RPHAST: detecting GC-biased gene conversion

M. J. Hubisz, K. S. Pollard, and A. Siepel

March 28, 2013

1 Introduction

This vignette describes some of the basic functions available for detecting GC-biased gene conversion (gBGC) using the RPHASTpackage. gBGC is a process in which GC/AT (strong/weak) heterozygotes are preferentially resolved to the strong allele during gene conversion. This confers an advantage to G and C alleles that mimics positive selection, without conferring any known functional advantage. Therefore, some regions of the genome identified to be under positive selection may be better explained by gBGC. gBGC has also been hypothesized to be an important contributor to variation in GC content and to the fixation of deleterious mutations.

For more information on gBGC, we suggest the following review: Duret and Galtier (2009).

In section 2, we will describe how to perform nucleotide tests for gBGC and selection. For background on these tests, please see: Kostka et˜al. (2012).

In section 3, we will describe how to perform tests for gBGC using a phylo-HMM. The background and method behind this model is described here: Capra et˜al. (2013).

2 Basic nucleotide model

Here we show how to set up an evolutionary model with a 4x4 transition matrix that incorporates gBGC. The idea is to start with a standard model of evolution, such as Jukes Cantor, HKY, or REV, but use an additional gBGC parameter B that affects the rates of strong-to-weak and weak-to-strong mutations on a particular branch of interest. We can also add a selection parameter S to this branch which models positive ($S > 0$) or negative ($S < 0$) selection. Given an alignment, we can estimate B, S, or both together, and assess the evidence of gBGC and/or selection using likelihood ratio tests. See Kostka et˜al. (2012) for a full description of these models and tests. We will take an example from that paper and analyze HAR1.

The first step is to get a neutral model of evolution. For the HAR analysis we obtained a neutral model based on fourfold-degenerate (4d) sites in ENCODE regions, and then re-scaled the branches for each HAR according to the mutation rate in noncoding regions surrounding the HAR. There are examples showing how to pull out a region of an alignment and estimate a model based on 4d sites in RPHAST's vignette 1. Here we will assume they have already been computed. The data for HAR1 and its neutral model are stored in our package of example files. They can be loaded like with the following commands:

```
> require("rphast")
> exampleArchive <- system.file("extdata", "examples.zip", package="rphast")
> unzip(exampleArchive, c("HAR_001_neutral_SSREV.mod", "HAR_001.fa"))
> neutralMod <- read.tm("HAR_001_neutral_SSREV.mod")
> align <- read.msa("HAR_001.fa")
```

Four different models are described by Kostka et˜al. (2012):

1. **null**: This model has a single free parameter, called the "selection parameter" (or sel.global), which scales the entire rate matrix relative to the neutral rates.

2. **selection only**: This model has two free parameters. sel.global acts as in the null model. An additional selection parameter, sel.ls (for lineage-specific selection), acts on a branch of interest. It is additive with sel.global, so that the total selection on the branch of interest is sel.global + sel.ls.

3. **gBGC only**: This model has two free parameters. sel.global acts as in the null model, affecting all branches. A parameter called bgc.ls models the effect of gBGC on a branch of interest. bgc.ls is usually constrained to be ≥ 0, since there is no biological concept of a negative gBGC effect.

4. **gBGC + selection**: This is the full model, with all three free parameters as described above: sel.global, sel.ls, and bgc.ls.

A single function will get the maximum likelihood estimates for these parameters in all four models:

```
> nuc.results <- bgc.nucleotide.tests(align, neutralMod, "hg18")
> nuc.results

        likelihood  sel.global sel.ls bgc.ls
null     -364.6712 1.456037942      0      0
sel      -302.2111 0.016192886    200      0
bgc      -297.3033 0.021113519      0    200
sel+bgc  -291.5415 0.007461312    200    200
```

The third argument tells RPHASTwhich branch of the tree to test for selection and/or gBGC. This function returns a data frame with a row for each model, giving the maximum likelihood and the corresponding parameter estimates. Note that the estimates for sel.ls (lineage-specific selection) and sel.bgc are both 200 in the models where they are not constrained to be zero. This is because there are so many strong mutations on the human branch of HAR1 that the parameters are being pushed to their upper boundary, which has a default of 200. We can investigate this:

```
> classify.muts.bgc(align, neutralMod, branch="hg18")

     branch   W.to.S       S.to.W       W.to.W       S.to.S
hg18   hg18 13.98254 4.947062e-09 1.619623e-06 6.282675e-10
```

The classify.muts.bgc function counts the expected number of each type of mutation (weak to strong, strong to weak, weak to weak, strong to strong), given the observed nucleotides at the leaf nodes, and a neutral model. Here, it indicates that there are about 14 weak to strong mutations expected on the human branch, which is quite a lot given the low counts in other category, and that the alignment only has

```
> ncol.msa(align)

[1] 106
```

106 columns in the alignment.

