Spring 4-15-2022

Measurement Error Modeling Applied to Phylogenetic Inference and Parametric Bootstrap Approach to Multifactor ANOVA Models with Unequal Variances and Unbalanced Data

Sarah Katharine Alver
University of New Mexico

Sarah Alver

Candidate

Mathematics and Statistics

Department

This dissertation is approved, and it is acceptable in quality and form for publication:

Approved by the Dissertation Committee:

James Degnan, Chairperson

Guoyi Zhang

Fletcher Christensen

Jeffrey Long

Measurement Error Modeling Applied to Phylogenetic Inference and Parametric Bootstrap Approach to Multifactor ANOVA Models with Unequal Variances and Unbalanced Data

by

Sarah K. Alver

B.S., Nutrition and Food Science, Utah State University, 2001
M.S., Biostatistics, University of Louisville, 2015

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
Statistics

The University of New Mexico

Albuquerque, New Mexico

May, 2022

Dedication

To my son, husband, siblings and parents.

Acknowledgments

I would like to thank my co-advisors, Professors James Degnan and Guoyi Zhang, for their patience, teaching and advice during these projects. Both of them have taught me a great deal about the methods studied here as well as the overall process of conducting research. Both have pushed me to gain new experiences and provided frequent encouragement and valuable feedback at every stage. James and Guoyi are both extremely knowledgable and helped ensure that I presented our research and results accurately and appropriately, while showing kindness and empathy. Additionally, I thank my committee members, Professors Jeffrey Long and Fletcher Christensen, for their willingness to serve on my committee as well as for their assistance and feedback on this research. Many of the UNM Statistics faculty members provided helpful feedback on this work, and all of them have been excellent teachers during my time in this program. I would also like to thank my friends and family for their support during these last few years.

Measurement Error Modeling Applied to Phylogenetic Inference and Parametric Bootstrap Approach to Multifactor ANOVA Models with Unequal Variances and Unbalanced Data

by

Sarah K. Alver

B.S., Nutrition and Food Science, Utah State University, 2001
M.S., Biostatistics, University of Louisville, 2015

Ph.D., Statistics, University of New Mexico, 2022

Abstract

This dissertation includes two main topics. The first uses measurement error modeling to improve upon an existing method of inferring species trees from gene trees that were estimated with error. The second involves extending the parametric bootstrap (PB) approach, which was previously shown to work well for one-and two-way analysis of variance models with unequal variance and unbalanced data (heteANOVA), to multi-factor heteANOVA models. An overall framework using PB is presented.

For each topic, the underlying theory is shown, and simulations and applications to empirical data are presented, demonstrating improvement over earlier methods. The proposed species tree inference method shows that species tree inference can be improved in the presence of gene tree estimation error, and the new method may be useful for inferring starting trees for other possibly slower methods. The PB methods developed here provide a viable alternative to transforming data to meet the equal variance assumption.

Contents

List of Figures

List of Tables

List of Tables

Chapter 1

Introduction

The topics covered in this work are organized into three main chapters, with the second chapter pertaining to improvement of a phylogenetic inference method, and the third and fourth chapters pertaining to a parametric bootstrap (PB) approach to analysis of variance (ANOVA) models with unequal variances and unbalanced data.

1.1 Measurement Error Modeling Applied to Phylogenetic Inference

Chapter 2 deals with issues in inferring phylogenetic trees, which depict evolutionary relationships between species. Species trees show the divergence of species over time from a common ancestor to several extant species. These can be inferred through gene trees, which show the coalescence of genes between species moving backward in time. As described by Maddison (1997), gene trees represent copies of a gene at a locus that are passed on to more than one offspring; since the gene copy (in general) has a single ancestral copy, the resulting history is a branching tree. If a gene copy is

sampled from several species, the gene tree relates these gene copies. A species tree depicts the pattern of branching of species lineages from reproductive communities split by speciation and can contain many gene trees. The contained gene trees may differ from each other and from the species tree, even if the gene trees are known without error.

Gene trees can be estimated through analysis of DNA sequences; methods for doing so are discussed, for example by Felsenstein (2004), and implemented in programs such as `dnamlk` in the software package PHYLIP (Felsenstein, 2009). However, there is some error in the estimation of these gene trees (in addition to discordance between gene trees and species trees due to the coalescent process). There are many methods which in turn use gene trees to infer species trees.

The focus of this work is improvement upon a method called Global LAteSt Split (GLASS)/Maximum Tree (MT) (two equivalent methods that were developed by separate authors (Mossel and Roch, 2010; Liu et al., 2010b)), which uses single-linkage clustering of minimum pairwise coalescence times between species. The use of minimum coalescence times is justified because pairwise coalescence times of two genes from different populations tend to overestimate species divergence times (Mossel and Roch, 2010). Also, the tree returned by these methods contains the maximum species divergence times that are possible with the assumption that gene split times predate speciation times. This assumption and related constraints on speciation times are satisfied by choosing the minimum pairwise coalescent times over loci and using single linkage clustering (Liu et al., 2010b).

The method is implemented in software such as STEM (Species Tree Estimation using Maximum likelihood (Kubatko et al., 2009)). This method of species tree inference performs well and is statistically consistent when inferring a species tree from known gene trees. Additionally, STEM estimates the maximum likelihood (ML) species tree from a sample of gene trees, assuming that discord between the observed

gene trees and the species tree arises solely from the coalescent process (Kubatko et al., 2009), and that the population scaled mutation rate θ is the same for each population (Kubatko et al., 2009; Liu et al., 2010b). Unfortunately, STEM has been shown to perform relatively poorly when the input is gene trees estimated from DNA sequences, and sufficient conditions for statistical consistency in this case can be unrealistic (DeGiorgio and Degnan, 2014). One possible reason for the relatively poor performance is that branch lengths of zero can occur in ML estimated gene trees when DNA sequences from two species are identical at a locus. In that case, STEM 2.0 chooses the minimum non-zero distance over loci, thus possibly overestimating the true species divergence time. This issue, along with differences in the way zeros are handled by STEM 1.1 and STEM 2.0, are discussed in detail by DeGiorgio and Degnan (2014). A modification to STEM is proposed here which aims to improve the method in the presence of gene tree estimation error (GTEE).

Several methods have been developed that attempt to address the various sources of error in species tree inference, such as gene tree heterogeneity and incomplete lineage sorting, but a challenge has also been to account for the error in estimating gene trees from DNA sequences. In this work, we develop a method of species tree inference that attempts to address this challenge through measurement error modeling. This method uses an estimated distribution for true gene trees, in which that distribution is either simulated after estimating its parameters from estimated gene tree data using method of moments, or obtained through Bayesian inference, and then realizations of that estimated distribution are used to infer a species tree. A goal of this measurement error method is to improve species tree inference when the input is gene trees that are estimated, possibly with error, from DNA sequences.

The method developed here is an application of the method of clustering in general measurement error models described by Su et al. (2018). The proposed method replaces the estimated pairwise coalescence times used by STEM with randomly gen-

erated realizations from the estimated distribution of the true pairwise coalescence times. This distribution is estimated through additive measurement error modeling or through Bayesian inference. As with STEM, the minimum of these realizations, or their Bayesian posterior iterates, is taken over all loci for each pairwise distance. These minimums then form a distance matrix, and single linkage clustering is performed to infer the species tree.

Our simulation studies find that the new methods outperform STEM in terms of Robinson-Foulds (RF) distance (Robinson and Foulds, 1981) from the true species tree. The RF distance provides a numerical way to describe the similarity between the topology of two phylogenetic trees by counting the number of clades that occur on one tree and not the other, thus two trees with matching topology would have an RF distance of 0, and higher numbers indicate less similar topologies. The `treedist` program in the PHYLIP package was used to calculate RF distances in this study (Felsenstein, 2009).

When applied to real data, the methods developed here do outperform STEM, but do not always outperform ASTRAL, another popular method of species tree inference (Zhang et al., 2018) (ASTRAL does not employ a distance matrix like STEM does). The Bayesian version of the method developed here requires more computation time and is prone to convergence issues, but is more accurate than the additive measurement error model version in some cases. A potential use for the new method could be to obtain starting trees for other possibly slower methods that search over tree space for optimal trees.

1.2 PB Approach to ANOVA Models with Unequal Variance

Chapter 3 addresses the issue of unequal variance in analysis of variance models with unbalanced data (heteANOVA) using a parametric bootstrap (PB) approach. This involves simulating a null distribution for a test statistic for which a standard distribution is not known. The issue of the unmet equal variance assumption in multi-factor ANOVA has been addressed in the literature with several methods, and parametric bootstrap (PB) has been found in the one-way and two-way cases to outperform other methods. Krishnamoorthy and Lu (2007) studied the PB procedure for the one-way heteANOVA model and compared the results to James' test (James, 1951), Welch's test (Welch, 1951), and the generalized F-test (Weerahandi, 1995), and found the PB procedure to be one of the best for controlling the Type I error rate, particularly for larger numbers of factor levels. Yigit and Gökpınar (2010) also compared several methods for dealing with one-way heteANOVA models and found the PB test to work well. Xu et al. (2013) extended the PB test to the two-way heteANOVA model and compared it with the generalized F-test for two-way ANOVA models (Ananda and Weerahandi, 1997), and found that the PB test outperformed the generalized F-test in terms of Type I error when the number of factorial combinations or treatments increases.

We extend the PB procedures of Krishnamoorthy, Xu and Zhang (Xu et al., 2013; Krishnamoorthy and Lu, 2007; Zhang, 2015a,b) to models with at least three factors and illustrate with a three-way ANOVA model with unequal group variances and unbalanced data. Additionally, an overall framework is given for testing model parameters using this PB approach, including the PB approach to multiple comparison procedures (MCP). Pairwise MCP with unequal group sizes and/or variances was previously developed by Games and Howell (1976) but they note that the procedure

is limited with smaller sample sizes. Tukey's test was extended to unequal group sizes by Kramer (1956); this method was later proven by Hayter (1984) to be conservative. MCP was also previously addressed for one-way and two-way heteANOVA models by Zhang (2015a,b) using PB methods; here the PB approach to MCP is extended to at least three factors, incorporating multiple comparison procedures into the overall framework of testing the models. While caution is warranted in interpreting hypothesis testing conclusions when treatment group variances are unequal, this issue still comes up in practice, so methods to address it without the need for transformation of the observations are desirable. Transformation of the data can make interpretation of the results more complicated, so avoiding this can simplify the process of analyzing ANOVA data. The PB methods here are analogous to usual multi-factor ANOVA procedures, with F-tests and Tukey's MCP (Kutner et al., 2005; Kramer, 1956; Hayter, 1984), replaced by PB procedures. Using simulation, we compare these methods to F-tests for each step in model selection, as well as to Tukey's test for MCP. The results of the simulations indicate that the PB methods outperform F-tests and Tukey's test in terms of Type I error when group variances are unequal and data are unbalanced. An example dataset is analyzed, using both traditional methods (F-tests and Tukey's test) and the PB methods, to demonstrate use of the PB methods and compare them to traditional methods.

Chapter 4 applies the PB approach to a special case of the multiple comparison procedures. In one-way ANOVA models, it is sometimes of interest to perform simultaneous multiple comparisons of treatment groups with a control group, rather than performing all pairwise comparisons of the groups. Dunnett's test is used for such comparisons (Dunnett, 1955). The assumptions of ANOVA and of Dunnett's test require that the variance of the outcome of interest is the same for each group. However, this assumption is not always met in practice even after transformation, and as noted earlier, transforming the data can make interpreting results more complicated. In this research, we developed a PB method for comparing multiple treatment group

means against the control group when the constant variance assumption is violated and data are unbalanced, i.e., a parametric bootstrap analog of Dunnett's test. The simulation studies performed here under various settings show that the proposed method outperforms Dunnett's test in controlling the Type I error and does not suffer from loss of power when the treatment group variances are unequal and particularly with unbalanced data. An example using real data is presented to illustrate usage of the proposed method.

Each of Chapters 2 – 4 contains a more detailed introduction specific to those topics. The final chapter summarizes the conclusions made from studying these methods and discusses potential future areas of study.

