An extension of testing 6-state recoding strategies to address substitution bias and compositional heterogeneity

Principal Investigator: Joseph Ryan and Alexandra Hernandez

Version Number: v.1.5.1

2 July 2020

LIST OF ABBREVIATIONS

JTT	Jones-Thornton-Taylor (a model for amino acid substitution)
PAM	Point accepted mutation (a model for amino acid substitution)
RFD	Robinson-foulds distance
SICB	Society for Integrative and Comparative Biology
S&R-6	Susko & Roger 6-state recoding
RAxML	Randomized Axelerated Maximum Likelihood
TOPD	TOPological Distance
FMTS	From Multiple to Single
GTR	General time-reversible model
LG	Le & Gascuel (a model for amino acid substitution)
ML	Maximum-likelihood
RCFV	Relative compositional frequency variability

1   INTRODUCTION: BACKGROUND INFORMATION AND SCIENTIFIC RATIONALE

1.1   BACKGROUND INFORMATION

Dayhoff, JTT, and LG matrices are 20-state amino acid replacement models used to score amino acid substitutions in phylogenetic analyses. Recently, recoding techniques have been employed to address difficulties that these models have dealing with compositional heterogeneity and substitution saturation (Susko & Roger 2007). Dayhoff recoding (i.e., Dayhoff-6) specifically recodes amino acids from Dayhoff matrices according to 6 groups of chemically related amino acids that frequently replace one another (Hrdy et al. 2004), while JTT recoding (i.e., S&R-6) is a 6-state recoding strategy based off binning experiments on the JTT model by Susko & Roger (2007).

1.2   RATIONALE

The principle of using recoding to address substitution saturation and compositional heterogeneity is appealing from a theoretical perspective but has never been tested empirically. Evidence from our analysis presented at the SICB 2018 Conference showed that under all simulations in the study, Dayhoff-6 recoding performed worse than the PAM250 (Dayhoff) model. These preliminary results raised doubts about the benefits of using recoding approaches. However, we received feedback that our analyses did not directly address compositional heterogeneity and that since we used the same model for simulation and testing, we did not consider problems stemming from poor model fit. The proposed analyses herein aim to address these concerns.

1.3   OBJECTIVES

This is a three-part study that will serve as an extension to our project investigating 6-state recoding strategies (*https://github.com/josephryan/Hernandez_Ryan_2018_RecodingSim*). The objectives are 1) determine how recoding performs compared to a model that was not used for simulation (i.e. does recoding improve results when the non-recoded model fit is poor?) 2) verify the performance of recoding strategies using inferred model parameters from the data under a range of saturation levels 3) determine if recoding addresses problems with compositional heterogeneity.

2   STUDY DESIGN AND ENDPOINTS

2.1 Effect of model fit on recoding vs. non-recoding using LG model

2.1.1. Reconstruct topologies in RAxML (Stamatakis 2014) using the LG model to estimate trees for the data generated in the previous analysis via simulations in Seq-Gen (Rambaut & Grassly 1997) (*https://github.com/josephryan/Hernandez_Ryan_2018_RecodingSim)*. These previous datasets were simulated over the Chang et al. (2015) phylogeny under both the Dayhoff and JTT models with branch-length scaling factors (-s) set to 1, 5, 10, 15, and 20.

```
raxmlHPC -p 420 -m PROTGAMMALG -n Chang.1.LG -s Chang.PAM.1.phy
```

2.1.2 Use the program TOPD/FMTS (Puigbó et al. 2007) to compute Robinson-Foulds distances (RFDs) between LG-generated trees and the tree from Chang et al. (2015) that was used to simulate the data. We will compare these RFDs to those from comparisons between the trees estimated using recoded amino acids.

```
cat Chang_orig_phylobayes.tre RAxML_best* > Chang_all_trees.tre

perl topd_v4.6.pl -f Chang_all_trees.tre -m split -r no
```

2.2 Effect of model fit on recoding vs. non-recoding using data simulated w/ estimated model

2.2.1 Simulate the evolution of amino acids along the phylogeny produced in Chang et al. (2015) using Seq-Gen and apply model parameters inferred from the data (i.e., amino acid frequency, alpha parameter, and transition rates taken from the main ML tree of Chang et al. 2015). We asked the authors of Chang et al. (2015) for their RAxML output files, but they did not have these available, so we will reestimate these values using the exact RAxML run that was used in the original study. The branch-length parameter (-s) for simulations will be set to 1, 5, 10, 15, and 20.

```
seq-gen -z420 -mGENERAL -r[rates] -f[frequency] -a[alpha] -n1000 -s1 -or
Chang_orig_phylobayes.tre  > Chang.mismatch.1.phy

seq-gen -z420 -mGENERAL -r[rates] -f[frequency] -a[alpha] -n1000 -s5 -or
Chang_orig_phylobayes.tre  > Chang.mismatch.5.phy

seq-gen -z420 -mGENERAL -r[rates] -f[frequency] -a[alpha] -n1000 -s10 -or
Chang_orig_phylobayes.tre  > Chang.mismatch.10.phy

seq-gen -z420 -mGENERAL -r[rates] -f[frequency] -a[alpha] -n1000 -s15 -or
Chang_orig_phylobayes.tre  > Chang.mismatch.15.phy

seq-gen -z420 -mGENERAL -r[rates] -f[frequency] -a[alpha] -n1000 -s20 -or
Chang_orig_phylobayes.tre  > Chang.mismatch.20.phy
```

divide.pl (For all 5 commands above, divide the 1000 datasets output by Seq-Gen into separate phylip files)

```
perl divide.pl Chang.mismatch.1.phy Chang.mismatch.1
```

2.2.2 Convert simulated sequences to Dayhoff-6 recoded datasets using the script chunkify2.pl. Chunkify2.pl also generates scripts to perform maximum-likelihood analyses in RAxML for each non-recoded dataset under the Dayhoff model and for each recoded dataset under the MULTIGAMMA multi-state model with GTR.

