aes(Phi_sim,Phi)) + theme_bw(base_size = 20) +
geom_point(alpha=0.5) + geom_abline(slope=1,intercept=0,col="red") +
xlab(expression("Simulated"~phi)) + ylab(expression("Estimated"~phi))
}

#Fig S1: sim_mod_var
figure_sim_var('sim_variable_phi.out')
```



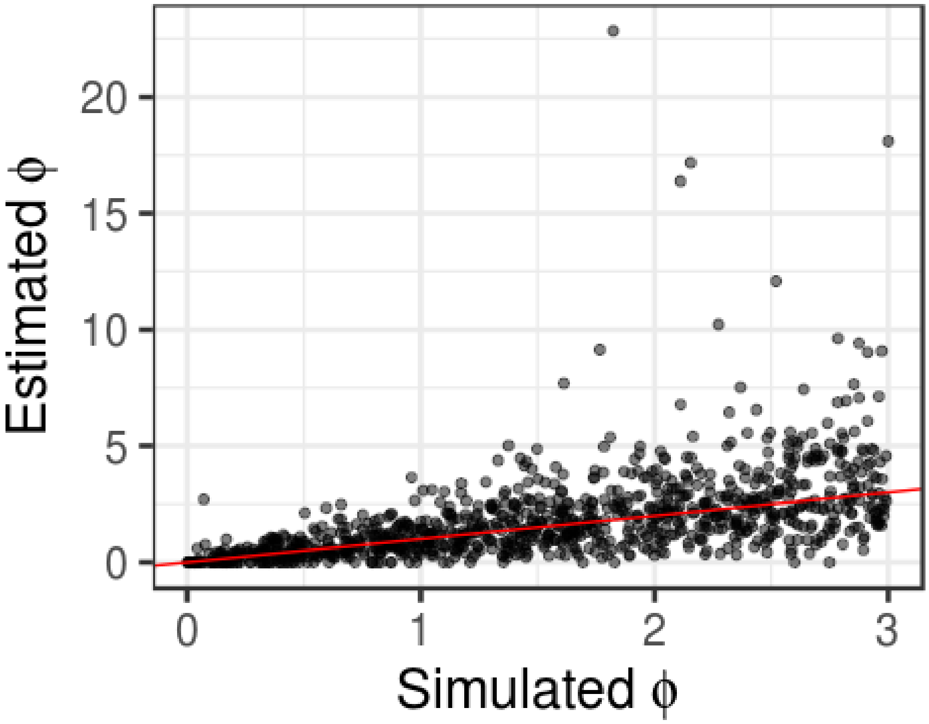
Description

Fig S2

Results of simulations based on 1,000 orthologue alignments, randomly sampled from OrthoMaM. Sequences were simulated over the phylogenetic trees corresponding to the sequence alignments, under a stop-extended codon model based on GY with empirical codon frequencies obtained from introns of the same genes. Only genes with at least 1000bp of introns were used (see Methods for details). Simulated values of φ were sampled uniformly between 0 and 3. Branch lengths of the phylogenies were re-estimated under a different model to that used for the simulation (MG F1x4), using codonPhyml. The stop-extended codon model, based also on MG F1x4 was then fitted to the simulated alignments. The figure shows the maximum likelihood estimates, plotted against simulated values of φ . The identity line is shown in red.

Code

```
#figure_sim_var: check bias in the estimate of phi
figure_sim_var = function(file) {
  sim = read.table(file,h=T)
  ggplot(sim,aes(Phi_sim,Phi)) + theme_bw(base_size = 20) +
geom_point(alpha=0.5) + geom_abline(slope=1,intercept=0,col="red") +
xlab(expression("Simulated"~phi)) + ylab(expression("Estimated"~phi))
}

#Fig S2: sim_emp_var
figure_sim_var('sim_empirical_variable.out')
```



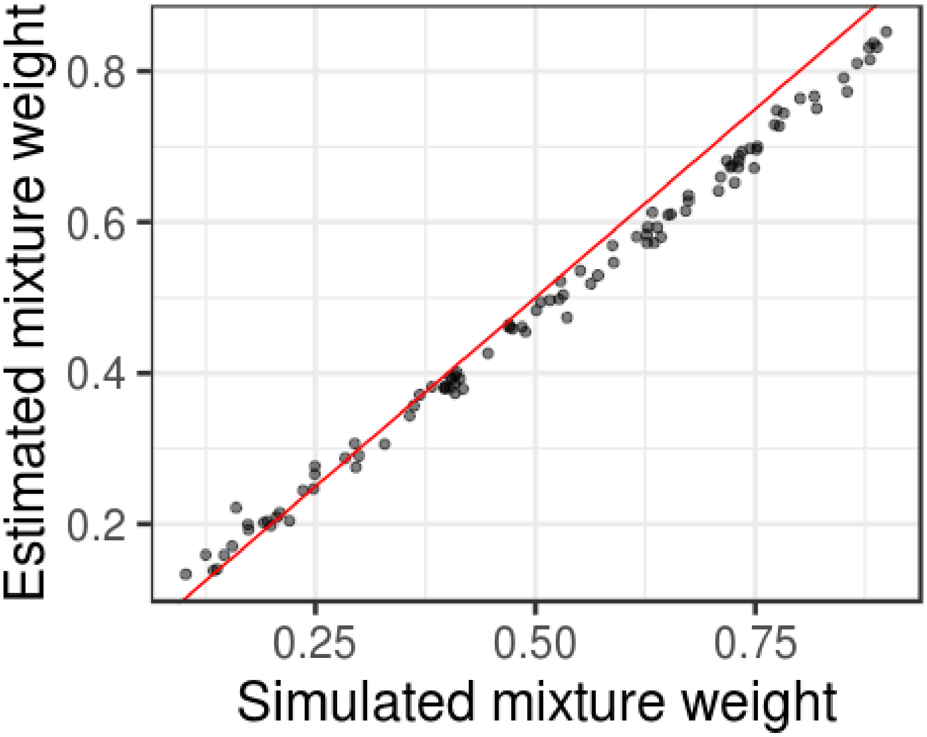
Description

Fig S3

Results of simulation to assess the accuracy and robustness to modeling assumptions of the estimate of the proportion of stop codons under selective constraint from the mixture model. Data was simulated under the GY model with empirical codon frequencies derived from intron sequences, as described in Methods. For each of 100 simulations, the proportion of genes under purifying selection was sampled uniformly between 0.1 and 0.8. The mixture model was fitted to the resulting alignments, under the MG model with the F1x4 model of codon frequencies. The figure shows the maximum likelihood value of the mixture weight corresponding to purifying selection, plotted against the proportion of alignments in the simulation for which the stop codon was under purifying selection. The identity line is shown in red. ***

Code