Chapter 2

Species Tree Inference Using Measurement Error Modeling

2.1 Introduction and Literature Review

The GLASS/MT (Mossel and Roch, 2010; Liu et al., 2010b) method is one way to infer species trees from gene trees, and performs well when the input are true gene trees. As discussed in Mossel and Roch (2010) and Liu et al. (2010b), these methods estimate species trees from multiple genes and are statistically consistent under the multispecies coalescent model (MSC) when the input contains correct estimates of coalescence times. They are also computationally efficient compared to some other methods. When the gene trees are estimated with sufficient accuracy, GLASS/MT remains a statistically consistent estimator of the species tree. However, this theoretical advantage is lost when gene trees are estimated with error (Roch and Warnow, 2015). A sufficient condition for the accuracy of estimated gene trees needed to retain statistical consistency of the method is given in Mossel and Roch (2010): that the absolute differences between the true and estimated coalescence times for all loci

are less than half the shortest branch length in the species tree. Unfortunately, as shown by DeGiorgio and Degnan (2014), this condition fails to be satisfied with increasing numbers of loci and bounded sequence lengths for gene trees estimated from DNA sequences using maximum likelihood. It was also shown through simulations that this method as implemented in STEM (Kubatko et al., 2009) performed relatively poorly compared to other methods (DeGiorgio and Degnan, 2014; Leaché and Rannala, 2011; Wu, 2012). A review of several challenges to species tree inference, including gene tree estimation error (GTEE) is given in Mirarab et al. (2021).

Previous efforts have been made to quantify and correct for GTEE in summary methods (i.e. those that combine information from multiple loci to infer species trees) other than STEM; to our knowledge similar correction attempts have not been made for STEM/GLASS though a study of the potential sources of error is described below. A correction to the bias in the GLASS tree (iGLASS) was derived by Jewett and Rosenberg (2012), but this addresses the systematic overestimation of the species divergence time due to interspecific gene coalescences occurring more anciently than the divergence time under the MSC, rather than attempting to correct for GTEE. Some examples of these error quantification and error correction studies follow.

One such study by Huang et al. (2010) examines sources of error in species tree inference: they quantify error from coalesence vs. mutation by inferring species trees using STEM and minimizing deep coalescence (MDC), with both true and estimated gene trees as input, and then comparing the differences in Robinson-Foulds distance from true species trees. In this manner, they could attribute error from true gene trees to the coalescent process, error from estimated gene trees to both coalescence and mutation, and the difference to mutation. They found that the error due to mutation increased with sampling both more individuals and more loci, particularly with STEM. They note that errors attributed to the coalescent decrease with increased sampling, but discord from mutation persists, possible further illustration of

statistical consistency with correct gene trees but not in the presence of GTEE.

Additionally, Mirarab et al. (2014) looked at statistical binning, a procedure which groups gene trees that are not highly conflicting into "bins", concatenates sequences from each bin into a "supertree" and then applies a summary method of choice to these supertrees. They noted that it improved another method, Maximum Pseudolikelihood Estimation of Species Trees (MP-EST) (Liu et al., 2010a) in some cases, but did not look at this procedure with STEM. MP-EST (not binned) was found in simulation studies to outperform STEM by DeGiorgio and Degnan (2014).

Malloy and Warnow (2018) consider gene filtering to possibly improve the data quality of estimated gene trees; that is, removing genes from a dataset based on criteria such as low bootstrap support. They discuss low phylogenetic signal, which may result from shorter sequence lengths or low rates of evolution, as possible contributors to low bootstrap support and high GTEE. They did find improvement in accuracy of several methods by gene filtering for some levels of incomplete lineage sorting (ILS) and GTEE, but not if the number of remaining genes became too small, noting that balance is needed between quantity and quality of input gene trees. Mirarab et al. (2014), point out that "restricting loci is problematic for statistically consistent coalescent-based summary methods, because the conditions under which they are guaranteed to be accurate (with high probability) require a large enough random sample of true gene trees; removing loci can violate this condition and potentially bias the analysis." The idea of shorter sequence lengths contributing to GTEE is also discussed in Roch et al. (2019).

The aim of the current study is to address the effects of GTEE on the STEM/GLASS tree through measurement error modeling, to improve its accuracy in practical application while also providing theoretical support through maintaining statistical consistency. Some of the findings of the above previous studies, such as shorter sequence length and lower rates of evolution (at least through lower per-site muta-

tion rates) contributing to GTEE and thus less accurate species tree estimates, are echoed in our findings.

2.1.1 STEM and Measurement Error Modeling

The GLASS/Maximum tree estimates species trees from multiple loci using the minimum pairwise species coalescence time among loci, as coalescence times from gene trees overestimate species divergence times (Mossel and Roch, 2010; Liu et al., 2010b), and then uses single linkage clustering of those minimums to infer the species tree. The use of minimum coalescence times is justified as described in Chapter 1 and by the multispecies coalescent (MSC) model, in which times to coalescence from two genes sampled in different species have a shifted exponential distribution, where the shift parameter is the unknown species divergence time (Rannala and Yang, 2003). This method is implemented in software such as STEM 2.0 and STEM 1.1 (Kubatko et al., 2009). The simulations performed in this study compare results to those obtained from STEM 2.0.

As an example of the GLASS/STEM procedure, consider the following two gene trees, shown in Figure 2.1.

Gene tree 1: $((1:0.003, 3:0.003):0.02, (2:0.02, 4:0.02):0.003)$

Gene tree 2: $((4:0.004, 1:0.004):0.01, (3:0.009, 2:0.009):0.005)$

For GLASS as implemented in STEM 2.0, these branch lengths are divided by the population scaled mutation rate θ. For this example, suppose $\theta = 0.01$. Then the distances are taken to be those shown in Table 2.1. Dividing the branch lengths by $\theta/2$ converts these lengths from mutation units (the expected number of mutations per site per generation) to coalescent units (the number of generations per effective population size N_e), shown in Table 2.2.

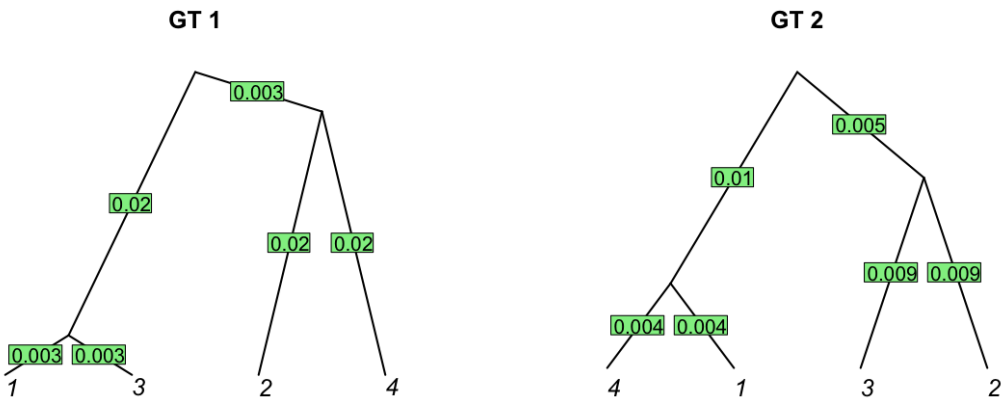

Figure 2.1: Example Gene Trees.

Pair (j, j')	$(1,2)$	$(1,3)$	$(1,4)$	$(2,3)$	$(2,4)$	$(3,4)$
Locus 1	2.300	0.300	2.300	2.300	2.000	2.300
Locus 2	1.400	1.400	0.400	0.900	1.400	1.400

Table 2.1: Pairwise distances with branch lengths divided by θ.

Pair (j, j')	$(1,2)$	$(1,3)$	$(1,4)$	$(2,3)$	$(2,4)$	$(3,4)$
Locus 1	4.600	0.600	4.600	4.600	4.000	4.600
Locus 2	2.800	2.800	0.800	1.800	2.800	2.800

Table 2.2: Pairwise distances with branch lengths divided by $\theta/2$.

These can also be shown in distance matrix form, $D_{jj'}^{(i)}$ for locus i and species j:

$$D_{jj'}^{(1)} = \begin{bmatrix} 0.000 & 4.600 & 0.600 & 4.600 \\ & 0.000 & 4.600 & 4.000 \\ & & 0.000 & 4.600 \\ & & & 0.000 \end{bmatrix} \quad D_{jj'}^{(2)} = \begin{bmatrix} 0.000 & 2.800 & 2.800 & 0.800 \\ & 0.000 & 1.800 & 2.800 \\ & & 0.000 & 2.800 \\ & & & 0.000 \end{bmatrix}$$

For GLASS/STEM, the minimum over loci (minimum non-zero distance for STEM 2.0, see (DeGiorgio and Degnan, 2014) for more discussion of this vs. STEM 1.1) is then taken for each pairwise distance to form a distance matrix of the minimum pairwise distances scaled by θ:

$$D_{jj'}^{(\min)} = \begin{bmatrix} 0.00 & 1.40 & 0.30 & 0.40 \\ & 0.00 & 0.90 & 1.40 \\ & & 0.00 & 1.40 \\ & & & 0.00 \end{bmatrix}$$

Single linkage clustering is then applied. In this case, species 1 and 3 have the shortest distance between them, forming the clade (1:0.3, 3:0.3). The next species that is closest to either 1 or 3 is species 4. So 4 is then grouped with the (1, 3) clade, giving ((1:0.3, 3:0.3):0.1, 4:0.4). Then the smallest distance from any of 1, 3 or 4 to 2 is 0.9. So the GLASS tree/MT is (((1:0.3, 3:0.3):0.1, 4:0.4):0.5, 2:0.9), which is returned by STEM and shown in Figure 2.2.

In order to clarify the steps for implementing and programming the proposed method, which will be shown in the next section, we present this procedure in a slightly different way. Note that the pairwise distances (hereafter referred to as W) divided by $\theta/2$ as above give the distance matrix:

$$D_{jj'}^{(\min)} = \begin{bmatrix} 0.00 & 2.80 & 0.60 & 0.80 \\ & 0.00 & 1.80 & 2.80 \\ & & 0.00 & 2.80 \\ & & & 0.00 \end{bmatrix}$$

Applying single linkage clustering as implemented in the `hclust` function in R, consider that Species 1 and 3 are a distance of 0.6 apart, again forming the clade (1:0.3, 3:0.3). Taking the next smallest distance from this first clade with 1 and 3, note that Species 4 is a distance of 0.8 from Species 1, so its total distance is 0.8 from the first clade (1,3). This produces (4:0.4, (1:0.3, 3:0.3):0.1). Similarly, Species 2 is a total distance of 1.8 from all three other species. Thus, the tree (2:0.9, (4:0.4, (1:0.3, 3:0.3):0.1):0.5), shown in Figure 2.2, is obtained as before.

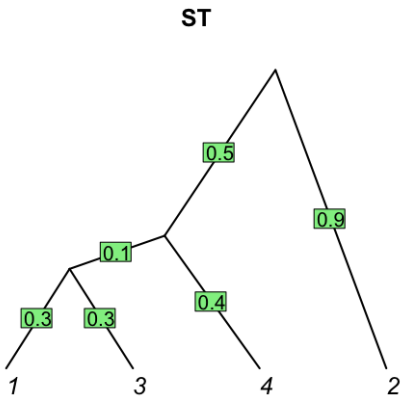

Figure 2.2: Example Species Tree Returned by STEM.

2.1.2 Measurement Error

The classical additive measurement error model is discussed in Carroll et al. (2006). In this model, X is an unobserved variable, W is an observed surrogate for X, and

U represents the error, so that $W = X + U$ with X and U independent. This model assumes that the expected value of U is 0. The application described here can also employ a multiplicative error model where $W = XU$, $E(U) = 1$ and X and U are independent. A particular distribution for the errors is not necessarily assumed.

2.1.3 Clustering in General Measurement Error Models

The methods proposed in this paper were inspired by the algorithm discussed by Su et al. (2018), in which values of the unobserved variable X are simulated from an estimated distribution for them and then clustered. The estimated distribution for X is based on measurement error analysis. This algorithm was formed to address the goal of performing the clustering so that in large samples, it reproduces the clusters that would have been formed had the latent variable actually been observed (Su et al., 2018) . As noted previously, we consider the classical additive measurement error model setup as well as the multiplicative error model and Bayesian approaches discussed in (Carroll et al., 2006) (in particular, see Chapter 2, Chapter 9 and Section *12.1.6.2*).

The rest of this chapter is organized as follows. In Section 2, the GLASS tree as implemented by STEM is modified through applying Su, Reedy and Carroll's method and a related Bayesian approach. Section 3 describes simulations performed to test these new methods, with comparison to STEM 2.0, results of those simulations, and application of the methods to a real dataset. Section 4 includes discussion of the results, limitations and areas for further research.