*Note: Chunkify2.pl requires the user to input data on servers being used and processors available for each server (this is hard-coded in the script and should be altered when reproducing results on another machine).

```
perl chunkify_estimate.pl
```

To execute the scripts generated from chunkify_estimate.pl for maximum-likelihood analyses of both recoded and non-recoded datasets:

```
ls -1 servername* | perl -ne 'chomp; print "sh $_ &\n";' | sh
```

2.2.3 Concatenate trees from section 2.2.2 into one tree file along with the true tree used for simulation, then calculate RFDs in TOPD/FMTS to compare differences between trees. Box plots will be used to visually compare RFDs and t-tests will be used to determine if there are significant differences in RFDs between non-recoded and recoded trees in R (R Core Team 2017). The script compare_trees_to_sim.pl.v04 performs these analyses.

*Note: The directories in which to perform these analyses are hardcoded and should be altered when reproducing results on another machine.

```
perl compare_trees_to_sim_estimate.pl
```

2.3 Effect of compositional heterogeneity

2.3.1 To create compositionally heterogeneous datasets, we will use P4 (Foster 2004) to simulate the evolution of amino acids 1,000 times along a 20-taxa tree (Figure 1). For each simulation, we will apply the GTR model and set the same amino acid frequencies for clades A and C using the setModelThing parameter, while applying a different set of amino acid frequencies to the rest of the tree. This will make clades A and C compositionally homogenous to each other, clades B and D compositionally homogeneous to each other, and any other combination of these clades as compositionally heterogeneous to each other.

The initial rates and frequencies we will use will be determined from the tree generated from Chang et al. (2015). For clades A and C, frequencies for the following pairs of amino acids ((A,L), (R,K), (N,M), (D,F), (C,P), (Q,S), (E,T), (G,W), (H,Y), (I,V); chosen based on an alphabetical pattern) will be adjusted by X, where X is the inflation parameter (i.e., 0.1, 0.5, 0.9) multiplied by the lowest frequency of the pair. The amino acid of the pair with the lowest frequency is incremented by X and the other is decremented by X. These datasets are generated in our script comphet.pl.

```
perl comphet.pl
```

2.3.2 To create a null distribution, we will simulate the evolution of amino acids 1,000,000 times along the same 20-taxa tree (Figure 1) to produce non-heterogeneous datasets using P4. We will use the amino acid frequencies and rates from Chang et al. (2015) and simulate under GTR. The shell scripts we use to create these datasets are generated in our script split_random_runs.pl.

```
perl split_random_runs.pl
```

To execute the scripts generated from split_random_runs.pl we run the following:

```
ls -1 scripts/ | grep 'servername*' | perl -ne 'chomp; print "sh scripts/$_
&\n";' | sh
```
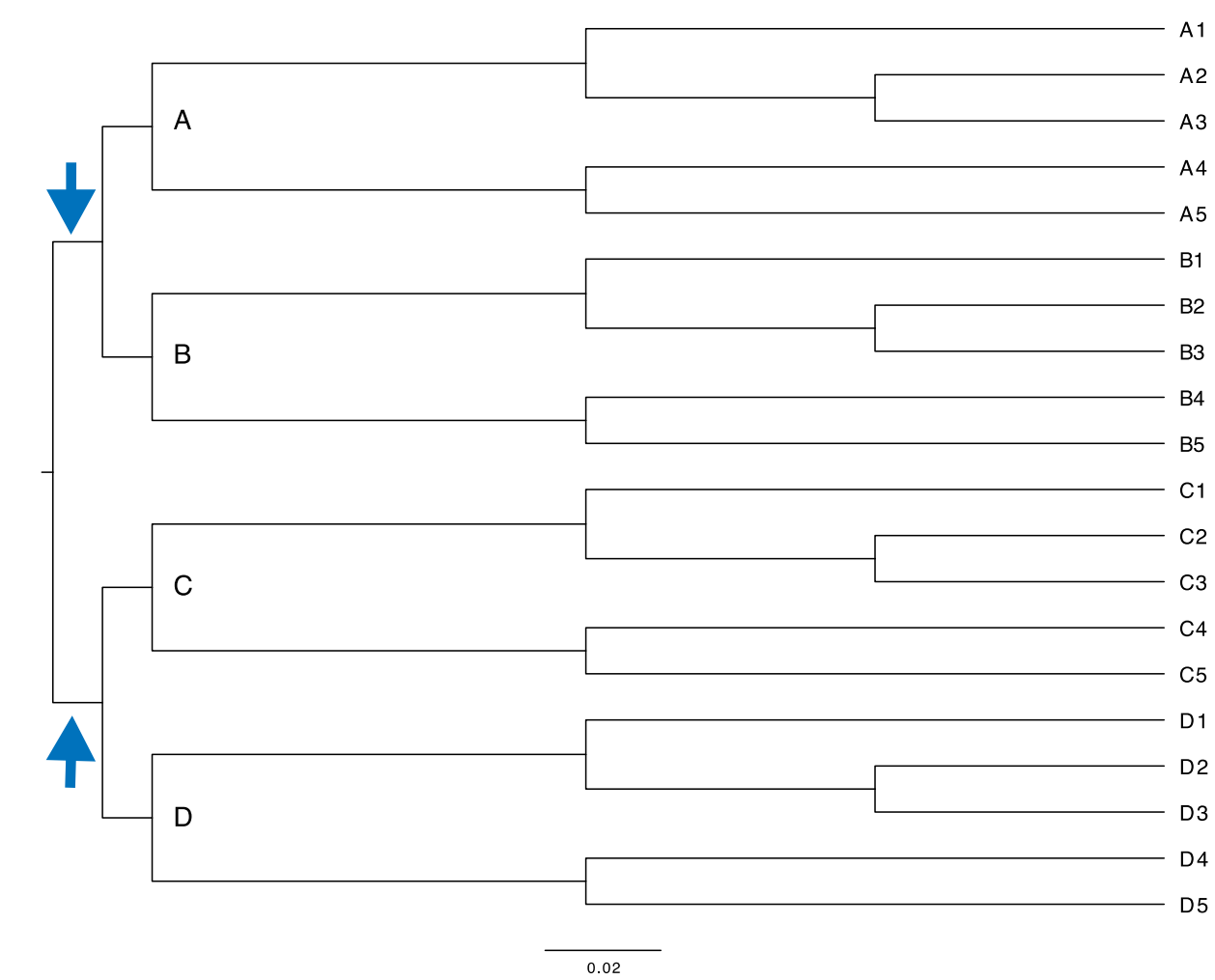