```
###########figure_sim_mix: check accuracy of mixture weight estimates
figure_sim_mix = function(file) {
  comp = read.table(file)
  colnames(comp) = c("Simulated","Estimated")
  ggplot(comp,aes(Simulated, Estimated)) + theme_bw(base_size = 20) +
geom_point(alpha=0.5) + geom_abline(slope=1,intercept=0,col="red") +
xlab("Simulated mixture weight") + ylab("Estimated mixture weight")
}

#Fig S3: sim_mix
figure_sim_mix('sim_mix.out')
```



***

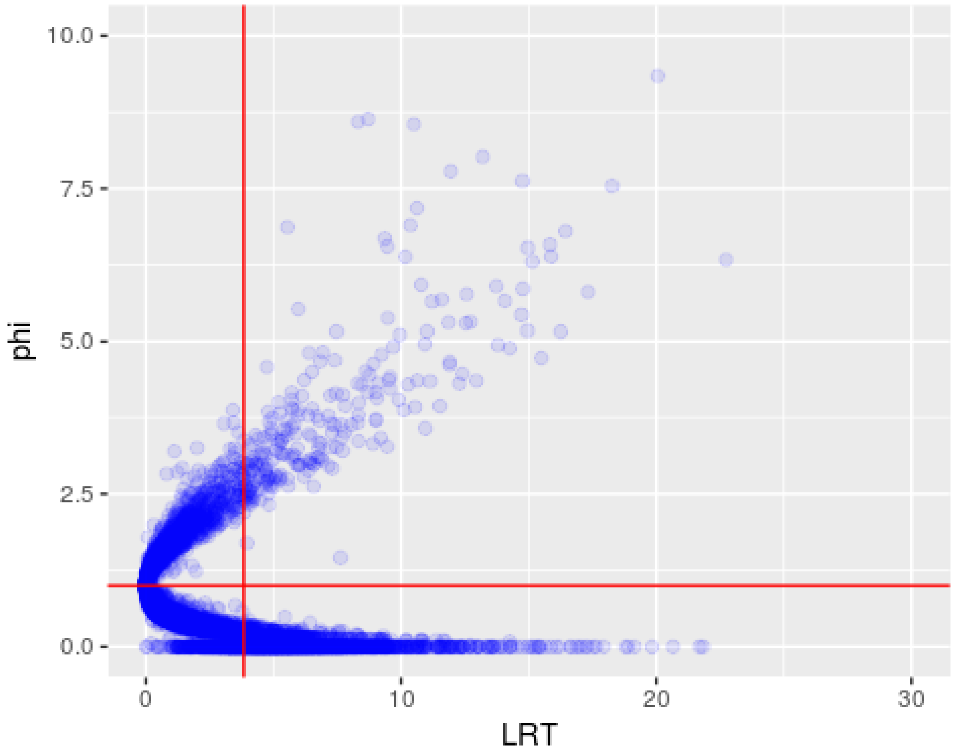
Description

Fig S4

Scatterplot showing the results of fitting the stop-extended model to the OrthoMaM alignments. The likelihood ratio test statistic ($2\Delta \ln L$) corresponding to the fit of the null model ($\varphi = 1$) compared to the alternative model (φ a free parameter) is plotted against the maximum likelihood estimate of φ . The horizontal line shows the critical value of the χ 2 distribution with one degree of freedom and the vertical line shows $\varphi = 1$.

Code

```
#Figure: phiscatter
figure_phiscatter = function() {
  t = cbind(t,2*t$deltaL)
  names(t)[ncol(t)] = "LRT"
  ggplot(t,aes(x=LRT,y=phi)) +
    geom_point(alpha=0.1,size=2,col="blue") +
geom_vline(xintercept=qchisq(0.95,1),col="red") + ylim(0,10) + xlim(0,30) +
    geom_hline(yintercept=1,col="red")
}

#Fig S4: phiscatter
figure_phiscatter()

## Warning: Removed 12 rows containing missing values (geom_point).
```



Description

Fig. S5: An example of a gene with stop codon evolving at (a) a low and (b) a high rate. The phylogeny is coloured according to the stop codon colour nomenclature as in figure 1. Taxa with gaps in the sequence alignment at the stop codon are indicated in grey. The genes corresponding to panels a and b are INSR and PARP1, respectively.

Code

```
#######Figure: Example_genes
figure_examplegenes = function() {
  return("Figure S5 was produced by hand from trees obtained from
Orthomam for INSR (a) and PARP1 (b)")
}

#Fig S5: example_genes
figure_examplegenes()