2.2 GenX and Bayesian Methods

In this section, the method of Su et al. (2018) is applied to the problem of clustering pairwise distances obtained from gene trees estimated with error, when the true gene trees are not observed. As with the classical additive measurement error model, let $W = X + U$, where W are estimated pairwise distances between species (coalescent times estimated from observed DNA sequences), X are the true pairwise distances (unobserved coalescence times from true gene trees) and U is error. Assume $\mathrm{E}(U) = 0$ and that X and U are independent, so $\mathrm{E}(W) = \mathrm{E}(X)$ and $\mathrm{Var}(W) = \mathrm{Var}(X) + \mathrm{Var}(U)$. We also consider, particularly for the Bayesian approach, a multiplicative error model where $W = XU$ with X and U independent, and $\mathrm{E}(U) = 1$, so again $\mathrm{E}(W) = \mathrm{E}(X)$. For the classical additive measurement error model, one feature is that $\mathrm{Var}(W) > \mathrm{Var}(X)$. This seemed to be true in the simulated data with $W \div (\theta/2)$. Following the MSC model, X is shifted exponential with unknown location parameter τ, where X and τ are in coalescent units (Rannala and Yang, 2003; DeGiorgio and Degnan, 2014; Wakeley, 2009). No specific distribution is assumed for U or W, except in the Bayesian approach where a sampling distribution is assumed for W.

When $\mathrm{E}(W) = \mathrm{E}(X)$ as above, since the rate parameter for X is 1, the sample mean of the estimated pairwise distances, $\overline{W}$, can be used to obtain a Method of Moments estimator of τ and simulate the true pairwise distances X:

$$\mathrm{E}(\overline{W}) = \mathrm{E}(W) = \mathrm{E}(X) = \tau + 1 \implies \widehat{\tau} = \overline{W} - 1;\ f_X(x) = e^{-(x-\tau)}, x > \tau$$

Because $z = (x - \tau) \sim \mathrm{Exp}(1)$, generating random $z + \widehat{\tau}$ simulates values of the true pairwise distances X; hereafter this method will be called genX. A limitation is that in the simulated data, the assumption of $\mathrm{E}(U) = 0$ for the additive model or $\mathrm{E}(U) = 1$ for the multiplicative model only appeared to be true for higher values of θ and longer DNA sequence lengths. However, the genX method seems to perform well even when these sample error means are not near 0 (additive model) or 1

(multiplicative model).

Implementation Procedure

- Calculate all the pairwise distances $W \div \theta/2 = 2W/\theta$ in a set of estimated gene trees. The estimated branch lengths $\boldsymbol{W}$ are now contained in an $n \times \binom{n_s}{2}$ matrix, where n is number of loci and n_s is number of species.
- Take the mean $\overline{W} - 1$ for each set of pairwise distances — in this implementation, this is the mean of each column of $\boldsymbol{W} - 1$, giving a $\widehat{\tau}$ for each pairwise distance.
- Create $\widetilde{X}$, a matrix the same size as $\boldsymbol{W}$. Each entry of $\widetilde{X}$ is a randomly generated exponential random variable with rate 1, plus the $\widehat{\tau}$ corresponding to its column.
- Take the minimum of each column of $\widetilde{X}$ to form a distance matrix and perform single linkage clustering (as with STEM) to estimate the species tree.

Returning to the previous example, and taking $2W/\theta$, $\widehat{\tau} = \overline{W} - 1$ for each column of the $\boldsymbol{W}$ matrix, which has the same dimensions and entries as Table 2. In this case, the $\boldsymbol{W}$ matrix is:

$$\boldsymbol{W}_{ik} = \begin{bmatrix} 4.600 & 0.600 & 4.600 & 4.600 & 4.000 & 4.600 \\ 2.800 & 2.800 & 0.800 & 1.800 & 2.800 & 2.800 \end{bmatrix}$$

We then generate a matrix $\mathbf{Z}$ of Exp(1) random variables and add the value $\widehat{\tau}$ from the appropriate column to the $\mathbf{Z}$ matrix to obtain the matrix $\widetilde{X}$. Then $\widetilde{X}$ is replacing

Pair (j, j')	$(1,2)$	$(1,3)$	$(1,4)$	$(2,3)$	$(2,4)$	$(3,4)$
Matrix column k	1	2	3	4	5	6
$\hat{\tau}$	2.7	0.7	1.7	2.2	2.4	2.7

$$\widehat{\tau} = \overline{W} - 1$$

the estimated, observed surrogate $\boldsymbol{W}$ with randomly generated observations from the distribution of the latent variable $\boldsymbol{X}$.

$$\boldsymbol{Z}_{ik} = \begin{bmatrix} 0.755 & 0.146 & 0.436 & 1.230 & 0.957 & 1.391 \\ 1.182 & 0.140 & 2.895 & 0.540 & 0.147 & 0.762 \end{bmatrix}$$

$$\widetilde{\boldsymbol{X}}_{ik} = \begin{bmatrix} 3.455 & 0.846 & 2.136 & 3.430 & 3.357 & 4.091 \\ 3.882 & 0.840 & 4.595 & 2.740 & 2.547 & 3.462 \end{bmatrix}$$

As with the GLASS/STEM method, a distance matrix is formed from the minimums for each pair in the $\widetilde{X}$ matrix.

$$D_{jj'}^{(\min)} = \begin{bmatrix} 0.000 & 3.455 & 0.840 & 2.136 \\ & 0.000 & 2.740 & 2.547 \\ & & 0.000 & 3.462 \\ & & & 0.000 \end{bmatrix}$$

Here Species 1 and 3 are closest, with a distance of 0.84 apart, giving the clade (1:0.42, 3:0.42), followed by Species 4 being 2.136 away from the first clade (1,3), giving (4:1.068, (1:0.42, 3:0.42): 0.648), and then the final smallest distance for Species 2 is 2.547. Thus, (2:1.274, (4:1.068, (1:0.42, 3:0.42):0.648):0.206) is the inferred species

tree. In this case the genX tree happens to have the same topology as the STEM tree.

Code for the procedure in R (R Core Team, 2021) using the packages `ape` (Paradis and Schliep, 2019) and `gdata` (Warnes et al., 2017) is shown in Appendix A.

2.2.1 Statistical Consistency

As discussed in Liu et al. (2010b), the GLASS/Maximum Tree, with input of estimated gene trees, is a consistent estimator of the species tree when the estimates of the gene trees are consistent. The following argument demonstrates that the genX procedure also consistently estimates the species tree when E(W) = E(X).

To illustrate this, we will show that $\min_i \widetilde{X}_i \xrightarrow{P} \min_i X_i$, where i is the index for loci, i.e. that $\lim_{n\to\infty} P(|\widetilde{X}_{(1)} - X_{(1)}| \geq \varepsilon) = 0$, where n is the number of loci. Note that $\widetilde{X}$ has the same distribution as $\widetilde{Z} + \widehat{\tau}$ where $\widetilde{Z} \sim \text{Exp}(1)$ and $\widehat{\tau} = \overline{W} - 1$. For this proof, $\widetilde{Z}$ indicates randomly generated values from an Exp(1) distribution as in the implementation procedure described above, and Z represents the Exp(1) portion of $X = Z + \tau$ from the true gene trees. By the Glivenko-Cantelli theorem as discussed in Su et al. (2018), the empirical distribution function converges uniformly almost surely to the true distribution function (Ferguson, 1996), so $\widetilde{Z}_{(1)} \xrightarrow{P} Z_{(1)}$, and

$$\begin{aligned}
\lim_{n\to\infty} P(|\widetilde{X}_{(1)} - X_{(1)}| \geq \varepsilon) &= \lim_{n\to\infty} P(|\widetilde{Z}_{(1)} + \widehat{\tau} - X_{(1)}| \geq \varepsilon) \\
&= \lim_{n\to\infty} P(|\widetilde{Z}_{(1)} + \widehat{\tau} - (Z_{(1)} + \tau)| \geq \varepsilon) \\
&= \lim_{n\to\infty} P(|\widehat{\tau} - \tau| \geq \varepsilon) \\
&= \lim_{n\to\infty} P(|\overline{W} - 1 - \tau| \geq \varepsilon).
\end{aligned}$$

The last statement applies to both the additive and multiplicative model as both assume $E(W) = E(X)$. The remaining argument differs slightly for the two mod-

els, so the error U is denoted as U_a under the additive model and U_m under the multiplicative model.

For the additive model,

$$\begin{aligned}\lim_{n\to\infty} P(|\overline{W} - 1 - \tau| \geq \varepsilon) &= \lim_{n\to\infty} P(|\overline{X} + \overline{U}_a - 1 - \tau| \geq \varepsilon) \\ &= \lim_{n\to\infty} P(|\overline{Z} + \tau + \overline{U}_a - 1 - \tau| \geq \varepsilon).\end{aligned}$$

By the Weak Law of Large Numbers as shown in e.g. (Casella and Berger, 2002), $\overline{U} \xrightarrow{P} 0$ under the additive model, and $\overline{Z} \xrightarrow{P} 1$. By Theorem 2.1.3 in Lehman (1999), $\overline{Z} + \overline{U}$ converges in probability to 1 under the additive model.

For the multiplicative model,

$$\begin{aligned}\lim_{n\to\infty} P(|\overline{W} - 1 - \tau| \geq \varepsilon) &= \lim_{n\to\infty} P(|\overline{X}\ \overline{U}_m - 1 - \tau| \geq \varepsilon) \\ &= \lim_{n\to\infty} P(|(\overline{Z} + \tau)\overline{U}_m - 1 - \tau| \geq \varepsilon).\end{aligned}$$

By the Weak Law of Large Numbers as shown in e.g. (Casella and Berger, 2002), $\overline{U} \xrightarrow{P} 1$ for the multiplicative model, and $\overline{Z} \xrightarrow{P} 1$. By Theorem 2.1.3 in Lehman (1999), $\overline{Z}\ \overline{U}$ and $\tau\overline{U}$ converge in probability to 1 and τ respectively under the multiplicative model. Thus,

$$\lim_{n\to\infty} P(|\overline{Z} + \tau + \overline{U}_a - 1 - \tau| \geq \varepsilon) = \lim_{n\to\infty} P(|(\overline{Z} + \tau)\overline{U}_m - 1 - \tau| \geq \varepsilon) = 0,$$

and $\min_i \widetilde{X}_i \xrightarrow{P} \min_i X_i$. So genX is a consistent estimator of the species tree when $E(U_m) = 1$ or $E(U_a) = 0$.

In simulated data, this assumption of $E(U_m) = 1$ or $E(U_a) = 0$ only appeared to be approximately true for large θ and longer sequence lengths. However, the simulation results discussed in Section 3.2 show that the genX method still outperformed STEM even for smaller θ and shorter sequence lengths.

2.2.2 Bayesian Approach

Recall that the multiplicative measurement error model assumes $W = XU$ and $E(U) = 1$, with X and U independent. If we again assume that $(X - \tau) \sim \text{Exp}(1)$, and assume that $W|X$ follows a gamma distribution with mean X, the Bayesian model is a natural approach (Carroll et al., 2006). In this case, rather than randomly generating realizations of $\widetilde{X}$ from an estimated distribution, the estimated distribution for X is the set of Bayesian posterior iterates of X. Consider the hierarchical model:

$$
\begin{aligned}
&W|X \sim \text{Gamma}(\alpha, \beta) \\
&\alpha = X\beta, \text{so that } E[W|X] = X \\
&X = Z + \tau \\
&Z \sim \text{Exp}(1)
\end{aligned}
$$

Using the Bayesian hierarchical model above, one can infer a species tree from a set of estimated gene trees by fitting the model for each pairwise distance, and then taking either the mean of the posterior distribution for τ or the minimum of the posterior distribution for X over all loci. As with STEM and genX, these values form a distance matrix consisting of pairwise distances, and single-linkage clustering is applied to infer a species tree. Here this method is implemented using JAGS/rjags software (Plummer, 2003, 2021).

To elicit a prior for τ in the above model, a total of 100 species trees were simulated using the `sim.bd.taxa` function from the R package `TreeSim` (Stadler, 2019; R Core Team, 2021), with the speciation rate $\lambda = 1$ and five different tree sizes: 4, 8, 16, 22 or 32 species, 20 trees each. The extinction rate was kept at 0. A simulated distribution for τ was then obtained by calculating pairwise distances between all species in each tree. Then, similarly to methods of eliciting a normal prior used in Christensen et al. (2011), we used the mean of this distribution as the

mean of a normal prior, and then took the 95th percentile of the distribution to solve for a standard deviation, i.e. set $(\tau_{.95} - \overline{\tau})/\sigma = 1.645$. This procedure yielded a Normal$(2, 1)$ prior for τ, which was truncated at 0.