We can adjust the boundaries:

```
> bgc.nucleotide.tests(align, neutralMod, "hg18", bgc.limits=c(0, 2000), sel.limits=c(-2000,2000))

        likelihood  sel.global    sel.ls   bgc.ls
null     -364.6712 1.456037942   0.00000    0.000
sel      -299.1694 0.005093476 423.76721    0.000
bgc      -282.5656 0.001752459   0.00000 1296.724
sel+bgc  -282.5557 0.002397168 -47.30931 1344.165
```

though it is important to keep in mind that if the boundaries are pushed too high, in some cases the function may quit with an error due to numerical problems computing/exponentiating the rate matrix.

2.1 Classifying the alignment

The likelihoods obtained by `bgc.nucleotide.tests` can be compared to each other to obtain likelihood ratios. The likelihood ratios can be compared to a null distribution to assess the evidence for selection and gBGC on the branch of interest.

In the paper, seven different likelihood ratios were computed:

```
> nullLike <- nuc.results["null", "likelihood"]
> selLike <- nuc.results["sel", "likelihood"]
> bgcLike <- nuc.results["bgc", "likelihood"]
> bgcSelLike <- nuc.results["sel+bgc", "likelihood"]
> lrs <- c(selLike - nullLike,
+          bgcLike - nullLike,
+          selLike - bgcLike,
+          bgcLike - selLike,
+          bgcSelLike - bgcLike,
+          bgcSelLike - selLike)
> lrs

[1] 62.460162 67.367943 -4.907781  4.907781  5.761773 10.669554
```

Now that we have all the likelihood ratios, the difficult part is figuring out the significance and choosing the best model. In Kostka et˜al. (2012), null distributions were obtained by simulation separately for each HAR, so that each of the seven likelihood ratios could be declared as significant or not. Given the significance of each of the seven tests, a series of rules was used to assign the HAR to a category. See the paper for more details. In the case of HAR1, the best fit is the model with gBGC but no selection. Therefore, HAR1 is suspected to be a "false positive" HAR: the burst of substitutions observed on the human lineage is likely a result of gBGC and not of functional importance.

3 gBGC Hidden Markov Model

In this section we'll show how to set up a Hidden Markov Model (HMM) with four states: conserved, neutral, conserved with gBGC, and neutral with gBGC. As always, the first step is to obtain an alignment and a model of neutral evolution. Let's use a 4-species alignment of a 100 kilobase chunk of chromosome 1, and UCSC's neutral model for autosomes which was created for the 44-way alignment displayed on the hg18 browser:

```
> unzip(exampleArchive, c("chr1_100k_4way.maf", "placentalMammals.mod"))
> align <- read.msa("chr1_100k_4way.maf")
> mod <- read.tm("placentalMammals.mod")
> mod$tree <- prune.tree(mod$tree, names(align), all.but=TRUE)
```

The function `phastBias` can then be used to set up the HMM and obtain gBGC tract predictions and posterior probabilities for each state. However, there are many choices to make with regard to which parameters will be estimated. There are several relevant parameters:

- B: the level of gBGC on the foreground branch
- scale: an overall scaling factor for the tree in all models
- ρ: a scaling factor for the branches in conserved models
- bgc.in: the transition rate into the gBGC state
- bgc.out: the transition rate out of the gBGC state

- cons.in: the transition rate into the conserved state
- cons.out: the transition rate out of the conserved state

It is important not to try to estimate too many parameters; in the paper the only parameter we optimized was bgc.out. We tried various values of B and chose $B = 3$, which was the lowest value we could use that did not produce many false positives in simulations. We started with a neutral model and then held scale to 1.0. We set $\rho = 0.31$, as this has been found to be a fairly robust estimate for the level of conservation within conserved elements across several different species sets.