Figure 1. Tree used for simulation in P4. Blue arrows point out which branches we will shorten if there is no significant difference in the comp-het index between the heterogeneous datasets and null distribution.

2.3.3 For our heterogeneous datasets, we will verify that clades A and C are compositionally different from clades B and D by computing amino acid frequencies across the 1,000 datasets. We will calculate the sum of frequency differences for each of the following clade combinations: AB, AD, BC, and CD. This sum of differences will represent clades that should be heterogeneous to each other. We will then compute the average sum of frequency differences for all heterogenous clade combinations (AB+AD+BC+CD/4).

For clade combinations expected to be homogeneous to each other (AC and BD), we will sum frequency differences and calculate the average for these values (AC + BD/2). The average value for the homogeneous clade combinations will be subtracted from the average value of heterogenous clade combinations:

((AB+AD+BC+CD/4) - (AC+BD/2))

This is called the compositional heterogeneity index, or comp-het index for short. This comp-het index will be used as a test statistic for our heterogeneous datasets. Using this same script, we will compute 1,000 comp-het indices from the 1,000,000 non-heterogeneous datasets and calculate the number of these that are greater than our test statistic. If our test statistic is not significant (i.e., we identify more than 50 comp-het indices from non-heterogeneous datasets that are greater than our test statistic), we will shorten the shortest internode branches on the 20-taxa tree (Figure 1) to decrease the amount of shared evolution between clades A and B as well as between clades C and D and perform 2.3.1 and test again (as done in this section 2.3.3). The script comphet.pl will generate the p-value telling whether there is a significant difference between our heterogeneous datasets (generated in 2.3.1) and our null distribution (generated in 2.3.2).

```
perl comphet.pl all.out > pvals.out
```

2.3.4 We will recode the 1,000 heterogenous datasets selected from section 2.3.3 using Dayhoff-6 recoding and perform maximum-likelihood analyses in RAxML for both the recoded and non-recoded datasets under the Dayhoff model using the script chunkify_comphet_trees.pl (performs the same as chunkify2.pl but with parameters specifically set for this dataset).

```
perl chunkify_comphet_trees.pl
```

To execute the scripts generated from chunkify_comphet_trees.pl for maximum-likelihood analyses of both recoded and non-recoded datasets:

```
ls -1 scripts/ | grep 'servername*' | perl -ne 'chomp; print "sh scripts/$_
&\n";' | sh
```

2.3.5 We will use the is_mono.pl to determine what proportion of trees resolve a monophyletic clade of A and B and a monophyletic clade of C and D.

```
perl is_mono.pl dir/
```

2.3.6 To get an idea of the extent that compositional heterogeneity impacts tree reconstruction we will run 2.3.1-2.3.5 until there are trees correctly resolved that fall into the following categories: 5-25% correctly resolved, 25-50% correctly resolved, and 50-75% correctly resolved.

2.3.7 We will use a z-test to compare the percentages of incorrect trees under the non-recoded and recoded datasets for each inflation parameter and tree tested. The z-test will be performed in R using the script ztest_proportions_comphet.R.

2.4 Effect of compositional heterogeneity with alternative Dayhoff recoding strategies

2.4.1 We used the data generated for tree 0.002 in 2.3.6 and reconstructed trees using several alternative Dayhoff recoding strategies including Dayhoff-9, Dayhoff-12, Dayhoff-15, and Dayhoff-18.

The following Dayhoff recoding schemes were applied:

Dayhoff-9: DEHNQ ILMV FY AST KR G P C W

Dayhoff-12: DEQ MLIV FY KHR G A P S T N W C

Dayhoff-15: DEQ ML IV FY G A P S T N K H R W C

Dayhoff-18: ML FY I V G A P S T D E Q N H K R W C

NOTE: We used an exhaustive algorithm *get_categories_from_state_csv.pl* to identify the 9-state recoding scheme with a category structure of 5,4,3,2,2,1,1,1,1 (other schemes with this same structure may score as well, but none should be better). This script did not complete when applied to 12-, 15-, and 18-state recodings. Here are the commands we ran for exhaustive searches:

```
perl get_categories_from_state_csv.pl 5,4,3,2,2,1,1,1,1
perl get_categories_from_state_csv.pl 4,3,3,2,1,1,1,1,1,1,1,1
perl get_categories_from_state_csv.pl 3,2,2,2,1,1,1,1,1,1,1,1,1,1,1
perl get_categories_from_state_csv.pl 2,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
perl get_categories_from_state_csv.pl 2,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
```

Since they did not complete, we tested some reasonable schemes using a second script *score.pl* and chose the scheme with the highest scores. In these cases, other schemes may score higher.   The following are the tested schemes:

#12 state tests
```
perl score.pl 'DEQ MLIV FY G A P S T N KHR W C'
perl score.pl 'DEQ M LIV FY G APST N K H R W C'
perl score.pl 'DEQ M LIV FY GAPS T N K H R W C'
perl score.pl 'DEQ M LIV FY GAPT S N K H R W C'
perl score.pl 'DEQ MLIV FY GAP T S N K H R W C'
perl score.pl 'D E Q MLIV FY GAP S T N KHR W C'
```