To elicit a prior for β in the above model, a distribution for $W|X$ was generated by simulating true and estimated gene trees from true species trees simulated with `TreeSim` as above. These were rooted with an outgroup and made ultrametric. True gene trees were simulated from the species trees using Hybrid-Lambda (Zhu et al., 2015), and sequences were simulated with Seq-Gen (Rambut and Grassly, 1997) under the F84 substitution model with transition transversion (TS/TV) ratio of 4.6 and base frequencies of 0.3, 0.2, 0.2 and 0.3 for A, C, G and T respectively. Then gene trees were estimated from the sequence data using the `dnamlk` program in PHYLIP (Felsenstein, 2009), with the TS/TV ratio and set of frequencies above specified in the `dnamlk` settings. Note that `dnamlk` was used so that the estimated gene trees satisfy the molecular clock, as this is an assumption for STEM. The substitution model settings were chosen to be similar to those in DeGiorgio and Degnan (2014). We then removed the outgroup from the estimated gene trees and obtained pairwise distances between species (as before, pairwise distances are denoted by X from true gene trees and by W from estimated gene trees).

For this purpose, 20-taxon trees were simulated with $\theta = 0.001$, sequence length of 500 nucleotides, and 500, 1000, 1500 or 2000 loci. We also included values for θ of 0.005 and 0.01 for 500 loci, but included more data with the lower value of θ to elicit this prior because the error U_m tended to be larger for smaller values of θ. Separate data were purposefully used to elicit the priors and then to infer trees and check the accuracy of the Bayesian approach.

Values of X were taken corresponding to the percentiles 0.05, 0.25, 0.5, 0.75, 0.95 of its distribution and to its mean, and then the distribution of W found for those fixed values of X (X within 0.001 of each of these percentiles). Note that $\text{Var}[W|X] =$

X/β in the Gamma$(X\beta, \beta)$ model, so an estimate of β can be calculated from the sample variance of W for a given X. In this manner, a simulated distribution for β was obtained and the above method again used to elicit a normal prior, obtaining $\beta \sim \text{Normal}(1.9, 3.8^2)$ and truncating at 0.

To help develop and initially test the Bayesian version, previously simulated data from 100 species trees with eight taxa were used; these had been simulated using `TreeSim` with $\lambda = 1$ and extinction rate 0 as above. From those, there were 50 true and estimated gene trees for each species tree, which were simulated using Hybrid-Lambda, Seq-Gen and the PHYLIP program `dnamlk`, with values of 0.003, 0.004, 0.006, 0.007 and 0.008 for the population-scaled mutation rate θ, and sequence lengths of 200, 500, and 1000 nucleotides. The smaller number of loci (50) made these data more convenient for testing the Bayesian method since it requires more computation time. The gene trees and DNA sequences were simulated with the same substitution model settings described above, and the values for θ and sequence lengths were chosen arbitrarily to represent a range between 0.001 and 0.01 for θ and a range between 200 and 1000 nt for the sequence length.

Convergence was assessed by examining trace plots and considering the convergence diagnostic $\widehat{R}$ as calculated by the `gelman.diag` function in the `rjags` R package (Plummer, 2021, 2003). According to the JAGS documentation, $\widehat{R} < 1.1$ is one indicator of convergence. Since this Bayesian approach can involve fitting a large number of models ($\binom{n_s}{2}$ for each species tree inferred, where n_s is the number of taxa), an automated method for checking convergence was desirable, but as noted, we also examined trace plots for the parameters, particularly for those with $\widehat{R} \geq 1.1$. Some convergence issues were noted for τ in terms of $\widehat{R}$, especially with larger θ and longer sequence length, but trace plots for τ were usually acceptable, in that no appreciable difference was seen between these plots and those corresponding to $\widehat{R}$ values below 1.1, for slightly high values (up to 1.19) of $\widehat{R}$. The set with $\theta = 0.007$

and sequence length of 1000 nt had the most convergence issues for τ (8/2800 models had $\widehat{R} \geq 1.1$), so for this set, the models were re-run with 40000 iterations and then no values of $\widehat{R}$ were 1.1 or greater.

Neither X nor β showed convergence issues in these models for any values of θ or sequence length, even with only 10000 iterations.

Sensitivity Analysis for Prior on τ

Because a wide range of species divergence times can be observed in real data, a sensitivity analysis was performed to determine whether the posterior estimates of τ and X would be strongly influenced by the prior for τ. We elicited an additional prior for τ by simulating a total of 300 species trees, again using the `sim.bd.taxa` function. In an attempt to capture the possible wide range of τ, three different settings were used for the speciation rate λ: 0.1, 0.5 and 1, with five different tree sizes: 4, 8, 16, 22 or 32 species, and 20 trees were simulated for each combination of these settings. The extinction rate was again kept at 0. Following the same procedure as above, a Normal$(9, 14^2)$ prior for τ was obtained and truncated at 0. The 100 species trees were then inferred from 50 loci each for two sets with lower and higher settings of θ and sequence length (0.003 and 200 nt for one set, and 0.007 and 1000 nt for the other set). These two sets were chosen from those described above to represent the least and most potential convergence issues, to help determine if the larger variance would adversely affect convergence. These models were used to calculate the difference between the means of the posterior distributions of τ from each prior. Inferring 100 species trees with eight taxa gave 2800 models from which to compare means of posterior iterates. In the same manner, the minimums of the posterior distributions of X were compared for each model.

For these datasets, the posterior distribution of X was not sensitive to the choice

of prior for τ, in that the average difference in minimums (smaller prior - larger prior) was -0.01 for the smaller θ and sequence length set and 0.001 for the larger θ and sequence length set. Only about 0.5% had a difference greater in magnitude than 0.5 and none with magnitude greater than 1 for the smaller θ/sequence length set, and the largest absolute difference for the set with larger θ/sequence length was 0.407. For the posterior distribution of τ, some sensitivity was noted for smaller values of θ and shorter sequence lengths, in that about 25.6% of the models had a difference greater in magnitude than 1, though the average difference was -0.675. For the larger θ and sequence length set, the average difference was 0.047, with only 2.9% having a difference greater than 0.5 in magnitude and 0.6% having a difference greater than 1 in magnitude. As expected, the prior on τ with larger variance did result in more convergence issues. For the dataset with smaller θ and sequence length, using the wider prior resulted in 33/2800 models with convergence issues for τ, and 36/2800 with convergence issues for X, with $\widehat{R}$ up to 1.357 for τ and 1.359 for X. Trace plots for these showed a small part of one chain diverging from the other and then coming back to converge with the other chain. For the dataset with larger θ and sequence length, no convergence issues for X or β were noted with the wider prior, but 26/2800 were noted for τ, with $\widehat{R}$ up to 1.246. For this model, the trace plot was not noticeably worse than for those with $\widehat{R}$ closer to or below 1.1. Another run with 40000 iterations and the wider prior on the set with $\theta = 0.007$ and sequence length of 1000 showed no convergence issues, with $\widehat{R}$ less than 1.1 for all models.

For both of these datasets, the mean Robinson-Foulds (RF) distance between the 100 trees inferred and their true species trees, calculated using the `treedist` program in PHYLIP and scaled by max RF of $2n_s - 4$ where n_s is the number of species, was very close to that obtained with the narrower prior despite the increase in potential convergence issues. With 10000 iterations and the narrower prior, the mean scaled RF distances for the set with smaller θ and sequence length were 0.352 and 0.335, for clustering $\min(X)$ and $\text{mean}(\tau)$ respectively, compared to 0.343 for

both $\min(X)$ and $\text{mean}(\tau)$ for 10000 iterations with the wider prior. For the set with larger θ and sequence length, these values were 0.175 and 0.25 (wider prior) vs 0.178 and 0.247 (narrow prior).

As the sensitivity to the prior was not severe, particularly for X, we elected to continue with the Normal$(9, 14^2)$ prior on τ for comparison to STEM and genX, since it may more closely reflect the values seen in practice. Because more potential convergence issues were noted with the wider prior, the higher number of iterations (40000) were used for subsequent comparisons.

2.3 Simulations and Results

2.3.1 Simulation Procedures

One hundred species trees with 8, 16, 20 or 32 taxa were randomly generated using the `sim.bd.taxa` function from the R package `TreeSim` (Stadler, 2019; R Core Team, 2021) with speciation rate $\lambda = 1$. From each of these species trees, using Hybrid-Lambda, Seq-Gen and `dnamlk` with substitution model settings as described in Section 2.2, we generated 250, 500 or 1000 estimated gene trees for each combination of settings with θ of $0.001, 0.005$, or 0.01 and sequence lengths of 500 or 1000 nucleotides. Due to computation time for simulating gene trees, only sequence lengths of 500 nucleotides were simulated for the 32-taxon trees.

Each set of estimated gene trees was then used to infer the species trees using the measurement error modified version of STEM (genX) and STEM 2.0 for comparison. For the Bayesian methods, the species trees were inferred for fewer tree sizes and simulation settings as described below due to increased computation time. The inferred trees were compared using the `treedist` program in PHYLIP, in terms of the scaled RF distance from the true species tree and percent of topologies matching

those of the true species trees.

Since the Bayesian methods developed here require considerably more computation time than GenX or STEM, only the first 50 out of 100 species trees were inferred, using the same simulated gene trees as for genX and STEM 2.0, and only for trees with eight or 20 taxa. Additionally, we only used the settings of 250 or 1000 loci and θ of 0.001 or 0.01 for the eight-taxon trees but used both sequence length settings of 500 and 1000 nt. For the 20-taxon trees, only settings of 250 loci, sequence length of 500 nt, and the two θ values of 0.001 or 0.01 were used. The simulation settings used for the Bayesian methods are listed in Table 2.3. Convergence was first checked using the Gelman diagnostic criteria $\widehat{R}$, and trace plots were checked for trees and models with $\widehat{R} \geq 1.1$. For the 8-taxon trees, with cases where less than 20 models with $\widehat{R} \geq 1.1$ were noted, trace plots were checked for all such models; otherwise trace plots were checked for the models with the three greatest values of $\widehat{R}$. This was done for the simulations due to the number of models fit for the simulation; e.g. for inferring a tree with eight species, the Bayesian method here fits $\binom{8}{2} = 28$ models for each tree inferred, so checking plots for all 1400 models fit for a simulation setting would be impractical. The 20-taxon trees were inferred in batches of 10 trees for the Bayesian methods, so in this case the trace plots for the highest three $\widehat{R}$ values from each batch (1900 models, i.e. $10 \times \binom{20}{2}$) were checked against a randomly chosen model from the same batch with $\widehat{R} < 1.1$.

In practice, at least for smaller trees, it may be preferable to first check convergence using more than one chain, determine the sufficient number of iterations needed to achieve convergence, and then re-running with one chain to avoid discarding data, thus obtaining a more precise estimate. As discussed earlier, when initially testing and developing the Bayesian method, some convergence issues were noted for τ when only 10000 iterations were performed, but not when the iterations were increased to 40000. Thus, all results for the Bayesian methods in the next section

used 40000 iterations and a burn-in of 4000 iterations unless otherwise noted.

2.3.2 Results

The results of the simulations are shown in Figures 2.3 – 2.13 and Tables 2.3 – 2.8. For Bayesian methods on 8-taxon trees, results shown in tables and figures are out of 50 species trees; for genX and STEM, results are out of 100 species trees. For the Bayesian methods on 20-taxon trees, results are briefly described in Section 3.2.2 since only limited settings were simulated for those trees. "Bayes X" refers to inferring a tree based on clustering the minimums of the posterior distributions of X, while "Bayes τ" refers to inferring a tree based on clustering the means of the posterior distributions of τ. The simulation settings that were used for the Bayesian methods are listed in Table 2.3 along with a summary of convergence assessment for the Bayesian models for those settings.