## [1] "Figure S5 was produced by hand from trees obtained from Orthomam
for INSR (a) and PARP1 (b)"
```

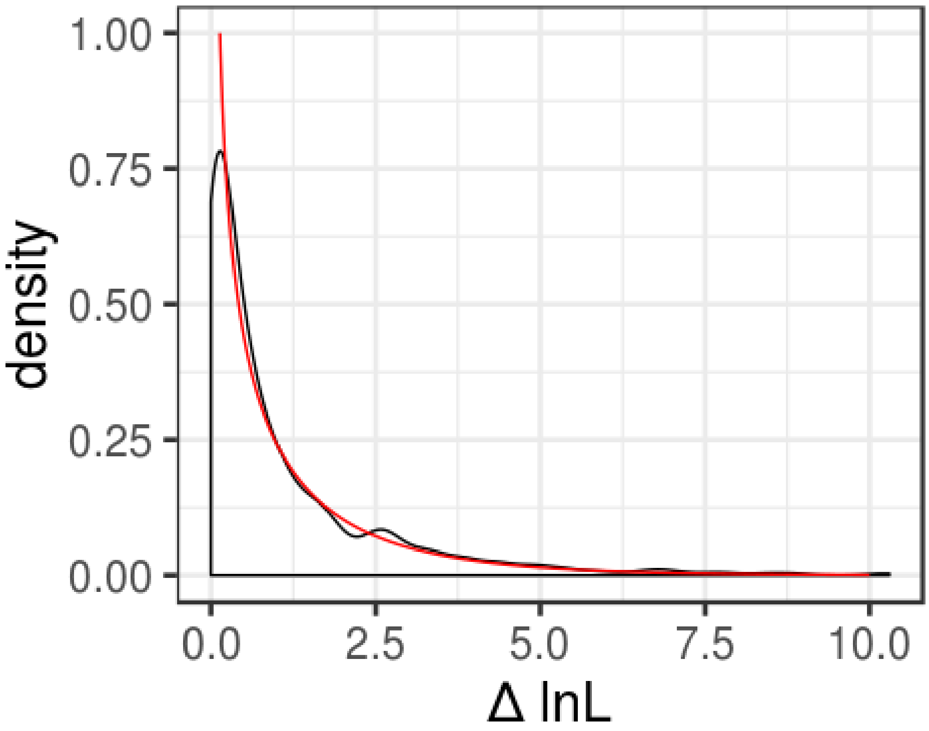
Description

Fig S6

Density plot of the likelihood ratio test statistic from simulations corresponding to figure S1. The red line shows the χ 2 distribution with one degree of freedom. ***

Code

```
#figure: sim_model and sim_empirical, check fit of chi-squared distribution to
the LRT statistic
figure_chifit = function(file,ylim) {
  sim = read.table(file,h=T)
  sim$Delta_lnL[sim$Delta_lnL < 0] = 0 # since we have already discovered a
higher likelihood solution with the null model, this maximum likelihood that
we have found with the alternative model can be set to this (if it is below it)
  x = seq(0,10,0.0001)
  sim$LRTstat = 2*sim$Delta_lnL
  ggplot() + theme_bw(base_size = 20) + geom_density(data = sim,
aes(LRTstat)) +xlab("\u0394 lnL") +
geom_line(data=data.frame(x=x,y=dchisq(x,1)), aes(x=x,y=y), col="red") +
ylim(c(0,ylim))
}
#Fig S6: sim_model
figure_chifit("sim_model.out",ylim=1)
```



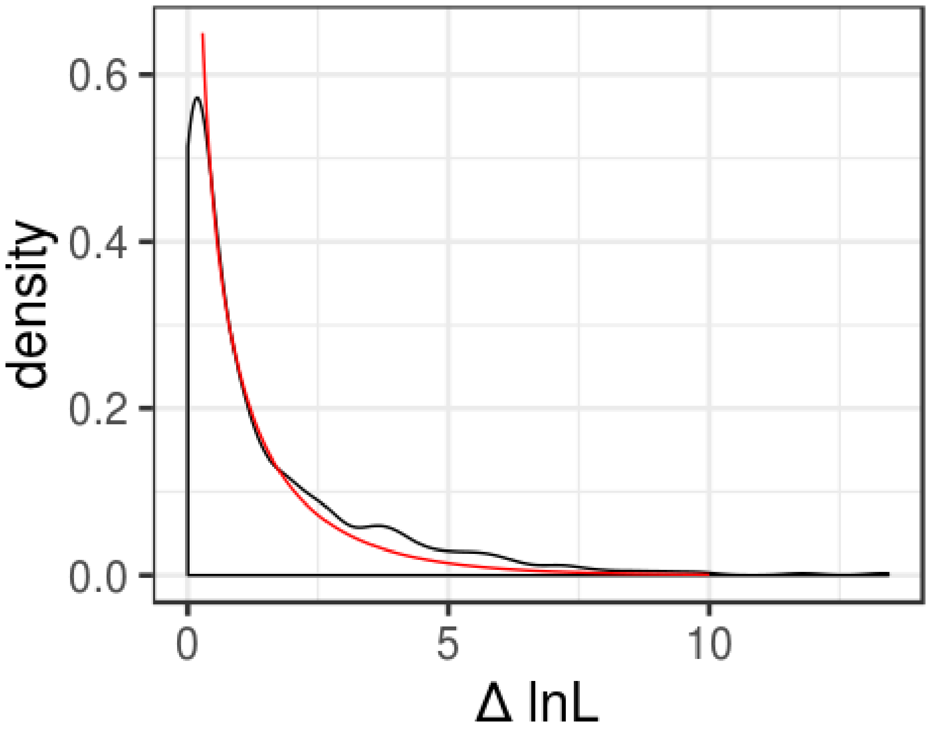
Description

Fig S7

Density plot of the likelihood ratio test statistic from simulations corresponding to figure S2. The red line shows the χ 2 distribution with one degree of freedom. ***

Code