The default `phastBias` arguments are set to the values used Capra et˜al. (2013). However, different topologies or neutral models may perform better with different parameter settings. The choice for the parameter B is particularly important and may need some tuning. Higher values of B will give fewer, shorter, higher-confidence gBGC tracts, whereas lower values of B will yield a more inclusive but lower-confidence set of tracts. If B is too low, the HMM will not be able to distinguish between the gBGC and non-gBGC states, and may predict the entire alignment to be a "gBGC tract". The other parameters seem more forgiving and in most cases should not have to be adjusted.

Let's run `phastBias` on our alignment and examine the results:

```
> hmm.results <- phastBias(align, mod, foreground="hg18")
> hmm.results

phastBias results: list with the following elements:
  foreground: branch tested for gBGC
     value:  hg18
  likelihood: total posterior likelihood
     value: -282593.657755
  bgc: gBGC strength parameter B
     value: 3.000000
  bgc.in: rate into gBGC state
     value: 0.000018
  bgc.out: rate out of gBGC state
     value: 0.001000
  mu: rate out of conserved state
     value: 0.022222
  nu: rate into conserved state
     value: 0.009524
  scale: overall tree scale
     value: 1.000000
  rho: conserved state tree scale
     value: 0.310000
  tracts: features object with gBGC tracts
     value: features object with 1 rows covering 1471 bases
  post.prob: posterior probability for each state in each column
     value: data.frame with 100000 rows and 5 columns
  not.informative: features object with regions of alignment not informative for gBGC on foreground bra
     value: data.frame with 232 rows and 7 columns
```

phastBias returns an R list object containing the likelihood, all the final parameter values, posterior probabilities for each state at every site, and predicted gBGC tracts. The tracts are obtained by thresholding the posterior probability that each site is in one of the gBGC states at 0.5. In this example, only one tract is predicted:

```
> hmm.results$tracts
```

```
Features object
hg18        phastBias        gBGC_tract        1082696        1084166        1281.756        .        .
```

```
> coverage.feat(hmm.results$tracts)
```

```
[1] 1471
```

The alignment in this region can be extracted and analyzed:

```
> tractAlign <- split.by.feature.msa(align, hmm.results$tracts)[[1]]
> classify.muts.bgc(tractAlign, mod, "hg18")
```

```
     branch   W.to.S   S.to.W   W.to.W    S.to.S
hg18   hg18 16.40768 2.259461 7.027347 0.3196058
```

```
> classify.muts.bgc(align, mod, "hg18")
```

```
     branch   W.to.S   S.to.W   W.to.W   S.to.S
hg18   hg18 414.0602 573.2176 113.0958 49.16901
```

So we can see that there is indeed an excess of W→S mutations on the human branch within the tract, but not within the entire alignment.

We can also create a plot of the posteriors and the tract:

```
> tracks <- list(as.track.feat(hmm.results$tracts, name="gBGC tracts"),
+                as.track.wig(score=(hmm.results$post.prob[,"gBGC_neutral"] +
+                                    hmm.results$post.prob[,"gBGC_conserved"]),
+                             name="gBGC posterior",
+                             coord=hmm.results$post.prob$coord),
+                as.track.feat(hmm.results$not.informative, name="not informative"))
> plot.track(tracks)
>
```

The plot produced by this code can be seen in Figure 1.

We can also manually inspect the alignment to see the W→S changes on the human branch; however the display only works for alignments of about a 100 bases (more if the display is wider). So, we cannot see the entire alignment at once, but we can see a part of it:

```
> plot.msa(tractAlign, xlim=c(1083115, 1083175), pretty=TRUE)
```

The region plotted here contains four W→S mutations on the human lineage in a 60 bp stretch and is shown in figure 2.

3.1 Non-informative columns

The regions labeled in the `phastBias` results as "not informative" for gBGC require some explanation. A column may be uninformative for gBGC if it has missing data in certain leaf nodes of the tree. In this example, there must be non-missing data in the human, chimp, and at least one outgroup for a substitution to be unambiguously assigned to the human branch. `phastBias` does not allow evidence for gBGC to accumulate at columns where there is insufficient information. Otherwise, the phylo-HMM tends to assign these regions to the gBGC state, because without information to differentiate the foreground and background branches, the W→S substitutions can be assigned to the foreground branch, and the S→W substitutions to a background branch. Therefore, regions that are not informative for gBGC are essentially masked (by replacing the gBGC states with non-gBGC states at these sites). These sites may still be assigned to gBGC tracts, but only due to evidence in informative surrounding sites. `phastBias` returns a features object annotating non-informative regions so that the user can be aware of how many sites did not contribute to the final result.