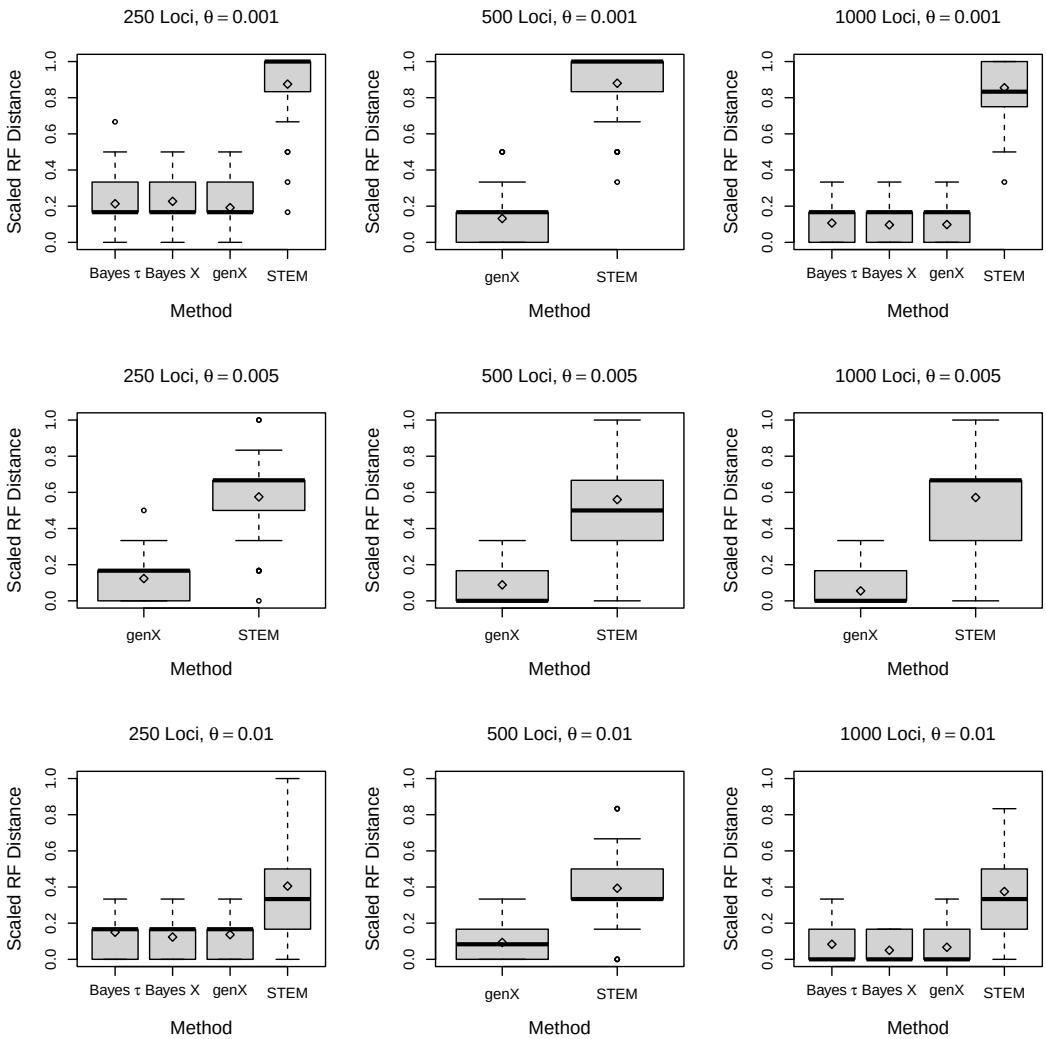

Figure 2.3: Scaled RF distances assuming rooted trees from true species trees for 8-taxon trees, 500 nt.

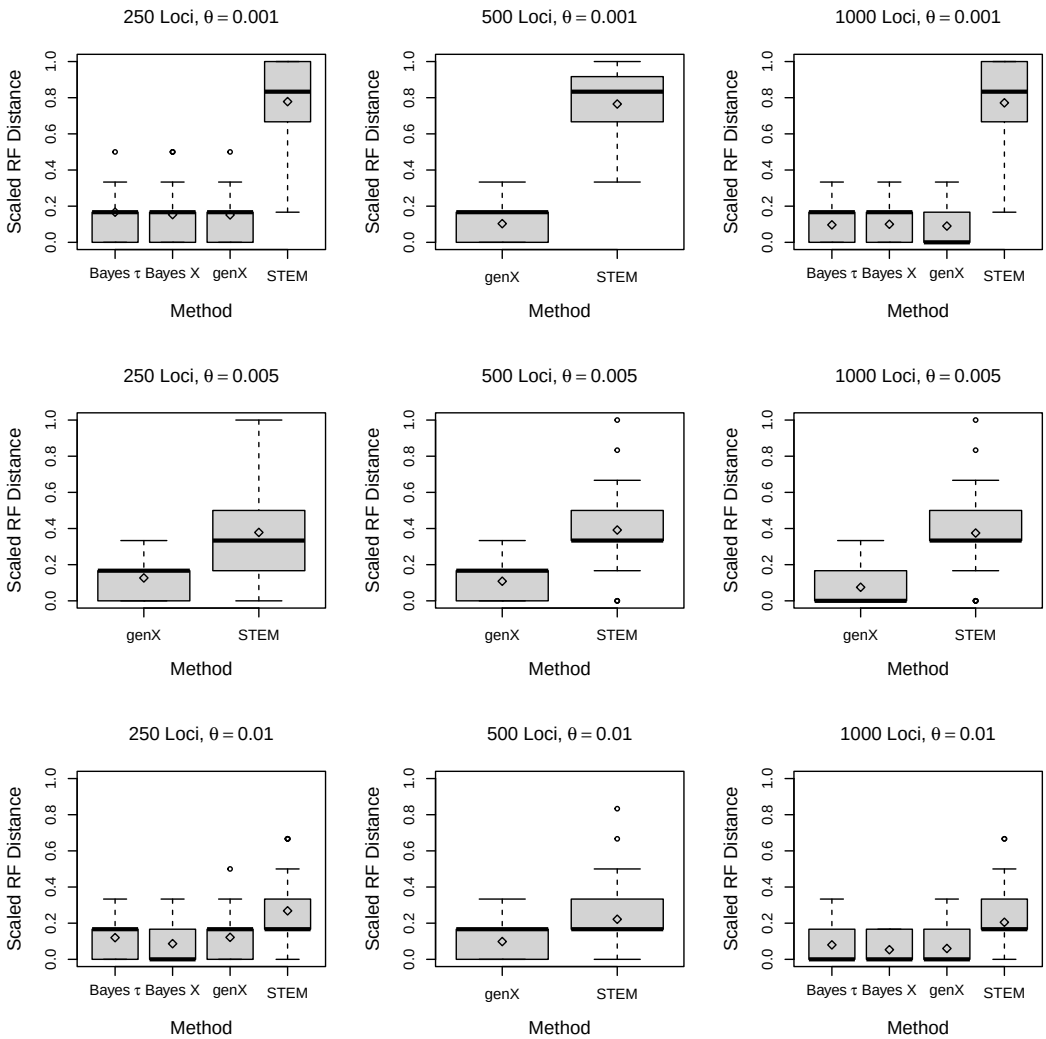

Figure 2.4: Scaled RF distances assuming rooted trees from true species trees for 8-taxon trees, 1000 nt.

Convergence of Bayesian Methods, 8-Taxon Trees

For 250 loci, no convergence issues were noted with $\theta = 0.001$ and 500 nt. For $\theta = 0.01$ and 500 nt, two models had $\widehat{R}$ of 1.11 and 1.15 for τ, but trace plots were examined for these models and found to be acceptable. For $\theta = 0.01$ and 1000 nt, one model had $\widehat{R}$ of 1.11, but again the trace plot was examined and found to be acceptable.

Similarly, for 1000 loci, with $\theta = 0.001$ and 500 nt, 13/1400 models were noted to have $\widehat{R} \geq 1.1$ with range 1.1 to 1.21 for τ, and no convergence issues with X or β. Trace plots were checked for these and found to be acceptable, with no appreciable difference from an arbitrarily chosen trace plot with $\widehat{R}$ of 1.01 from this set of models. For 1000 loci with $\theta = 0.001$ and 1000 nt, 11/1400 models had $\widehat{R} \geq 1.1$ with range 1.1 to 1.24; again the related trace plots appeared to be acceptable and not noticeably different from one with $\widehat{R}$ of 1.04.

For 1000 loci with $\theta = 0.01$ and 500 nt, 113/1400 models were noted to have $\widehat{R} \geq 1.1$, with range 1.1 to 1.51 for τ and no convergence issues with X or β. Trace plots were checked for the three models with the highest $\widehat{R}$ values (1.35, 1.41 and 1.51) and compared to that of an arbitrarily chosen model with $\widehat{R}$ of 1.01. The plot corresponding to the highest $\widehat{R}$ in this case was not noticeably worse, but the two trace plots corresponding to $\widehat{R} = 1.35$ and $\widehat{R} = 1.41$ did show some separation of the two chains.

For 1000 loci with $\theta = 0.01$ and 1000 nt, 133/1400 models were noted to $\widehat{R} \geq 1.1$, with range 1.1 to 1.54 for τ and no convergence issues with X or β. Trace plots were checked for the three models with the highest $\widehat{R}$ values (1.39, 1.41 and 1.54) and compared to that of a randomly chosen model with $\widehat{R}$ of 1.07. These did appear to show more separation of the two chains compared to the model with lower $\widehat{R}$, but not severely. These results are summarized in Table 2.3 along with those for 20-taxon trees.

Results and Convergence of Bayesian Methods, 20-Taxon Trees

For 250 loci, no convergence issues were noted with $\theta = 0.001$ and 500 nt. For 250 loci with $\theta = 0.01$ and 500 nt, there were 326/9500 models ($\binom{20}{2} = 190$ models for each of 50 trees inferred) which had values of $\hat{R}$ ranging from 1.1 to 1.402. The trace plots visualized for the highest values of $\hat{R}$ were again compared to those for randomly chosen models with $\hat{R} < 1.1$ and did not appear noticeably worse in terms of separation of the two chains. These results are summarized in Table 2.3 along with those for 8-taxon trees.

For $\theta = 0.001$, Bayes X had a scaled RF range of 0 to 0.5 with mean of 0.191 and 2% correct topology, where Bayes τ had range of (0, 0.444), mean of 0.172 and 4% correct topology. This is an improvement over STEM but not as good as genX for this setting.

For $\theta = 0.01$, Bayes X and Bayes τ both had a scaled RF range of 0 to 0.333 with mean of 0.074 and 0.107, respectively, and 34% and 22% correct topology, respectively. This is again an improvement over STEM, and comparable to or better than genX, which had range of (0, 0.278), mean of 0.089, and 22% correct topology for this setting.

n	Loci	θ	Seq. Length	Max $\widehat{R}$	Models w/$\widehat{R} \geq 1.1$
8	250	0.001	500	< 1.1	0/1400
8	250	0.01	500	1.15	2/1400
8	250	0.001	1000	< 1.1	0/1400
8	250	0.01	1000	1.11	1/1400
8	1000	0.001	500	1.21	13/1400
8	1000	0.01	500	1.51	113/1400
8	1000	0.001	1000	1.24	11/1400
8	1000	0.01	1000	1.54	133/1400
20	250	0.001	500	< 1.1	0/9500
20	250	0.01	500	1.40	326/9500

Table 2.3: Summary of Convergence Diagnostics of Bayesian Methods on Simulated Trees.

In these simulations, most trace plots for models with $\widehat{R} \geq 1.1$ for the parameter τ were not appreciably different in appearance from those with $\widehat{R} < 1.1$. The value n in the table refers to the number of taxa. Simulation settings for the Bayesian methods were limited to those above due to longer computation time.

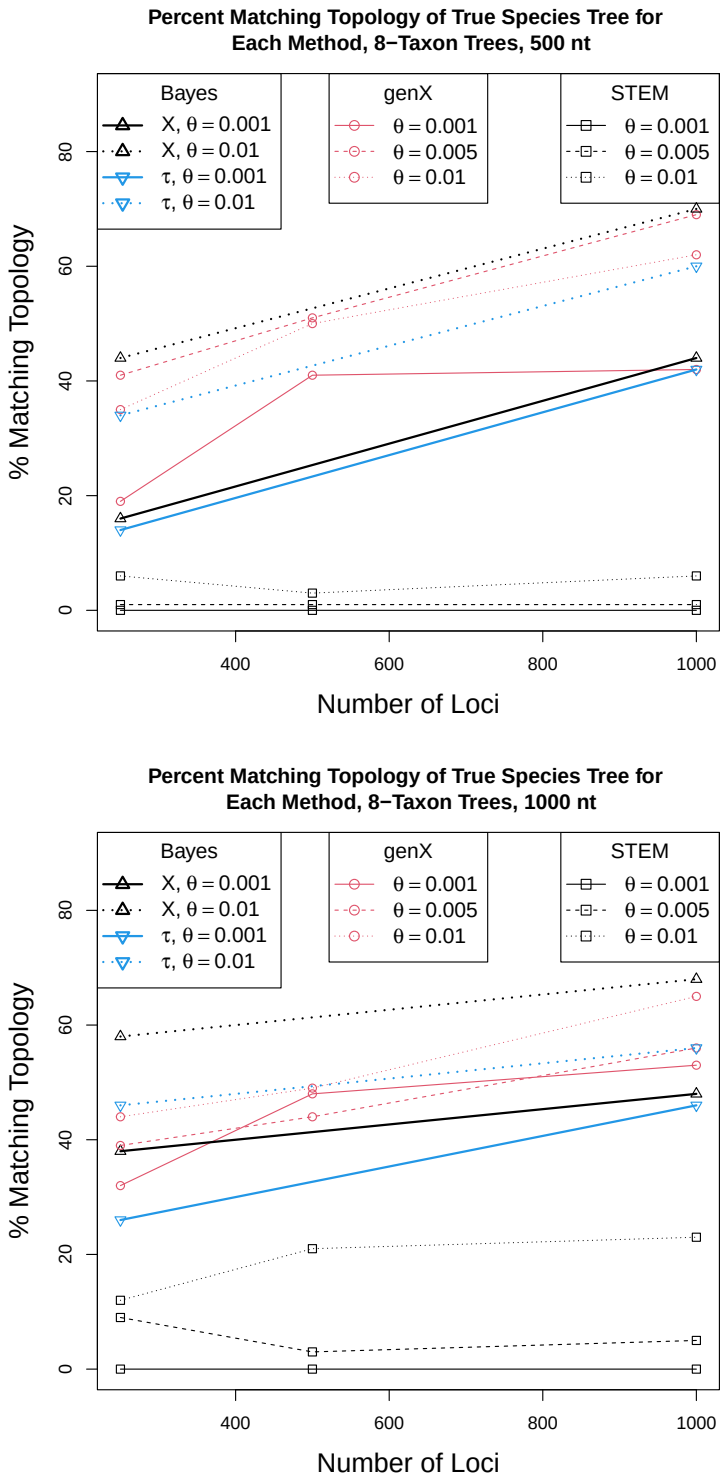

Figure 2.5: Percent of inferred rooted topology matching true species trees for 8-taxon trees, 500 nt or 1000 nt.