```
#figure: sim_model and sim_empirical, check fit of chi-squared distribution to
the LRT statistic
figure_chifit = function(file,ylim) {
  sim = read.table(file,h=T)
  sim$Delta_lnL[sim$Delta_lnL < 0] = 0 # since we have already discovered a
higher likelihood solution with the null model, this maximum likelihood that
we have found with the alternative model can be set to this (if it is below it)
  x = seq(0,10,0.0001)
  sim$LRTstat = 2*sim$Delta_lnL
  ggplot() + theme_bw(base_size = 20) + geom_density(data = sim,
aes(LRTstat)) +xlab("\u0394 lnL") +
geom_line(data=data.frame(x=x,y=dchisq(x,1)), aes(x=x,y=y), col="red") +
ylim(c(0,ylim))
}
#Fig S7: sim_empirical
figure_chifit("sim_empirical.out",ylim=.65)
```



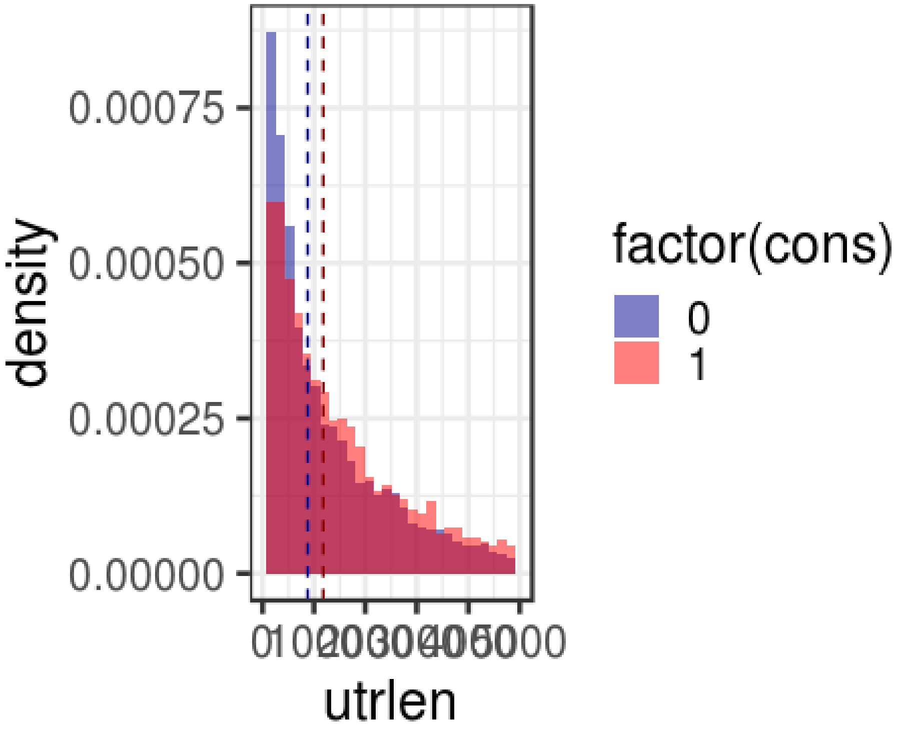
Description

Fig S8

Histogram of 3 0 UTR lengths for genes with conserved/non-conserved stop codons (based on statistical model comparison). The figure has been truncated at 5kb (the 8% of genes with 3 0 UTRs longer than this are not shown). Medians are shown as dashed vertical lines. ***

Code

```
#Figure S6: utr_hist - 3' UTR lengths histogram
figure_utrhist = function() {
  ggplot(t, aes(utrlen, fill = factor(cons))) +
    theme_bw(base_size = 20) + geom_histogram(alpha=0.5,aes(y =
..density..), position = 'identity') + scale_fill_manual(values=c("darkblue",
"red")) +

geom_vline(aes(xintercept=median(utrlen[cons==1],na.rm=T)),col="darkred", linetype="dashed") +

geom_vline(aes(xintercept=median(utrlen[cons==0],na.rm=T)),col="darkblue", linetype = "dashed") + xlim(0,5000) #+ xlab('3` UTR Length`)
}

#Fig S8: utr_hist
figure_utrhist()

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

## Warning: Removed 957 rows containing non-finite values (stat_bin).
```



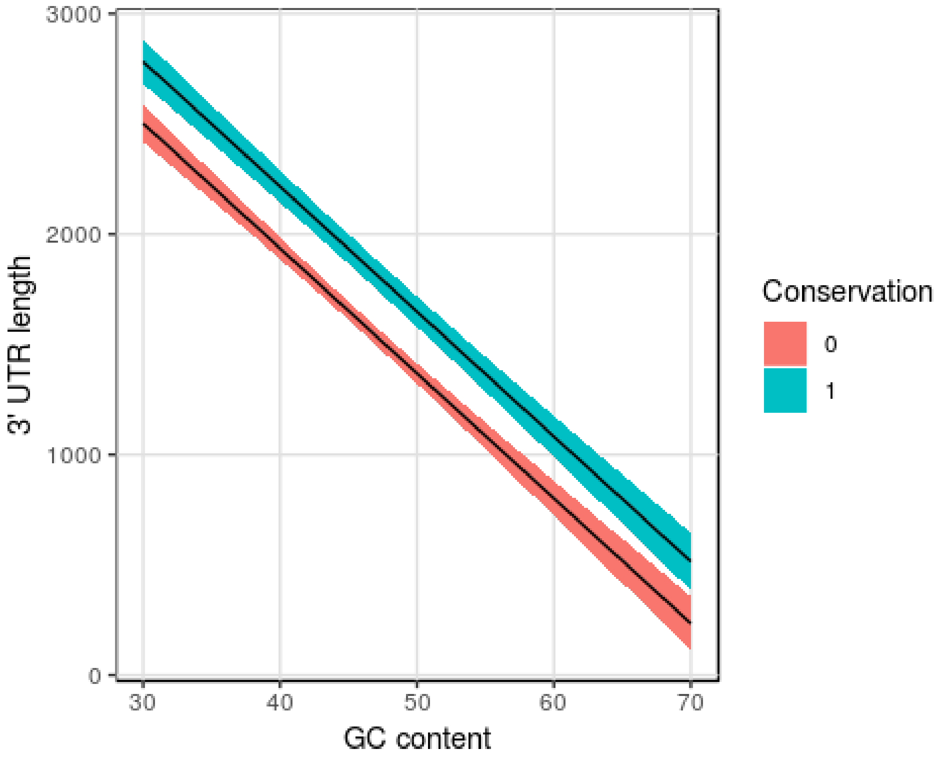
Description

Fig S9

Estimated 3 prime UTR length (and 95% confidence interval) as a function of GC-content for genes with conserved and non-conserved stop codons (based on statistical model comparison). The number of taxa for which the stop codon was positionally homologous with the end of the alignment was also included as a covariate in the model. ***

Code