15 state tests
```
perl score.pl 'ML IV G A P S T DE Q N K H RFY W C'
perl score.pl 'ML IV G A P S T DE Q N K H R FYW C'
perl score.pl 'ML IV G A P S T DEQ N K H R FY W C'
perl score.pl 'ML IV G A P S T DEN Q K H R FY W C'
perl score.pl 'ML I V GAP S T DE Q N K H R FY W C'
perl score.pl 'ML IVG A P S T DE Q N K H R FY W C'
perl score.pl 'ML IV G A P S T DEN Q K H R FY W C'
perl score.pl 'ML IV G A P S T D E Q N KHR FY W C'
perl score.pl 'ML I V G A P S T DE Q N KHR FY W C'
```

18 state tests
```
perl score.pl 'IV ML G A P S T D E Q N H K R F Y W C'
perl score.pl 'IV M L G A P S T D E Q N H K R F Y WC'
perl score.pl 'IV M L G A P S T DE Q N H K R F Y W C'
perl score.pl 'IV M L G A P S T D E QN H K R F Y W C'
perl score.pl 'IV ML G A P S T D E Q N H K R F Y W C'
perl score.pl 'MV IL G A P S T D E Q N H K R F Y W C'
perl score.pl 'MI VL G A P S T D E Q N H K R F Y W C'
perl score.pl 'ML IV G A P S T D E Q N H K R F Y W C'
perl score.pl 'ML I V GA P S T D E Q N H K R F Y W C'
perl score.pl 'ML I V G AP S T D E Q N H K R F Y W C'
perl score.pl 'ML I V G A P ST D E Q N H K R F Y W C'
perl score.pl 'ML I V G A P S T DE Q N H K R F Y W C'
perl score.pl 'ML I V G A P S T D E QN H K R F Y W C'
perl score.pl 'ML I V G A P S T D EQ N H K R F Y W C'
perl score.pl 'ML I V G A P S T D E Q N H K RF Y W C'
perl score.pl 'ML I V G A P S T D E Q N H K R FY W C'
perl score.pl 'ML I V G A P S T D E Q N HK R F Y W C'
perl score.pl 'ML I V G A P S T D E Q N H KR F Y W C'
```

```
perl score.pl 'ML I V G A P S T D E Q N K HR F Y W C'
```

We ran the following commands for each recoding method to convert simulated sequences to Dayhoff-n recoded datasets where n is the number of recoded groups. *chunkify_nscheme_recoding.pl* also generates scripts to run maximum-likelihood analyses in RAxML under the Dayhoff model for non-recoded datasets and under the MULTIGAMMA multi-state model with GTR for Dayhoff n-recoded datasets:

```
mkdir nscheme_scripts
```

```
perl chunkify_nscheme_recoding.pl
```

*Note: *chunkify_nscheme_recoding.pl* requires the user to input data on servers being used and processors available for each server (this is hard-coded in the script and should be altered when reproducing results on another machine).

To execute the scripts generated from *chunkify_nscheme_recoding.pl* for maximum-likelihood analyses of both recoded and non-recoded datasets:

```
perl ls -1 nscheme_scripts/ | grep 'servername*' | perl -ne 'chomp; print "sh
nscheme_scripts/$_ &\n";' | sh
```

2.4.2 We used the is_mono_comphet_nscheme.pl to determine what proportion of trees resolve a monophyletic clade of A and B and a monophyletic clade of C and D.

```
perl is_mono_comphet_nscheme.pl dir/
```


Additional analyses suggested by reviewers after submission to *Sytematic Biology*

2.5 Testing Dayhoff 6-state recoding under no induced compositional heterogeneity

2.5.1 A reviewer requested that we run an analysis that included no added compositional heterogeneity. We had already generated these data when calculating our null distribution (section 2.3.2) but did not analyze them as part of our main analyses. We will select the first 1,000 datasets out of the 1,000,000 generated for the null distribution in section 2.3.2 and reconstruct maximum-likelihood trees for these data using Dayhoff and Dayhoff-6 recoding.

```
perl chunkify_random_null.pl
```

2.5.2 Trees will be scored using is_mono.pl (as outlined in 2.3.5).

```
perl is_mono.pl DAYHOFF > is_mono_dayhoff.out
```

2.6. Effect of data size on the performance of recoding methods

2.6.1 A reviewer requested that we test the effect of alignment size in both our compositional heterogeneity analyses and our saturation analyses. To do this for saturation, we will simulate 1,000 datasets made up of 2,000 and 5,000 amino-acid columns using the Chang et al. (2015) topology and the Dayhoff model with the branch length scaling factor parameter set to 1, 5, 10, 15, and 20.

```
seq-gen -mPAM -z420 -n1000 -s1.0 -a1.0 -l2000 -or Chang_orig_phylobayes.tre >
Chang.2000.1.phy
```

```
seq-gen -mPAM -z420 -n1000 -s5.0 -a1.0 -l2000 -or Chang_orig_phylobayes.tre >
Chang.2000.5.phy
```

```
seq-gen -mPAM -z420 -n1000 -s10.0 -a1.0 -l2000 -or Chang_orig_phylobayes.tre
> Chang.2000.10.phy
```

```
seq-gen -mPAM -z420 -n1000 -s15.0 -a1.0 -l2000 -or Chang_orig_phylobayes.tre
> Chang.2000.15.phy
```

```
seq-gen -mPAM -z420 -n1000 -s20.0 -a1.0 -l2000 -or Chang_orig_phylobayes.tre
> Chang.2000.20.phy
```

```
seq-gen -mPAM -z420 -n1000 -s1.0 -a1.0 -l5000 -or Chang_orig_phylobayes.tre >
Chang.5000.1.phy
```

```
seq-gen -mPAM -z420 -n1000 -s5.0 -a1.0 -l5000 -or Chang_orig_phylobayes.tre >
Chang.5000.5.phy
```

```
seq-gen -mPAM -z420 -n1000 -s10.0 -a1.0 -l5000 -or Chang_orig_phylobayes.tre
> Chang.5000.10.phy
```

```
seq-gen -mPAM -z420 -n1000 -s15.0 -a1.0 -l5000 -or Chang_orig_phylobayes.tre
> Chang.5000.15.phy
```

```
seq-gen -mPAM -z420 -n1000 -s20.0 -a1.0 -l5000 -or Chang_orig_phylobayes.tre
> Chang.5000.20.phy
```

divide.pl (For all 10 commands above, divide the 1000 datasets output by Seq-Gen into separate phylip files)

```
perl divide.pl Chang.2000.1.phy Chang.2000.1
```