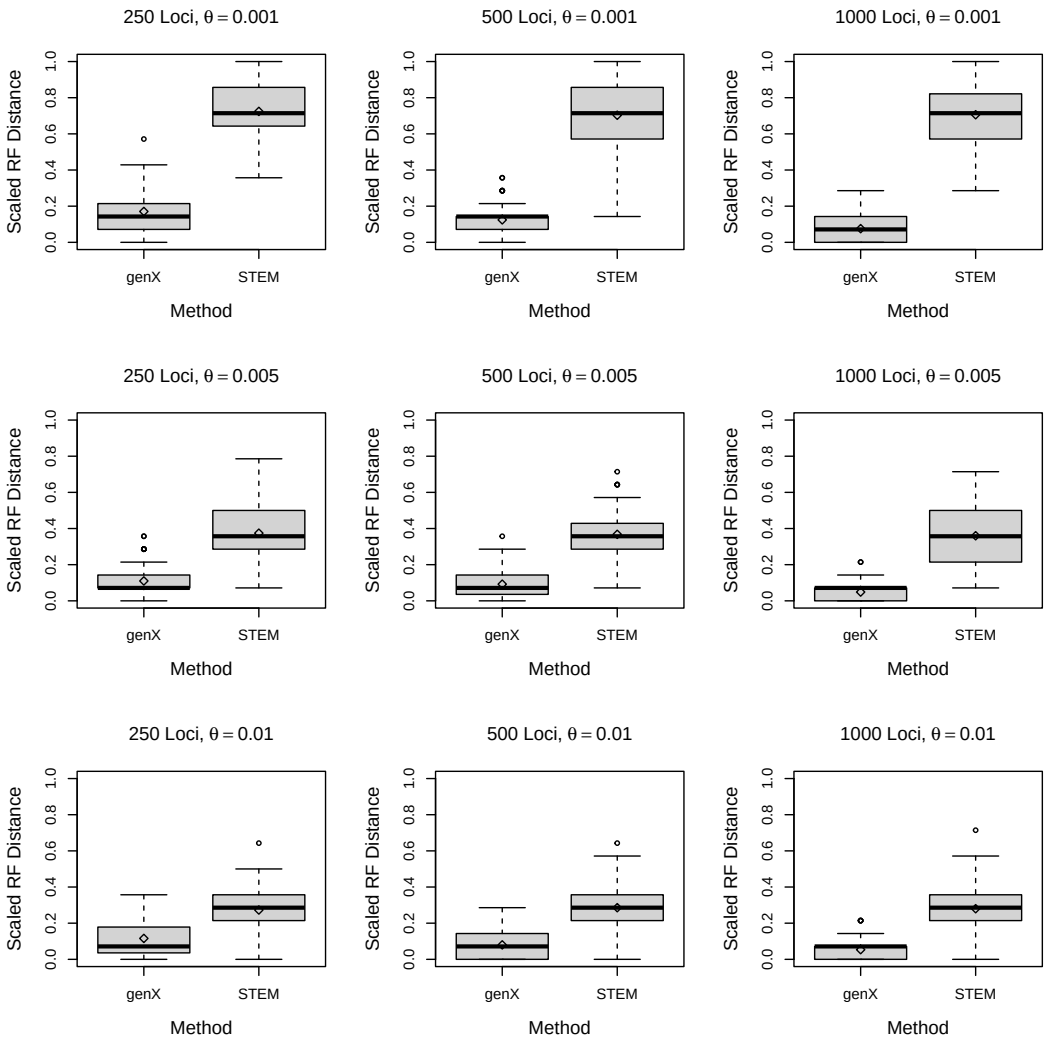

Figure 2.6: Scaled RF distances for species trees inferred by STEM or genX, assuming rooted trees, from true species trees for 16-taxon trees, 500 nt.

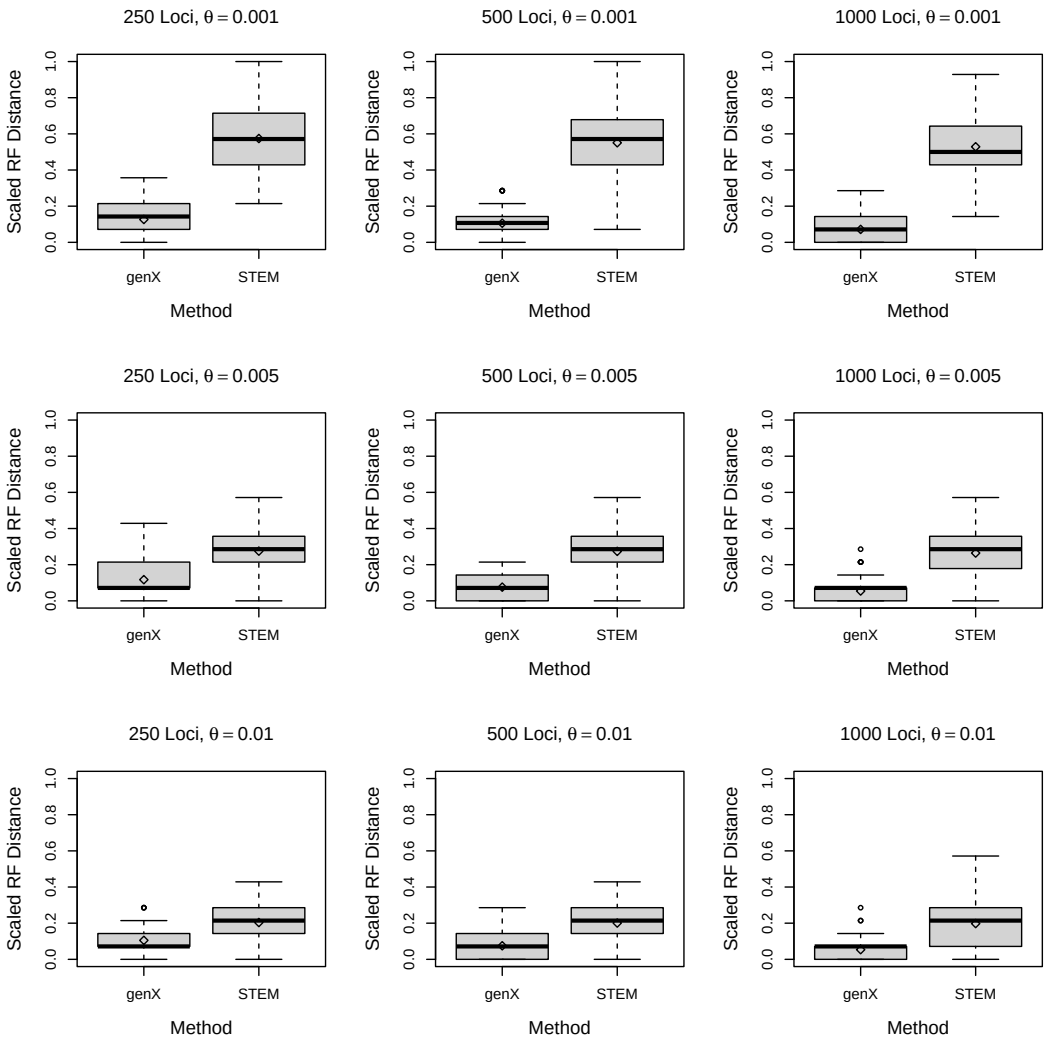

Figure 2.7: Scaled RF distances for species trees inferred by STEM or genX, assuming rooted trees, from true species trees for 16-taxon trees, 1000 nt.

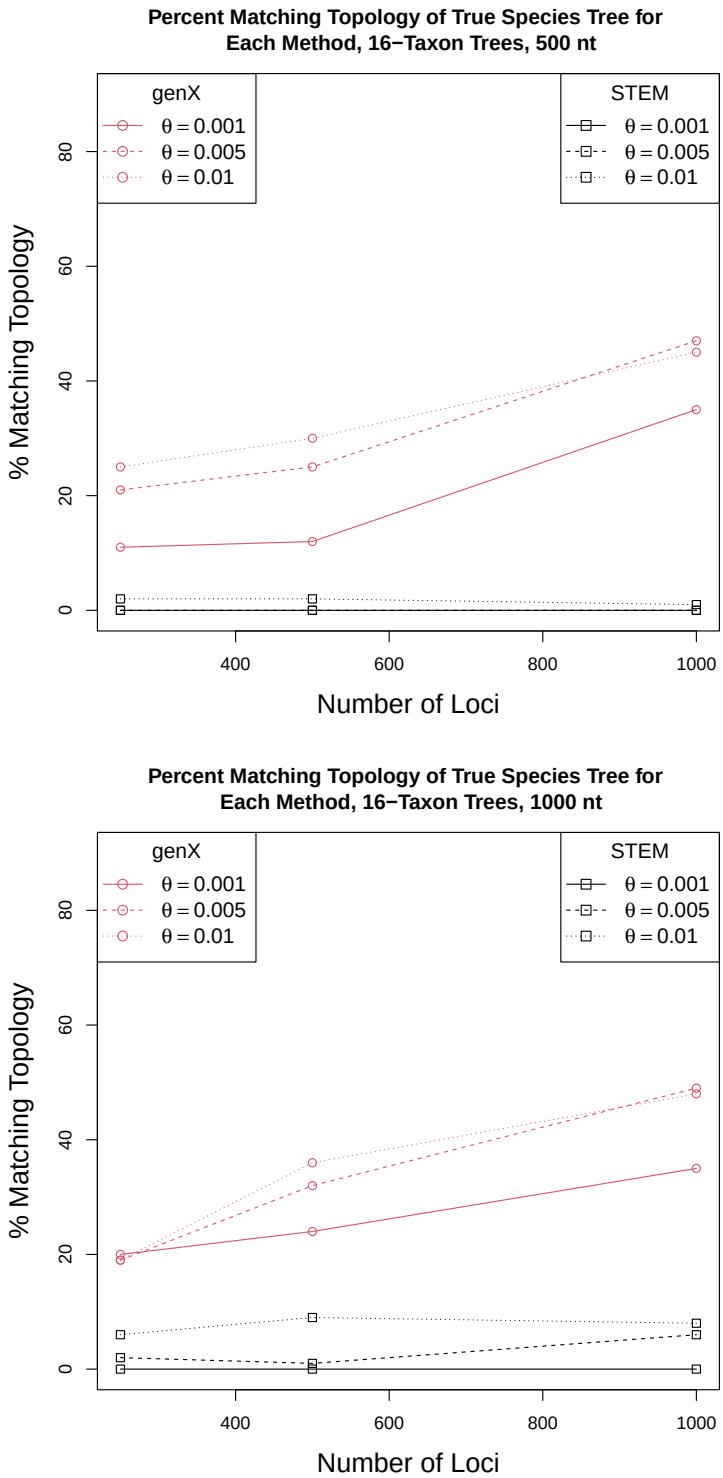

Figure 2.8: Percent of inferred rooted topologies for STEM or genX matching true species trees for 16-taxon trees, 500 nt or 1000 nt.