```
#Figure: utrlen_model
# Plot utr length as a function of GC content and stop codon conservation
figure_utrlen_model = function() {
  utrlm = lm(utrlen ~ genegc + cons + taxcount, data=t)
  newdata = data.frame( genegc = rep(seq(from=30,to = 70,
length.out=100),2),cons =
c(rep(0,100),rep(1,100)),taxcount=rep(median(t$taxcount,na.rm=T),200))
  predictdata = cbind(newdata,predict(utrlm,newdata=newdata,se=T))
  predictdata = within(predictdata, {
    LL = fit-1.96*se.fit
    UL = fit + 1.96*se.fit
  })
  ggplot(predictdata,aes(x=genegc,y=fit)) +
geom_ribbon(aes(ymin=LL,ymax=UL,fill=factor(cons))) +
   geom_line(data=predictdata[predictdata$cons==1,],color="black")+
   geom_line(data=predictdata[predictdata$cons==0,],color="black")+
   xlab("GC content") + ylab("3\' UTR length") + theme(panel.background =
element_rect(fill= "white", colour="black"), panel.grid.minor =
element_blank(), panel.grid.major = element_line(colour="grey88"))+
   guides(fill=guide_legend(title="Conservation"))
}
#Fig S9: utrlen_model
figure_utrlen_model()
```



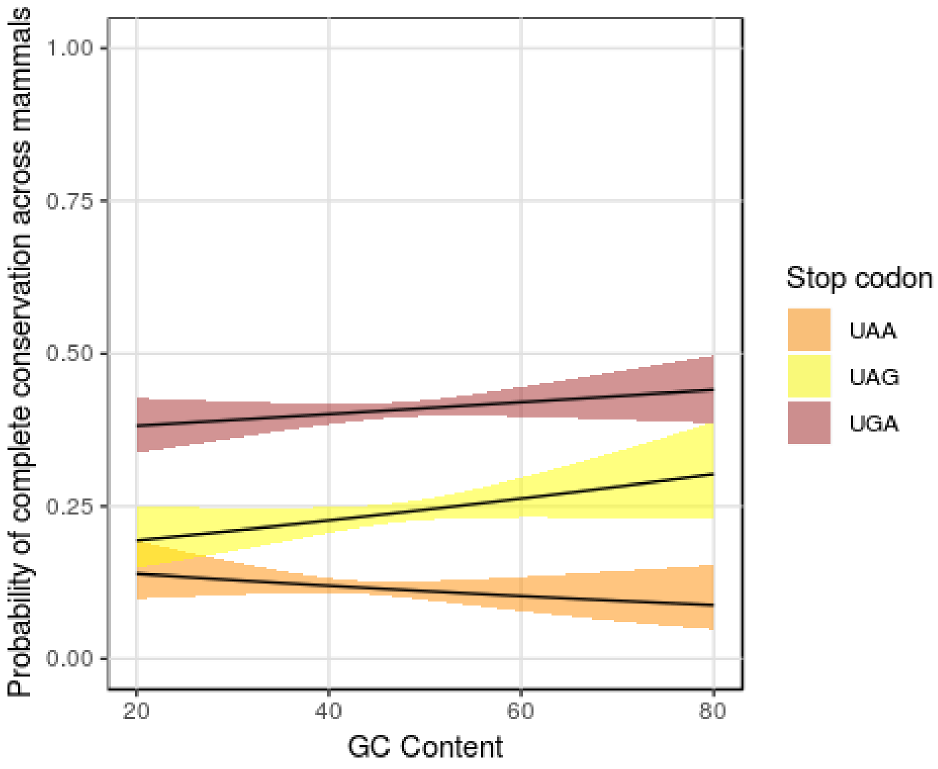
Description

Fig S10

Estimated probability of complete sequence conservation (and 95% confidence interval) as a function of GC-content. The plot is derived from a logistic regression model with interaction between stop codon and GC-content. This allows different slopes between the three lines, but no signifcant differences in the slope were observed. The number of taxa for which the stop codon was positionally homologous with the end of the alignment was included as a covariate in the model.
***

Code

```
#####Figure: complete_gc_logistic
# Plot 'conserved' (i.e. the completely conserved stop codons) as a function
of GC content, separately for the 3 stop codons
# Expectation is that this should be a strong function of GC content if stop
codon evolution is primarily neutral
figure_complete_gc_logistic = function() {
gcglm = glm(conserved ~ genegc * humstop + taxcount ,family="binomial",
data=t)
newdata = data.frame( genegc = rep(seq(from=20,to = 80,
length.out=100),3),humstop =
c(rep('UAG',100),rep('UGA',100),rep('UAA',100)),taxcount=rep(median(t$taxc
ount,na.rm=T),300))
predictdata = cbind(newdata,predict(gcglm,newdata=newdata,se=T))
predictdata = within(predictdata, { predictProb = plogis(fit)
LL = plogis(fit-1.96*se.fit)
UL = plogis(fit + 1.96*se.fit)
})

ggplot(predictdata,aes(x=genegc,y=predictProb)) +
   geom_ribbon(aes(ymin=LL,ymax=UL,fill=factor(humstop)),alpha=.5) +

geom_line(data=predictdata[predictdata$humstop=='UAA',],color="black")+

geom_line(data=predictdata[predictdata$humstop=='UAG',],color="black")+

geom_line(data=predictdata[predictdata$humstop=='UGA',],color="black")+
   xlab("GC Content") + ylab("Probability of complete conservation across
mammals") +
   theme(panel.background = element_rect(fill= "white", colour="black"),
panel.grid.minor = element_blank(), panel.grid.major =
element_line(colour="grey88"))+
   guides(fill=guide_legend(title="Stop codon")) + ylim(0,1) +
   scale_fill_manual(values=c("darkorange","yellow","brown"))
}