4 Appendix: nucleotide model details

Here we show in detail how the bgc.nucleotide.tests function works. It may shed some light on how models are set up and manipulated in RPHAST, and how you might customize your own tests.

Let's first re-load the data from HAR1:

```
> align <- read.msa("HAR_001.fa")
> neutralMod <- read.tm("HAR_001_neutral_SSREV.mod")
> neutralMod

ALPHABET: A C G T
ORDER: 0
SUBST_MOD: SSREV
BACKGROUND: 0.344340 0.155660 0.155660 0.344340
RATE_MAT:
  -0.841926    0.150247    0.347464    0.344214
   0.332367   -1.349681    0.248678    0.768636
   0.768636    0.248678   -1.349681    0.332367
   0.344214    0.347464    0.150247   -0.841926
TREE: (((((((hg18:0.000346,panTro2:0.000472):0.001592,rheMac2:0.00197):0.006272,(rn4:0.005314,mm8:0.0047
```

The null model consists of the neutral model, with a selection parameter that applies to the entire tree. This selection parameter can be added to the neutral model like so:

```
> neutralMod$selection <- 0
```

If neutralMod$selection is NULL, then it is implicitly zero, but will not be considered a model parameter and will not be optimized by phyloFit. If it is not NULL, then phyloFit will optimize it, unless the "sel" parameter is specified as part of the no.opt argument to phyloFit. So, we can optimize the selection parameter like so:

```
> nullMod <- phyloFit(align, init.mod=neutralMod, no.opt=c("backgd", "branches", "ratematrix"))
> nullMod

ALPHABET: A C G T
ORDER: 0
SUBST_MOD: SSREV
SELECTION_PAR: 1.456038
TRAINING_LNL: -364.671214
BACKGROUND: 0.344340 0.155660 0.155660 0.344340
RATE_MAT:
  -1.598604    0.285281    0.659747    0.653576
   0.631079   -2.562701    0.472177    1.459445
   1.459445    0.472177   -2.562701    0.631079
   0.653576    0.659747    0.285281   -1.598604
TREE: (((((((hg18:0.000346,panTro2:0.000472):0.001592,rheMac2:0.00197):0.006272,(rn4:0.005314,mm8:0.0047
```

Note that we tell phyloFit not to optimize the background frequencies, the branch lengths, or the rate matrix parameters. The selection parameter is therefore the only parameter being optimized. In this model, the selection parameter is equivalent to re-scaling the branch lengths, but it is parameterized differently, so that it can interact with the gBGC parameter in later models. You can see the difference between re-scaling the tree:

```
> phyloFit(align, init.mod=neutralMod, scale.only=TRUE, no.opt=c("backgd", "ratematrix", "sel"))
```

```
ALPHABET: A C G T
ORDER: 0
SUBST_MOD: SSREV
SELECTION_PAR: 0.000000
TRAINING_LNL: -364.671213
BACKGROUND: 0.344340 0.155660 0.155660 0.344340
RATE_MAT:
  -0.841926    0.150247    0.347464    0.344214
   0.332366   -1.349680    0.248678    0.768636
   0.768636    0.248678   -1.349680    0.332366
   0.344214    0.347464    0.150247   -0.841926
TREE: (((((((hg18:0.000656854,panTro2:0.000896055):0.00302229,rheMac2:0.00373989):0.0119069,(rn4:0.01008
```

Note that this gives the same likelihood as in nullMod, but this time the branch lengths have changed. It is about the same as rescaling the original tree using the selection parameter estimated above like so:

```
> rescale.tree(neutralMod$tree, nullMod$selection/(1-exp(-nullMod$selection)))

[1] "(((((((hg18:0.000656966,panTro2:0.000896208):0.0030228,rheMac2:0.00374053):0.0119089,(rn4:0.010089
```

So now we have the null model stored in nullMod. We can get the likelihood in a couple of ways:

```
> nullMod$likelihood

[1] -364.6712

> likelihood.msa(align, nullMod)

[1] -364.6712
```

and we can see what the estimate for the selection parameter is:

```
> nullMod$selection

[1] 1.456038
```