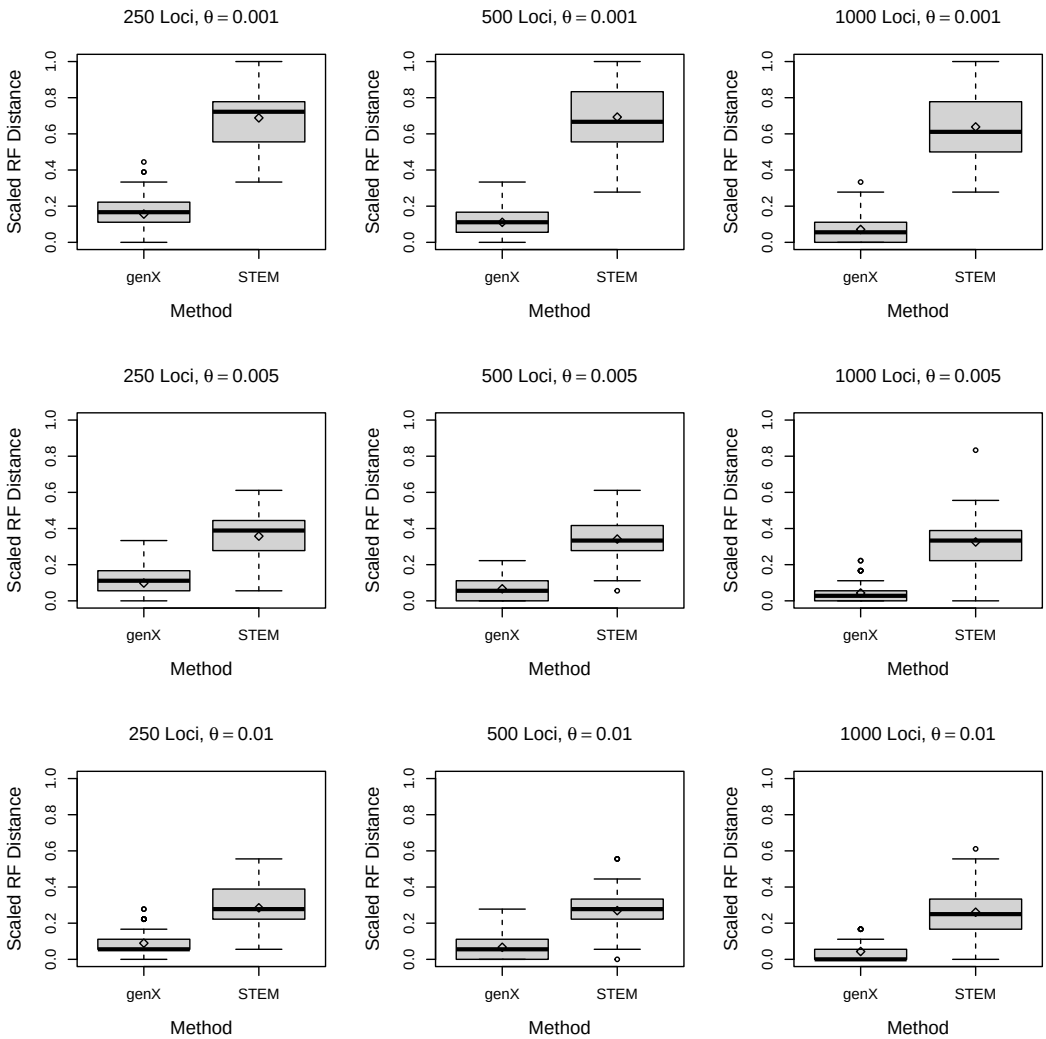

Figure 2.9: Scaled RF distances assuming rooted trees from true species trees for 20-taxon trees, 500 nt.

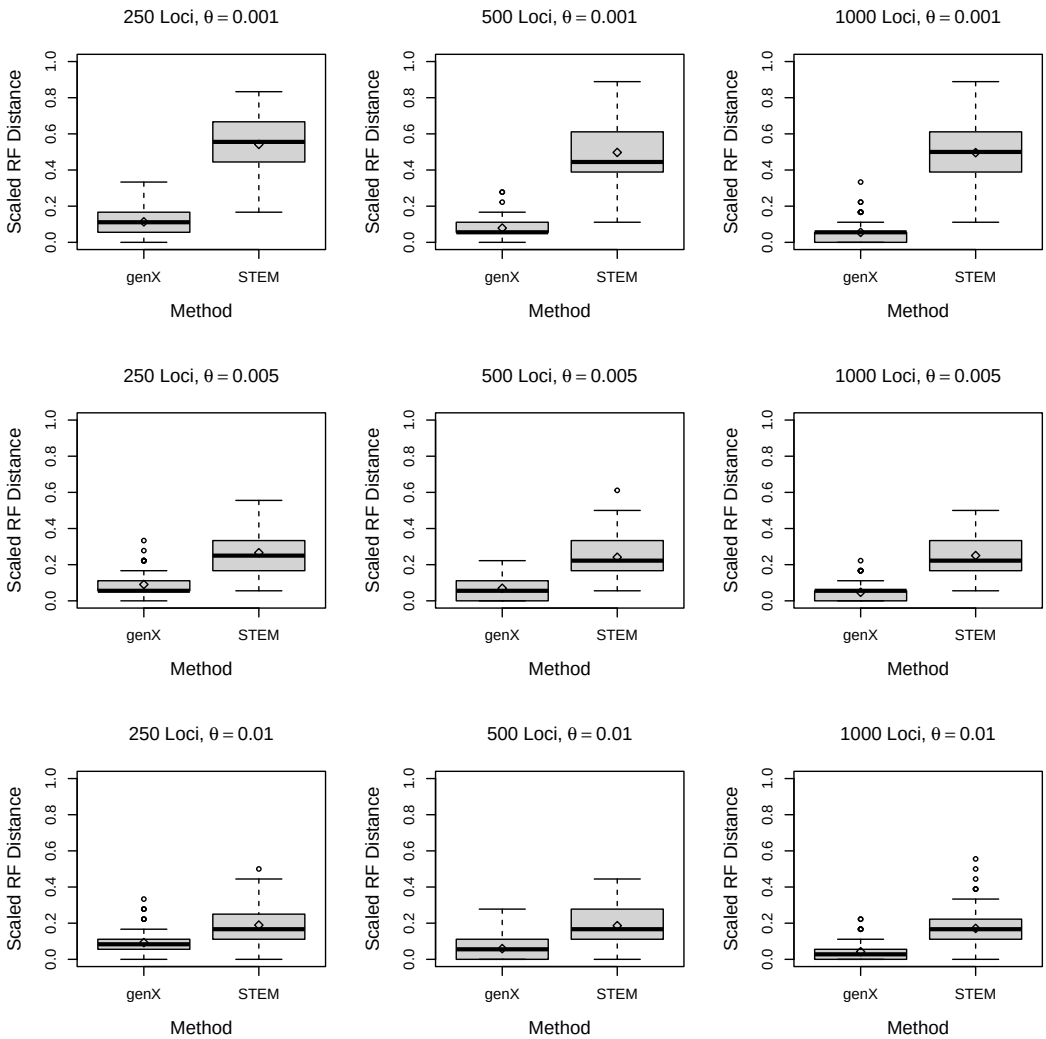

Figure 2.10: Scaled RF distances assuming rooted trees from true species trees for 20-taxon trees, 1000 nt.

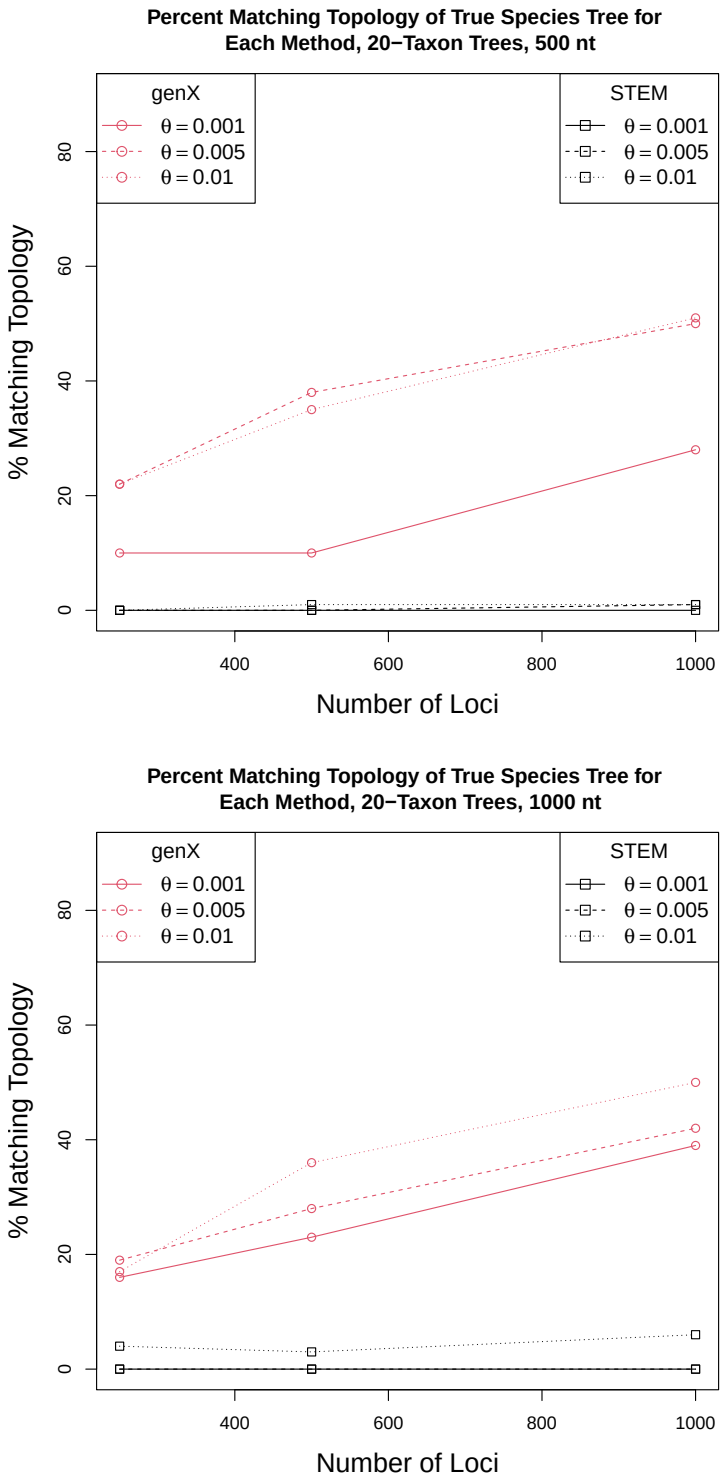

Figure 2.11: Percent of inferred rooted topologies matching true species trees for 20-taxon trees, 500 nt or 1000 nt.

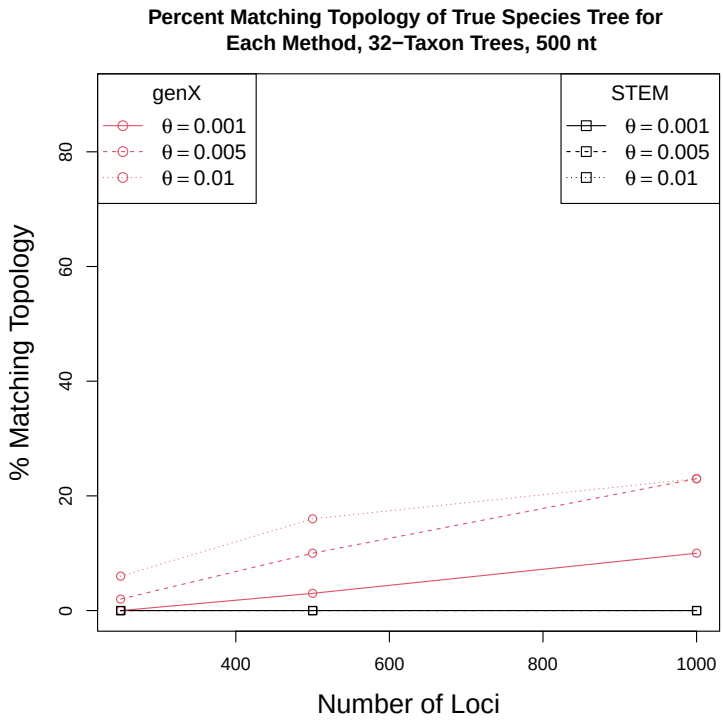

Figure 2.12: Percent of inferred rooted topologies for STEM or genX matching true species trees for 32-taxon trees, 500 nt.