#Fig S10: complete_gc_logistic
figure_complete_gc_logistic()
```



Description

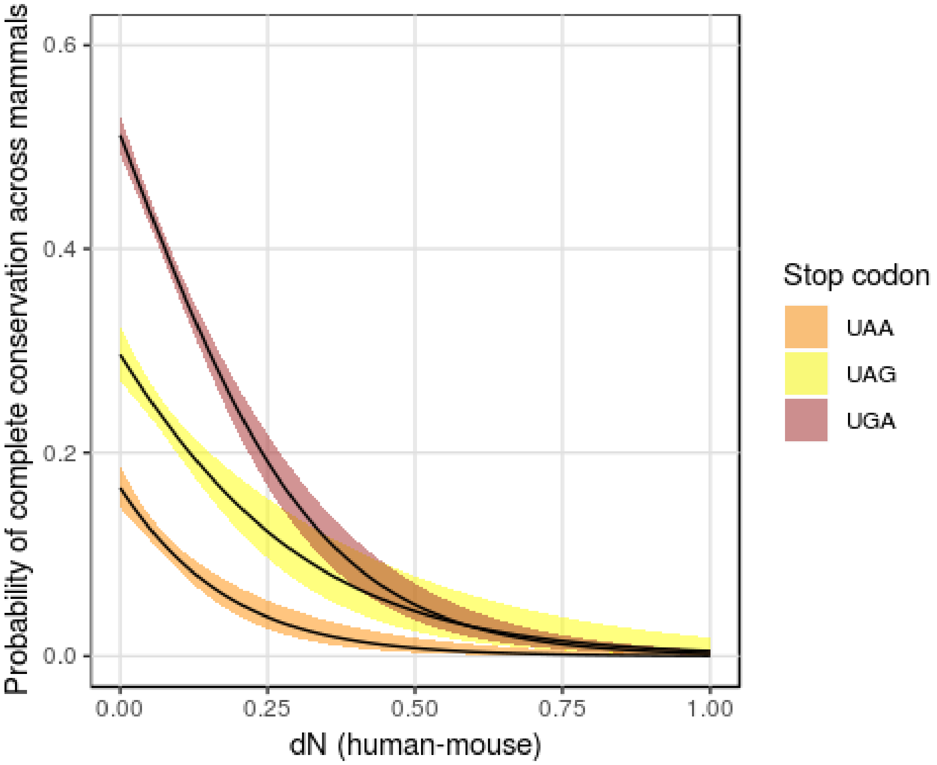
Fig S11a Fig S11b Fig S11d Fig S11d

Estimated probability of complete sequence conservation (and 95% confidence interval) as a function of (a) dN between human and mouse, (b) dS between human and mouse, (c) dN between human and macaque and (d) dS between human and macaque. In all cases the x-axis is truncated to 1, as most of the divergence values are lower than this. The number of taxa for which the stop codon was positionally homologous with the end of the alignment was included as a covariate in the model. ***

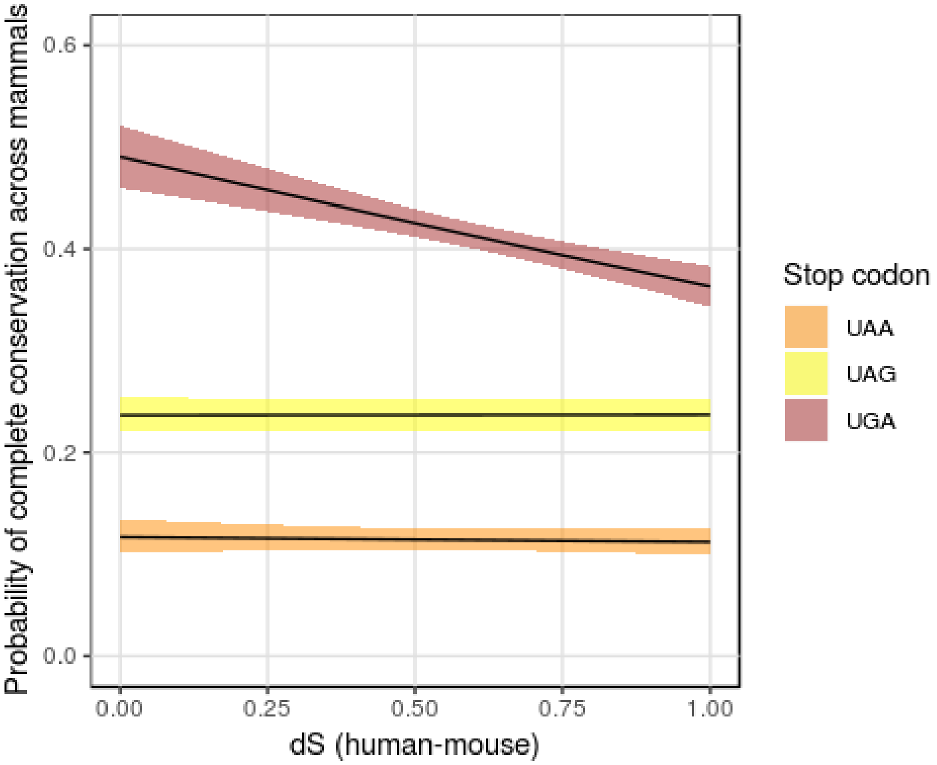
Code

```
#Figure: complete_dNdS
figure_complete_dNdS = function(panel) {
ds = read.table("ensembl_mus",row.names=1)
colnames(ds) = c("musds","musdn")
t = cbind(t,ds[rownames(t),])
ds = read.table("ensembl_macaque",row.names=1)
colnames(ds) = c("macds","macdn")
t = cbind(t,ds[rownames(t),])
dnds_ylim = 0.6
vblnames = list(a='musdn',b='musds',c='macdn',d='macds')
```

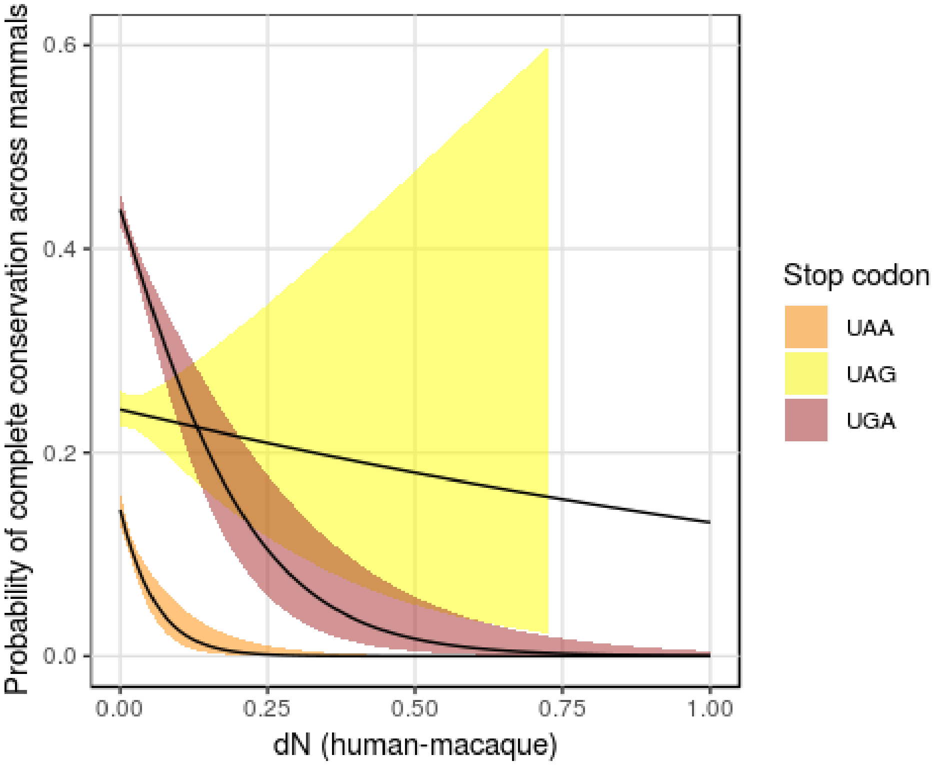
```r
t$x = t[[vblnames[[panel]]]]
myglm = glm(conserved ~ x * humstop + taxcount ,family="binomial",
data=t)
newdata = data.frame( x = rep(seq(from=0,to = 1,
length.out=100),3),humstop =
c(rep('UAG',100),rep('UGA',100),rep('UAA',100)),taxcount=rep(median(t$taxc
ount,na.rm=T),300))
xlablist = list(a='dN (human-mouse)',b='dS (human-mouse)',c='dN (human-
macaque)',d='dS (human-macaque)')
xlab = xlablist[[panel]]
predictdata = cbind(newdata,predict(myglm,newdata=newdata,se=T))
predictdata = within(predictdata, { predictProb = plogis(fit)
LL = plogis(fit-1.96*se.fit)
UL = plogis(fit + 1.96*se.fit)
})
ggplot(predictdata,aes(x=x,y=predictProb)) +
  geom_ribbon(aes(ymin=LL,ymax=UL,fill=factor(humstop)),alpha=.5) +
  geom_line(data=predictdata[predictdata$humstop=='UAA',],color="black")
+
  geom_line(data=predictdata[predictdata$humstop=='UAG',],color="black")
+
  geom_line(data=predictdata[predictdata$humstop=='UGA',],color="black")
+
  xlab(xlab) + ylab("Probability of complete conservation across mammals") +