2.6.2 All simulated datasets will be converted to Dayhoff-6 recoded datasets using the script chunkify_estimate.pl. chunkify_estimate.pl will also be used to generate shell scripts for maximum-likelihood analyses in RAxML. The Dayhoff model will be used for non-recoded datasets and the MULTIGAMMA multi-state model with GTR will be used for Dayhoff-6 recoded datasets.

```
mkdir scripts
```

```
perl chunkify_estimate.pl
```

To execute the scripts generated from chunkify_estimate.pl, we will run the following command:

```
ls -1 scripts/ | grep 'servername*' | perl -ne 'chomp; print "sh scripts/$_
&\n";' | sh
```

2.6.3 We will score trees using RFD values calculated in TOPD/FMTS as in section 2.2.3. We will also score trees by determining the proportion of trees that do not recover deep splits (as outlined in section 2.3.5).

```
perl rfd_comphet.pl TREE000C DAYHOFF > rfd_dayhoff.out
```

```
perl is_mono.pl DAYHOFF > is_mono_dayhoff.out
```

2.6.4 To test the effect of alignment width for compositional heterogeneity, we will simulate 1,000 datasets (as outlined in section 2.3.1) using tree 0.002 (Figure 1a; Hernandez & Ryan 2019) with the inflation parameter set to 0.1, 0.5, and 0.9; simulated sequences will be 2,000 and 5,000 amino acids in length.

```
perl comphet.pl TREE0002 0.1 2,000 ALPHABETICAL all.out > pvals.0.1.out

perl comphet.pl TREE0002 0.5 2,000 ALPHABETICAL all.out > pvals.0.5.out

perl comphet.pl TREE0002 0.9 2,000 ALPHABETICAL all.out > pvals.0.9.out

perl comphet.pl TREE0002 0.1 5,000 ALPHABETICAL all.out > pvals.0.1.out

perl comphet.pl TREE0002 0.5 5,000 ALPHABETICAL all.out > pvals.0.5.out

perl comphet.pl TREE0002 0.9 5,000 ALPHABETICAL all.out > pvals.0.9.out
```

2.6.5 We will convert all simulated datasets to Dayhoff-6 recoded datasets and generate shell scripts to perform maximum-likelihood analyses in RAxML using the script chunkify_estimate.pl as in section 2.6.2.

```
perl chunkify_estimate.pl
```

To execute the scripts generated from chunkify_estimate.pl, we will run the following command:

```
ls -1 scripts/ | grep 'servername*' | perl -ne 'chomp; print "sh scripts/$_
&\n";' | sh
```

2.6.6 Trees will be scored via RFD values and deep splits by applying the same analyses outlined in section 2.6.3.

```
perl rfd_comphet.pl TREE0002 DAYHOFF > rfd_dayhoff.out
```

```
perl is_mono.pl DAYHOFF > is_mono_dayhoff.out
```

2.7. Effect of tree shape and compositional heterogeneity on recoding methods

2.7.1 A reviewer was concerned with our starting tree having a simple balanced topology. To address this, we will simulate 1,000 compositionally heterogeneous datasets on the topology from Chang et al. (2015). Simulated sequences will be 1,000 amino acids in length. We will apply one set of amino acid frequencies to the clades Porifera and Choanoflagellata using the setModelThing parameter, while applying a different set of amino acid frequencies to the rest of the tree. Datasets will be simulated with the inflation parameter set to 0.1, 0.5, and 0.9.

```
perl comphet.pl CHANG 0.1 1,000 ALPHABETICAL all.out > pvals.0.1.out

perl comphet.pl CHANG 0.5 1,000 ALPHABETICAL all.out > pvals.0.5.out

perl comphet.pl CHANG 0.9 1,000 ALPHABETICAL all.out > pvals.0.9.out
```

2.7.2 We will convert all simulated datasets to Dayhoff-6 recoded datasets and generate shell scripts to perform maximum-likelihood analyses in RAxML using the script chunkify_estimate.pl as in section 2.6.2.

```
mkdir scripts
```

```
perl chunkify_estimate.pl
```

To execute the scripts generated from chunkify_estimate.pl, we will run the following command:

```
ls -1 scripts/ | grep 'servername*' | perl -ne 'chomp; print "sh $_ &\n";' |
sh
```

2.7.3 We will score trees by calculating RFD values in TOPD/FMTS. We will also score trees by determining the proportion of trees that do not recover deep splits (i.e., trees that do not recover a monophyletic group including Ctenophora, Placozoa, Cnidaria, and Bilateria).

```
perl rfd_comphet.pl TREE000C DAYHOFF > rfd_dayhoff.out
```

```
perl is_mono.pl DAYHOFF CHANG > is_mono_dayhoff.out
```

2.8. Effect of amino acid frequencies and compositional heterogeneity on recoding methods

2.8.1 A reviewer suggested that we repeat our compositional heterogeneity analyses from section 2.3.1 with starting amino acid frequency pairs sampled at random instead of alphabetically paired. To address this, we will simulate 1,000 datasets with random pairings of amino acid frequencies across replicates on tree 0.002 for each inflation parameter (i.e., 0.1, 0.5, 0.9).

```
perl comphet.pl TREE0002 0.1 1,000 RANDOM all.out > pvals.0.1.out
```

```
perl comphet.pl TREE0002 0.5 1,000 RANDOM all.out > pvals.0.5.out
```

```
perl comphet.pl TREE0002 0.9 1,000 RANDOM all.out > pvals.0.9.out
```