Next we want to estimate the model with selection added onto the foreground branch. In this case we will use the human branch. To do this, we need to modify the initial model so that there is a separate model on the human branch. This is referred to as an "lineage-specific" (LS) model in RPHAST. An LS model can specify an entirely different substitution model (such as REV, HKY85, JC, etc), or it can specify certain parameters which should be estimated separately for this model. In this case we want the selection parameter to be estimated separately on the human branch than the rest of the tree. We do this like so:

```
> initSelMod <- add.ls.mod(neutralMod, "hg18", separate.params="sel")
```

This adds a separate selection parameter to the human branch. The initial selection parameter is 0, though we could have specified a different initial value like this:

```
> initSelMod2 <- add.ls.mod(neutralMod, "hg18", separate.params="sel", selection=2)
```

Now that the initial model is set up, we can use phyloFit to get the maximum likelihood estimate for both selection parameters (global and hg18):

```
> selMod <- phyloFit(align, init.mod=initSelMod, no.opt=c("backgd", "branches", "ratematrix"))
> # print the likelihood:
> selMod$likelihood
```

```
[1] -302.2111

> # print the global selection parameter:
> selMod$selection

[1] 0.01619289

> # print the human selection parameter:
> selMod$ls.mod$selection

[1] 200
```

Note that when selection is defined in the main model as well as in an LS model, the total selection on the LS model is the sum of the selection parameters in the LS and main models. You can check by applying the selection parameter to the rate matrix manually:

```
> apply.bgc.sel(neutralMod$rate.matrix, sel=(selMod$selection+selMod$ls.mod$selection)) -
+   selMod$ls.model$rate.matrix

           [,1]          [,2]          [,3]          [,4]
0  1.759027e-04 -3.139098e-05 -7.259535e-05 -7.191633e-05
1  1.814338e-04 -1.307919e-04 -5.195608e-05  1.314125e-06
2  1.314125e-06 -5.195608e-05 -1.307919e-04  1.814338e-04
3 -7.191633e-05 -7.259535e-05 -3.139098e-05  1.759027e-04
```

which is within rounding error of zero.

The next model is the model with gBGC on the human branch, but no additional selection parameter on the human branch. There is no concept of a global gBGC parameter, since gBGC is a transient effect, so the gBGC parameter B is implicitly 0 in the main model. We can add gBGC to the human branch in the same way we added selection:

```
> initBgcMod <- add.ls.mod(neutralMod, "hg18", separate.params="bgc[0,2000]")
```

The "[0,2000]" assigns boundaries to the bgc parameter (the same can be done any parameters specified in `separate.params`, and probably should have been done for the selection parameter above, as phyloFit estimated it at the default maximum of 200).

Then we can maximize the likelihood using phyloFit in the same way:

```
> bgcMod <- phyloFit(align, init.mod=initBgcMod, no.opt=c("backgd", "branches", "ratematrix"))
> #print the likelihood, global selection, and bgc parameters:
> bgcMod$likelihood

[1] -282.5656

> bgcMod$selection

[1] 0.001752459

> bgcMod$ls.mod$bgc

[1] 1296.724
```

The final model incorporates both lineage-specific gBGC and selection. We can set it up, and maximize the likelihood, like so:

```
> initBgcSelMod <- add.ls.mod(neutralMod, "hg18", separate.params=c("bgc[0,2000]", "sel[-1000,1000]"))
> bgcSelMod <- phyloFit(align, init.mod=initBgcSelMod, no.opt=c("backgd", "branches", "ratematrix"))
> # print likelihood, global selection, lineage-specific selection, and bgc parameters:
> bgcSelMod$likelihood

[1] -282.5557

> bgcSelMod$selection

[1] 0.002397168

> bgcSelMod$alt.mod$selection

NULL

> bgcSelMod$alt.mod$bgc

NULL

>
```

Now that we have likelihoods from all four models, we can compare them as described in Section 2.

References

J.˜A. Capra, M.˜J. Hubisz, D.˜Kostka, K.˜S. Pollard, and A.˜Siepel. 2013.

Laurent Duret and Nicolas Galtier. Biased gene conversion and the evolution of mammalian genomic landscapes. *Annu Rev Genom Hum G*, 10(1):285–311, 2009.