  theme(panel.background = element_rect(fill= "white", colour="black"),
panel.grid.minor = element_blank(), panel.grid.major =
element_line(colour="grey88"))+
  guides(fill=guide_legend(title="Stop codon")) + ylim(0,dnds_ylim) +
  scale_fill_manual(values=c("darkorange","yellow","brown"))
}

#Fig S11: complete_dNdS
figure_complete_dNdS('a')
```



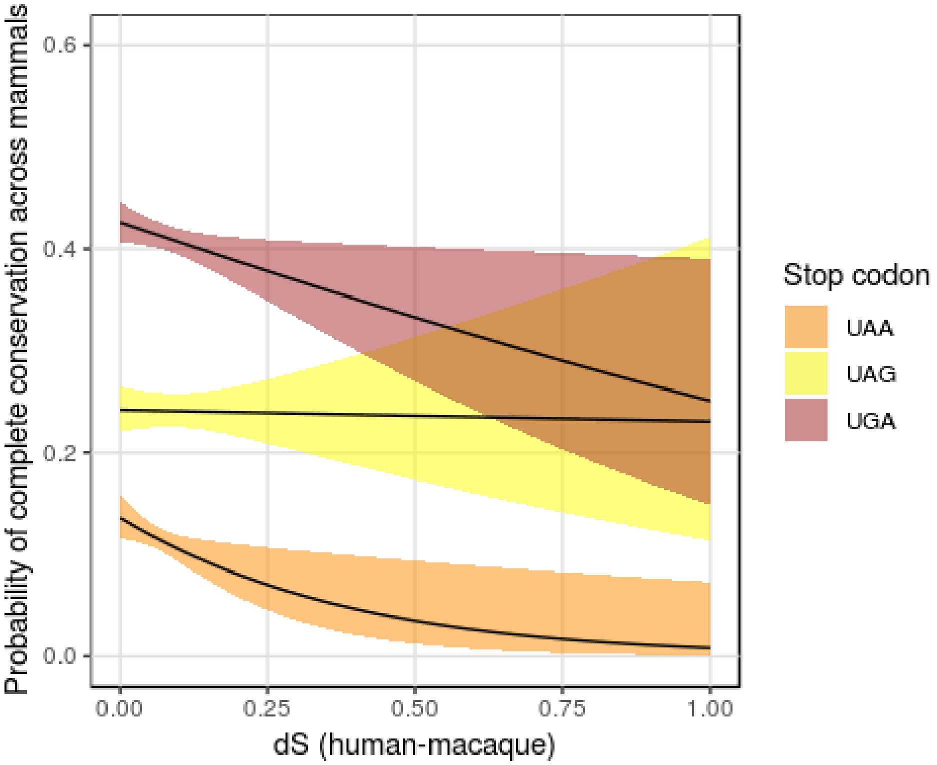
```
figure_complete_dNdS('b')
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```



```
figure_complete_dNdS('c')
```



```
figure_complete_dNdS('d')
```



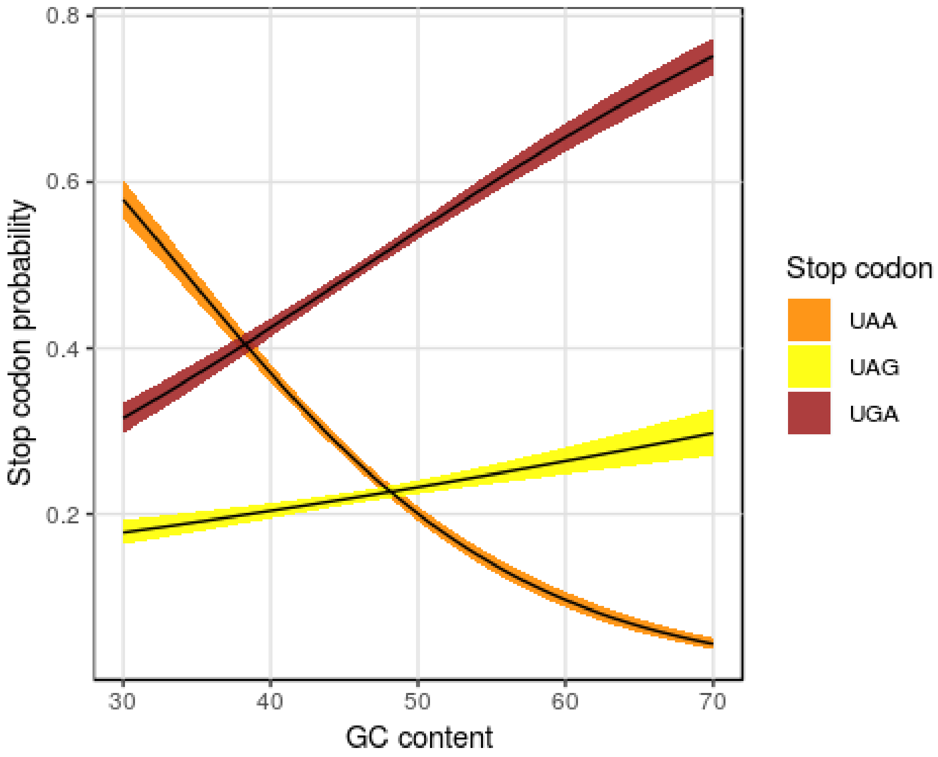
Description

Fig S12

Frequency of each stop codon in human protein-coding genes, as a function of GC-content. ***