2.8.2 We will convert all simulated datasets to Dayhoff-6 recoded datasets and generate shell scripts to perform maximum-likelihood analyses in RAxML using the script chunkify_estimate.pl as in section 2.6.2.

```
perl chunkify_estimate.pl
```

To execute the scripts generated from chunkify_estimate.pl, we will run the following command:

```
ls -1 scripts/ | grep 'servername*' | perl -ne 'chomp; print "sh scripts/$_
&\n";' | sh
```

2.6.6 Trees will be scored via RFD values and deep splits by applying the same analyses outlined in section 2.6.3.

```
perl rfd_comphet.pl TREE0002 DAYHOFF > rfd_dayhoff.out
```

```
perl is_mono.pl DAYHOFF > is_mono_dayhoff.out
```

2.9. Statistical analyses

2.9.1 A reviewer pointed out that the z-test originally performed (for compositional heterogenetiety analyses) was not appropriate and that we did not correct for multiple testing in our t-tests (for saturation analyses). To address these concerns, we will instead use a bootstrapping approach to estimate the amount of evidence supporting each analysis. For each starting tree (i.e., 0.008, 0.004, 0.002, 0.001) and inflation parameter (i.e., 0.1, 0.5, 0.9) combination, we will sample 10,000 random sets of 1,000 alignments (with replacement) from our initial simulated alignments. Since we have already generated trees for each of these alignments, we will not reestimate trees from sampled alignments. Instead, we will construct a matrix of scores (i.e., correct or incorrect) for each alignment and

method (i.e., Dayhoff, JTT, Dayhoff-6, S&R-6) from our previously generated trees. We will generate a p-value by counting the number of sampled alignments where the percentage of incorrect trees for recoded alignments are less than or equal to the percentage of incorrect trees for non-recoded alignments, divided by the number of samples (i.e., 10,000). We will do this for all compositional heterogeneity datasets. For saturated datasets, we will apply the same approach for each branch-length scaling factor parameter and generate a p-value by counting the number of sampled alignments where the RFD values for recoded alignments is less than or equal to the RFD values for non-recoded alignments, divided by the number of samples (i.e., 10,000).

```
perl bootstrap_comphet.pl
```

```
perl bootstrap_saturation.pl
```

Follow-up analyses

3.1. Statical analyses

3.1.1 After coding and executing the proposed bootstrapping approach, we realized that our design did not statistically test the significance of our results. Rather than redesign a Monte Carlo approach that did test significance, we chose instead to apply a t-test (as initially done in Hernandez & Ryan 2019) and correct for multiple testing using the Bonferroni correction. In addition we will apply a contingency table for our analyses of deep splits in compositionally heterogeneous datasets. Both of these approaches were recommended by reviewers. We will use an R script to perform the chi-square test.

```
deepsplits_chi_square.R
```

3.1.2 A reviewer also suggested that we calculate RFD values for compositionally heterogenous datasets to compliment the analysis of deep splits in section 2.3. We will calculate RFD values in TOPD/FMTS and generate box plots in R to visually compare RFDs. We will also perform t-tests to determine if there are significant differences in RFDs between non-recoded and recoded trees (as outlined section 2.2.3) and correct for multiple testing using the Bonferroni correction.

```
perl rfd_comphet.pl
```

3.2. Effect of data size and compositional heterogeneity on the performance of recoding

3.2.1 To test the effect of alignment width on the performance of recoding, we simulated datasets under several levels of compositional heterogeneity that were 2,000 and 5,000 amino acids in length (sections 2.6.4 -2.6.6). Phylogenetic reconstructions of these datasets showed a decreasing percentage of error with increasing data size. To gain a clearer understanding of the trend, we will complement our analyses of 1,000-, 2,000-, and 5,000-column simulated amino-acid alignments with 3,000- and 4,000-column alignments using tree 0.002 (Figure 1a; Hernandez & Ryan 2019) with the inflation parameter set to 0.1, 0.5, and 0.9.

```
perl comphet.pl TREE0002 0.1 3,000 ALPHABETICAL all.out > pvals.0.1.out
```

```
perl comphet.pl TREE0002 0.5 3,000 ALPHABETICAL all.out > pvals.0.5.out
```

```
perl comphet.pl TREE0002 0.9 3,000 ALPHABETICAL all.out > pvals.0.9.out
```

```
perl comphet.pl TREE0002 0.1 4,000 ALPHABETICAL all.out > pvals.0.1.out
```

```
perl comphet.pl TREE0002 0.5 4,000 ALPHABETICAL all.out > pvals.0.5.out
```

```
perl comphet.pl TREE0002 0.9 4,000 ALPHABETICAL all.out > pvals.0.9.out
```

3.2.2 We will convert all simulated datasets to Dayhoff-6 recoded datasets and generate shell scripts to perform maximum-likelihood analyses in RAxML using the script chunkify_estimate.pl as in section 2.6.2.

```
mkdir scripts
```

```
perl chunkify_estimate.pl
```

To execute the scripts generated from chunkify_lengths.pl, we will run the following command:

```
ls -1 scripts/ | grep 'servername*' | perl -ne 'chomp; print "sh scripts/$_
&\n";' | sh
```

3.2.3 Trees will be scored by RFD values and deep splits as outlined in section 2.6.3.

```
perl rfd_comphet.pl TREE0002 DAYHOFF > rfd_dayhoff.out
```

```
perl is_mono.pl DAYHOFF > is_mono_dayhoff.out
```