D.˜Kostka, M.˜J. Hubisz, A.˜Siepel, and K.˜S. Pollard. The role of GC-biased gene conversion in shaping the fastest evolving regions of the human genome. *Mol. Biol. Evol.*, 29(3):1047–1057, Mar 2012.

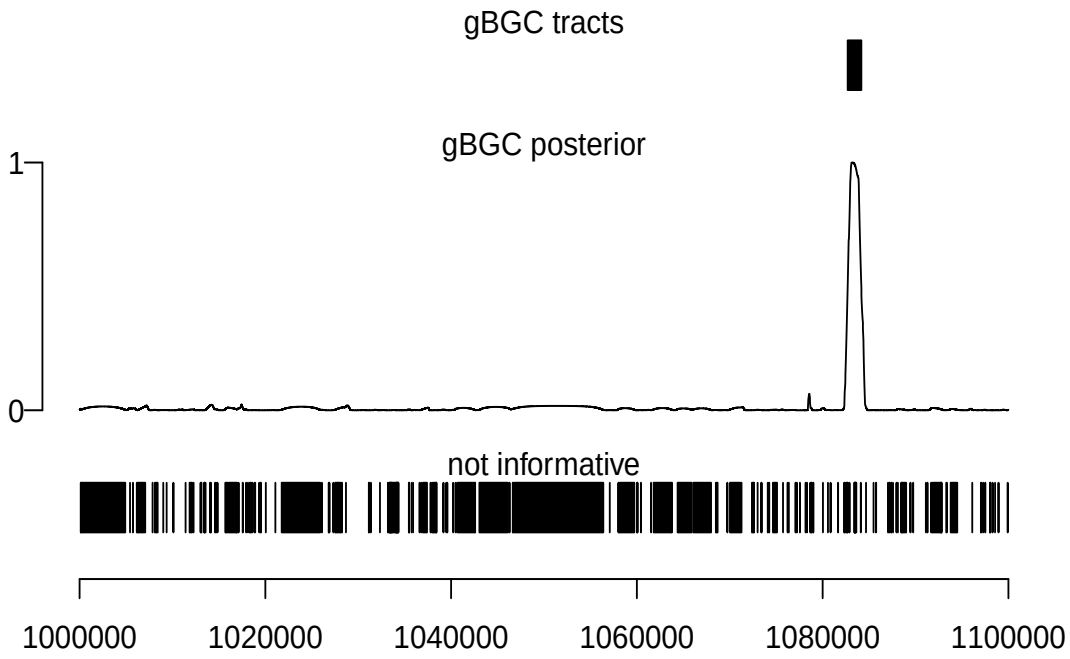

Figure 1: phastBias result plot

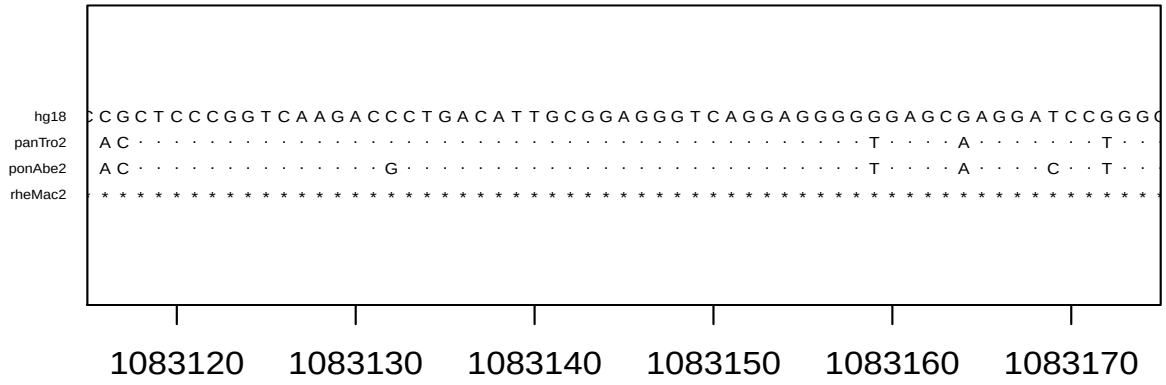

Figure 2: 60 bp region of the alignment in the phastBias tracts containing 4 W→S substitution on the human branch.