Code

```
#Figure: GCcontent
# Plot stop codon probability as a function of genegc
figure_GCcontent = function() {
  x = rep(0,nrow(t))
  x[t$humstop=='UAA'] = 1
  t$istaa = x
  x = rep(0,nrow(t))
  x[t$humstop=='UAG'] = 1
  t$istag = x
  x = rep(0,nrow(t))
  x[t$humstop=='UGA'] = 1
  t$istga = x

  newdata = data.frame( genegc = seq(from=30,to = 70, length.out=100))

  myglm1 = glm(istaa ~ genegc,family="binomial", data=t)
  myglm2 = glm(istag ~ genegc,family="binomial", data=t)
  myglm3 = glm(istga ~ genegc,family="binomial", data=t)
  predictdata = cbind(newdata,predict(myglm1,newdata=newdata,se=T))
  x = cbind(newdata,predict(myglm2,newdata=newdata,se=T))
  predictdata = rbind(predictdata,x)
  x = cbind(newdata,predict(myglm3,newdata=newdata,se=T))
  predictdata = rbind(predictdata,x)
  predictdata = within(predictdata, { predictProb = plogis(fit)
  LL = plogis(fit-1.96*se.fit)
  UL = plogis(fit + 1.96*se.fit)
  })

  humstop = c(rep('UAA',100),rep('UAG',100),rep('UGA',100))
  predictdata = cbind(predictdata,humstop)


  ggplot(predictdata,aes(x=genegc,y=predictProb)) +
geom_ribbon(aes(ymin=LL,ymax=UL,fill=factor(humstop)),alpha=.9) +

geom_line(data=predictdata[predictdata$humstop=='UAA',],color="black")+

geom_line(data=predictdata[predictdata$humstop=='UAG',],color="black")+

geom_line(data=predictdata[predictdata$humstop=='UGA',],color="black")+
  xlab("GC content") + ylab("Stop codon probability") +
theme(panel.background = element_rect(fill= "white", colour="black"),
panel.grid.minor = element_blank(), panel.grid.major =
element_line(colour="grey88"))+
  guides(fill=guide_legend(title="Stop codon")) +
  scale_fill_manual(values=c("darkorange","yellow","brown"))


}

#Fig S12: GCcontent
figure_GCcontent()
```



Description

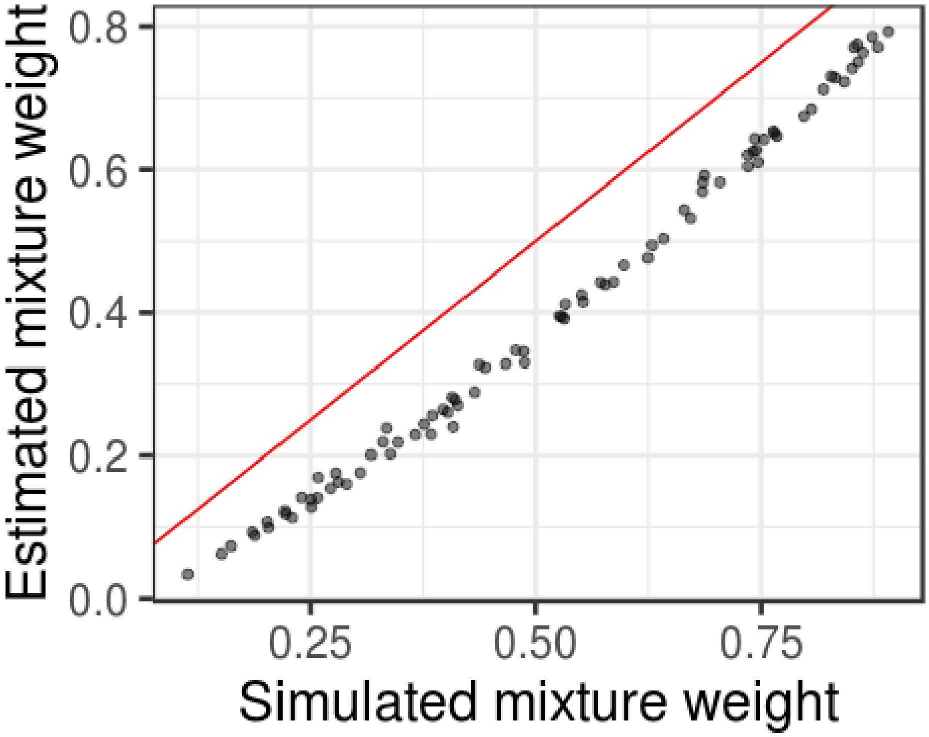
Fig S13

The same simulations as in figure S3 but in this case the GY model with the F3x4 model of codon frequencies was used. ***

Code

```
#figure_sim_mix: check accuracy of mixture weight estimates
figure_sim_mix = function(file) {
 comp = read.table(file)
 colnames(comp) = c("Simulated","Estimated")
```

```
  ggplot(comp,aes(Simulated, Estimated)) + theme_bw(base_size = 20) +
geom_point(alpha=0.5) + geom_abline(slope=1,intercept=0,col="red") +
xlab("Simulated mixture weight") + ylab("Estimated mixture weight")
}
figure_sim_mix('sim_mix_gy.out')
```



Authors

- **Cathal Seoighe**
- **Stephen J. Kiniry**
- **Andrew Peters**
- **Pavel V. Baranov**
- **Haixuan Yang**