3.2.4 There was very little difference in percentage of incorrect trees between 0.1 and 0.5 inflation parameters in our non-recoded analyses of 2,000-column datasets, but there was a substantial increase in the percentage of incorrect trees when the inflation parameter was increased to 0.9 (Figure 2). To better understand the dynamics of that shift, we will conduct the same analyses on 2,000-column datasets under inflation parameters of 0.6, 0.7, and 0.8.

```
perl comphet.pl TREE0002 0.6 2,000 ALPHABETICAL all.out > pvals.0.1.out
```

```
perl comphet.pl TREE0002 0.7 2,000 ALPHABETICAL all.out > pvals.0.5.out
```

```
perl comphet.pl TREE0002 0.8 2,000 ALPHABETICAL all.out > pvals.0.9.out
```
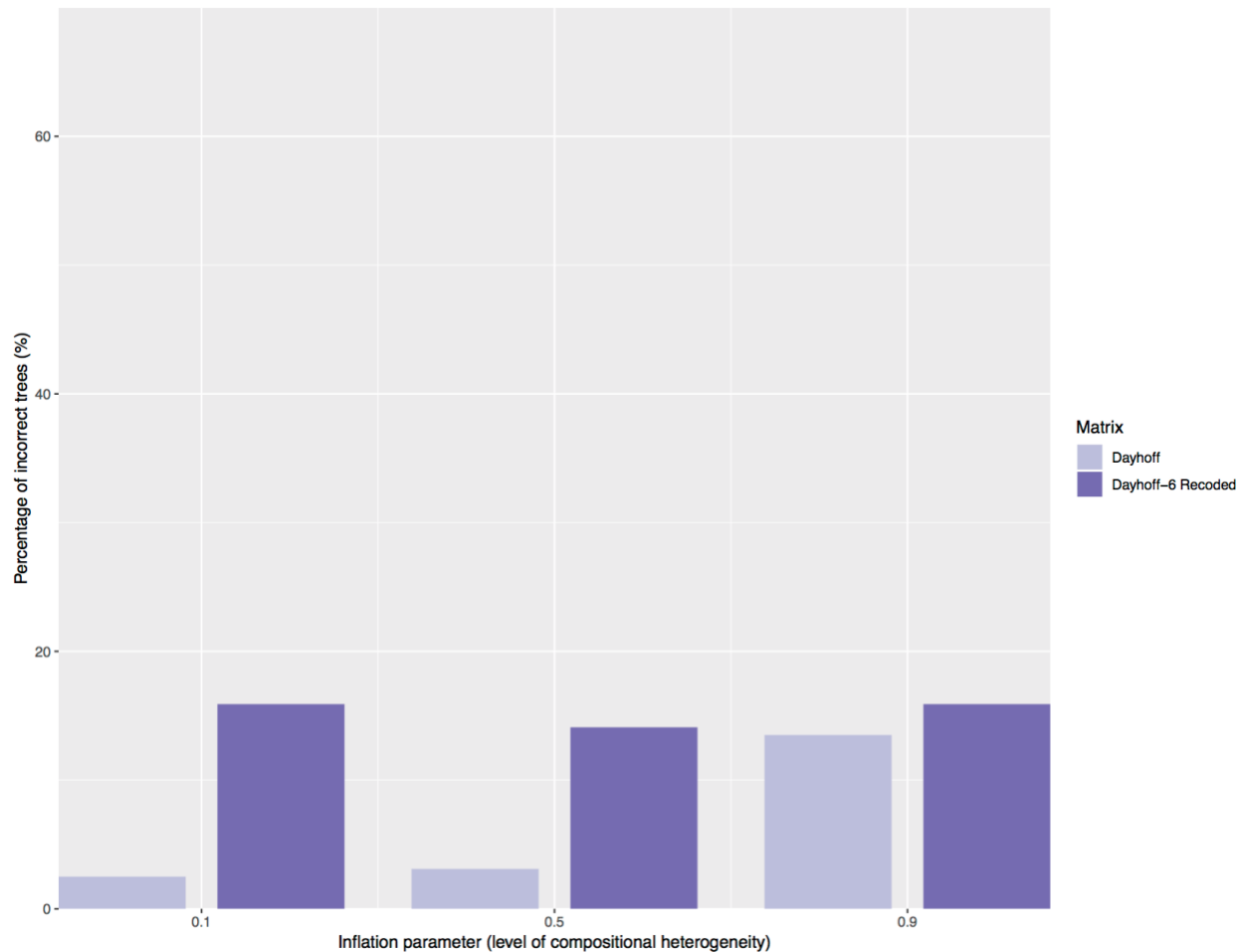
Figure 2. Percentage of error increases under non-recoding for datatsets simulated with the highest level of compositional heterogeneity. Datasets were simulated with a length of 2,000 amino acid positions. Percentage of incorrect trees were those that did not reconstruct a monophyletic group of taxa from clades A and B and a monophyletic group of taxa from clades C and D out of 1,000 datasets.

3.2.5 We will convert all simulated datasets to Dayhoff-6 recoded datasets and generate shell scripts to perform maximum-likelihood analyses in RAxML using the script chunkify_estimate.pl as in section 2.6.2.

```
mkdir scripts
```

```
perl chunkify_estimate.pl
```

To execute the scripts generated from chunkify_lengths.pl, we will run the following command:

```
ls -1 scripts/ | grep 'servername*' | perl -ne 'chomp; print "sh scripts/$_
&\n";' | sh
```

3.2.6 Trees will be scored using RFD values and deep splits using the same analyses outlined in section 2.6.3.

```
perl rfd_comphet.pl TREE0002 DAYHOFF > rfd_dayhoff.out
```

```
perl is_mono.pl DAYHOFF > is_mono_dayhoff.out
```

3.3. Compare levels of compositional heterogeneity from real datasets and to simulated data

3.3.1 Given the hit in performance of non-recoded analyses at very high inflation parameters, we will assess how our simulated levels of compositional heterogeneity compare to those present among real datasets. To do this, we will use RCFV values, which measure the absolute deviation for each amino acid and taxon (Kück and Struck 2014), to gauge the level of compositional heterogeneity in real vs. simulated datasets (from section 2.3.1). To make it simpler to generate RCFV scores for thousands of datasets, we wrote a simple Perl script rcfv.pl (which we will make available in our GitHub repo). For each alignment we will run:

```
perl rcvf.pl --alignment=ALN.phy --phylip
```

3.3.2 To compare RCFV values from simulated data to real data, we will run rcvf.pl on at least one dataset from the 41 papers that include an analysis of amino-acid recoded dataset as a main figure in the paper (Table 1 of the Hernandez & Ryan 2019 preprint). These alignments should be enriched in compositionally heterogeneous datasets as this is one of the common justifications given for applying recoding. We will only include datasets where the data is readily available (supplement, Dryad, github, etc.). If we are unable to find 25 from these 41 papers, we will dip into the papers that only include these analyses in the supplement (rather than as a main figure). We will start sampling the most recent papers first until we reach 25 datasets.

```
perl rcvf.pl --alignment=ALN.phy --phylip
```

3.3.3 We will test the RCFV scores calculated from simulated datasets and real datasets for normality using the Shapiro-Wilk's test. If the data falls under a normal distribution, we will test if there are significant differences in RCFV scores between the real data and simulated data using an ANOVA test. If the data falls under a non-normal distribution, we will test for significant differences amongst scores using the Kruskal–Wallis test.

```
normality_test.R
```

```
anova_simulated_v_real.R
```

or

```
kwtest_simulated_v_real.R
```

3   WORK COMPLETED SO FAR W DATES

June 26, 2018: We have not performed any additional tests after completing the work from the Hernandez_Ryan_2018_RecodingSim repo.

September 26, 2018: We have completed 2.1 and 2.2. We also completed a previous version of 2.3, where the tree used for simulation had very long branches connecting clades A with B and clades C with D. The alignments produced with this tree were predicted by all methods 100% correct every time, and therefore were not useful for testing or realistic. This new version of 2.3 is aimed at correcting this problem.

April 30, 2019: We completed sections 2.3.1-2.3.6 and section 2.4 prior to them being added to the phylotocol. All analyses related to these additions that were run are reported here.

May 13, 2020: The work from sections 2.5-2.8 have been completed.

4   LITERATURE REFERENCES

Chang, E. S., Neuhof, M., Rubinstein, N. D., Diamant, A., Philippe, H., Huchon, D., & Cartwright, P. (2015). Genomic insights into the evolutionary origin of Myxozoa within Cnidaria. *Proceedings of the National Academy of Sciences of the United States of America*, *112*(48), 14912–7. *https://doi.org/10.1073/pnas.1511468112*

Melissa B DeBiasse, William N Colgan, Lincoln Harris, Bradley Davidson, Joseph F Ryan (2020). Inferring tunicate relationships and the evolution of the tunicate Hox cluster with the genome of Corella inflata, *Genome Biology and Evolution*, evaa060, https://doi.org/10.1093/gbe/evaa060

Hernandez, A. M., & Ryan, J. F. (2019). Six-state amino acid recoding is not an effective strategy to offset the effects of compositional heterogeneity and saturation in phylogenetic analyses. *BioRxiv*, 729103.

Hrdy, I., Hirt, R. P., Dolezal, P., Bardonová, L., Foster, P. G., Tachezy, J., & Martin Embley, T. (2004). Trichomonas hydrogenosomes contain the NADH dehydrogenase module of mitochondrial complex I. *Nature*, *432*(7017), 618–622. *https://doi.org/10.1038/nature03149*

Kück, P., & Struck, T. H. (2014). BaCoCa–A heuristic software tool for the parallel assessment of sequence biases in hundreds of gene and taxon partitions. *Molecular Phylogenetics and Evolution, 70,* 94-98.

Marlétaz, F., Peijnenburg, K. T., Goto, T., Satoh, N., & Rokhsar, D. S. (2019). A new spiralian phylogeny places the enigmatic arrow worms among gnathiferans. *Current Biology, 29*(2), 312-318.

Puigbo, P., Garcia-Vallve, S., & McInerney, J. O. (2007). TOPD/FMTS: a new software to compare phylogenetic trees. *Bioinformatics*, *23*(12), 1556–1558. https://doi.org/10.1093/bioinformatics/btm135

R Development Core Team (2017). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org

Rambaut, A., & Grass, N. C. (1997). Seq-Gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. *Bioinformatics*, *13*(3), 235–238. https://doi.org/10.1093/bioinformatics/13.3.235

Stamatakis, A. (2014). RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*, *30*(9), 1312–1313. https://doi.org/10.1093/bioinformatics/btu033

Susko, E., & Roger, A. J. (2007). On Reduced Amino Acid Alphabets for Phylogenetic Inference. *Molecular Biology and Evolution*, *24*(9), 2139–2150. https://doi.org/10.1093/molbev/msm144

5   PHYLOTOCOL AMENDMENT HISTORY

Version	Date	Significant Revisions
1.1	26 June 2018	Changed methods in section 2.3
1.2	1 October 2018	Changed methods in section 2.3 because we could not adequately simulate and test compositional heterogeneity
1.3	30 April 2019	We added section 2.3.7 to test for statistical significance of results from 2.3.6. We also added section 2.4 to test alternative Dayhoff recoding strategies.

Version	Date	Significant Revisions
1.4	7 November 2019	After receiving comments from reviewers through our submission to *Systematic Biology*, we have added analyses in sections 2.5-2.9. We have also updated the language in section 2.3.1 for clarity.
1.5	22 May 2020	After consulting with several scientists about the proposed statistics in section 2.9, we are making necessary corrections (section 3.1). We are also adding additional analyses (section 3.2-3.3) to further explore the results from 2.5-2.8.
1.5.1	2 July 2020	Updated with appropriate commands to run analyses in sections 2.6-2.8 and 3.2.