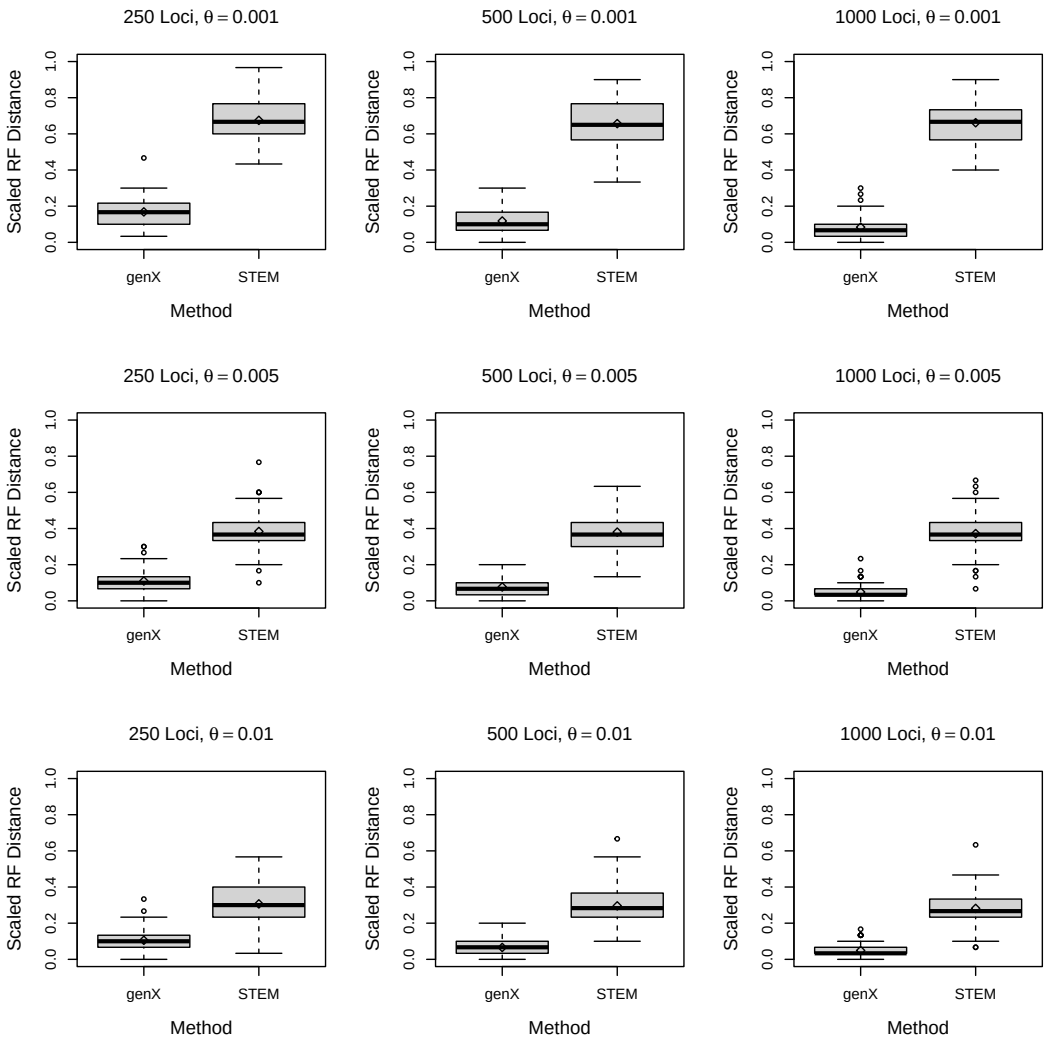

Figure 2.13: Scaled RF distances for species trees inferred by STEM or genX, assuming rooted trees from true species trees for 32-taxon trees, 500 nt.

Eight Taxa - GenX and STEM 2.0

Method	nLoci	θ	seqL	Min. RF	Max. RF	Mean RF	%Correct
GenX	250	0.001	500	0.000	0.500	0.192	19
STEM	250	0.001	500	0.167	1.000	0.875	0
GenX	500	0.001	500	0.000	0.500	0.132	41
STEM	500	0.001	500	0.333	1.000	0.880	0
GenX	1000	0.001	500	0.000	0.333	0.098	42
STEM	1000	0.001	500	0.333	1.000	0.855	0
GenX	250	0.005	500	0.000	0.500	0.123	41
STEM	250	0.005	500	0.000	1.000	0.575	1
GenX	500	0.005	500	0.000	0.333	0.088	51
STEM	500	0.005	500	0.000	1.000	0.560	1
GenX	1000	0.005	500	0.000	0.333	0.055	69
STEM	1000	0.005	500	0.000	1.000	0.572	1
GenX	250	0.010	500	0.000	0.333	0.137	35
STEM	250	0.010	500	0.000	1.000	0.405	6
GenX	500	0.010	500	0.000	0.333	0.093	50
STEM	500	0.010	500	0.000	0.833	0.393	3
GenX	1000	0.010	500	0.000	0.333	0.067	62
STEM	1000	0.010	500	0.000	0.833	0.375	6
GenX	250	0.001	1000	0.000	0.500	0.152	32
STEM	250	0.001	1000	0.167	1.000	0.778	0
GenX	500	0.001	1000	0.000	0.333	0.103	48
STEM	500	0.001	1000	0.333	1.000	0.765	0
GenX	1000	0.001	1000	0.000	0.333	0.090	53
STEM	1000	0.001	1000	0.167	1.000	0.772	0
GenX	250	0.005	1000	0.000	0.333	0.127	39
STEM	250	0.005	1000	0.000	1.000	0.378	9
GenX	500	0.005	1000	0.000	0.333	0.108	44
STEM	500	0.005	1000	0.000	1.000	0.392	3
GenX	1000	0.005	1000	0.000	0.333	0.075	56
STEM	1000	0.005	1000	0.000	1.000	0.375	5
GenX	250	0.010	1000	0.000	0.500	0.122	44
STEM	250	0.010	1000	0.000	0.667	0.268	12
GenX	500	0.010	1000	0.000	0.333	0.098	49
STEM	500	0.010	1000	0.000	0.833	0.222	21
GenX	1000	0.010	1000	0.000	0.333	0.060	65
STEM	1000	0.010	1000	0.000	0.667	0.205	23

Table 2.4: Results for GenX vs STEM, 100 trees with eight species: scaled RF distances from true ST assuming rooted trees; % correct topology.

Eight Taxa - Bayes X and Bayes τ

Method	nLoci	θ	seqL	Min. RF	Max. RF	Mean RF	%Correct
Bayes X	250	0.001	500	0.000	0.500	0.227	16
Bayes τ	250	0.001	500	0.000	0.667	0.213	14
Bayes X	1000	0.001	500	0.000	0.333	0.097	44
Bayes τ	1000	0.001	500	0.000	0.333	0.107	42
Bayes X	250	0.010	500	0.000	0.333	0.123	44
Bayes τ	250	0.010	500	0.000	0.333	0.150	34
Bayes X	1000	0.010	500	0.000	0.167	0.050	70
Bayes τ	1000	0.010	500	0.000	0.333	0.083	60
Bayes X	250	0.001	1000	0.000	0.500	0.153	38
Bayes τ	250	0.001	1000	0.000	0.500	0.167	26
Bayes X	1000	0.001	1000	0.000	0.333	0.100	48
Bayes τ	1000	0.001	1000	0.000	0.333	0.097	46
Bayes X	250	0.010	1000	0.000	0.333	0.087	58
Bayes τ	250	0.010	1000	0.000	0.333	0.120	46
Bayes X	1000	0.010	1000	0.000	0.167	0.053	68
Bayes τ	1000	0.010	1000	0.000	0.333	0.080	56

Table 2.5: Results for Bayes X and Bayes τ, 50 trees with eight species: scaled RF distances from true ST assuming rooted trees; % correct topology.

Sixteen Taxa - GenX and STEM 2.0

Method	nLoci	θ	seqL	Min. RF	Max. RF	Mean RF	%Correct
GenX	250	0.001	500	0.000	0.571	0.171	11
STEM	250	0.001	500	0.357	1.000	0.724	0
GenX	500	0.001	500	0.000	0.357	0.124	12
STEM	500	0.001	500	0.143	1.000	0.703	0
GenX	1000	0.001	500	0.000	0.286	0.076	35
STEM	1000	0.001	500	0.286	1.000	0.706	0
GenX	250	0.005	500	0.000	0.357	0.110	21
STEM	250	0.005	500	0.071	0.786	0.374	0
GenX	500	0.005	500	0.000	0.357	0.092	25
STEM	500	0.005	500	0.071	0.714	0.368	0
GenX	1000	0.005	500	0.000	0.214	0.049	47
STEM	1000	0.005	500	0.071	0.714	0.360	0
GenX	250	0.010	500	0.000	0.357	0.116	25
STEM	250	0.010	500	0.000	0.643	0.273	2
GenX	500	0.010	500	0.000	0.286	0.080	30
STEM	500	0.010	500	0.000	0.643	0.286	2
GenX	1000	0.010	500	0.000	0.214	0.054	45
STEM	1000	0.010	500	0.000	0.714	0.280	1
GenX	250	0.001	1000	0.000	0.357	0.126	20
STEM	250	0.001	1000	0.214	1.000	0.574	0
GenX	500	0.001	1000	0.000	0.286	0.106	24
STEM	500	0.001	1000	0.071	1.000	0.551	0
GenX	1000	0.001	1000	0.000	0.286	0.071	35
STEM	1000	0.001	1000	0.143	0.929	0.529	0
GenX	250	0.005	1000	0.000	0.429	0.117	19
STEM	250	0.005	1000	0.000	0.571	0.275	2
GenX	500	0.005	1000	0.000	0.214	0.076	32
STEM	500	0.005	1000	0.000	0.571	0.274	1
GenX	1000	0.005	1000	0.000	0.286	0.054	49
STEM	1000	0.005	1000	0.000	0.571	0.264	6
GenX	250	0.010	1000	0.000	0.286	0.105	19
STEM	250	0.010	1000	0.000	0.429	0.204	6
GenX	500	0.010	1000	0.000	0.286	0.075	36
STEM	500	0.010	1000	0.000	0.429	0.201	9
GenX	1000	0.010	1000	0.000	0.286	0.054	48
STEM	1000	0.010	1000	0.000	0.571	0.198	8

Table 2.6: Results for GenX vs STEM, 100 trees with 16 species: scaled RF distances from true ST assuming rooted trees; % correct topology.

Twenty Taxa - GenX and STEM 2.0

Method	nLoci	θ	seqL	Min. RF	Max. RF	Mean RF	%Correct
GenX	250	0.001	500	0.000	0.444	0.156	10
STEM	250	0.001	500	0.333	1.000	0.688	0
GenX	500	0.001	500	0.000	0.333	0.111	10
STEM	500	0.001	500	0.278	1.000	0.693	0
GenX	1000	0.001	500	0.000	0.333	0.071	28
STEM	1000	0.001	500	0.278	1.000	0.638	0
GenX	250	0.005	500	0.000	0.333	0.098	22
STEM	250	0.005	500	0.056	0.611	0.358	0
GenX	500	0.005	500	0.000	0.222	0.067	38
STEM	500	0.005	500	0.056	0.611	0.342	0
GenX	1000	0.005	500	0.000	0.222	0.043	50
STEM	1000	0.005	500	0.000	0.833	0.326	1
GenX	250	0.010	500	0.000	0.278	0.089	22
STEM	250	0.010	500	0.056	0.556	0.285	0
GenX	500	0.010	500	0.000	0.278	0.067	35
STEM	500	0.010	500	0.000	0.556	0.269	1
GenX	1000	0.010	500	0.000	0.167	0.043	51
STEM	1000	0.010	500	0.000	0.611	0.261	1
GenX	250	0.001	1000	0.000	0.333	0.113	16
STEM	250	0.001	1000	0.167	0.833	0.542	0
GenX	500	0.001	1000	0.000	0.278	0.079	23
STEM	500	0.001	1000	0.111	0.889	0.497	0
GenX	1000	0.001	1000	0.000	0.333	0.055	39
STEM	1000	0.001	1000	0.111	0.889	0.496	0
GenX	250	0.005	1000	0.000	0.333	0.091	19
STEM	250	0.005	1000	0.056	0.556	0.266	0
GenX	500	0.005	1000	0.000	0.222	0.069	28
STEM	500	0.005	1000	0.056	0.611	0.241	0
GenX	1000	0.005	1000	0.000	0.222	0.047	42
STEM	1000	0.005	1000	0.056	0.500	0.251	0
GenX	250	0.010	1000	0.000	0.333	0.093	17
STEM	250	0.010	1000	0.000	0.500	0.189	4
GenX	500	0.010	1000	0.000	0.278	0.059	36
STEM	500	0.010	1000	0.000	0.444	0.186	3
GenX	1000	0.010	1000	0.000	0.222	0.042	50
STEM	1000	0.010	1000	0.000	0.556	0.171	6

Table 2.7: Results for GenX vs STEM, 100 trees with 20 species: scaled RF distances from true ST assuming rooted trees; % correct topology.

Thirty-Two Taxa - GenX and STEM 2.0

Method	nLoci	θ	seqL	Min. RF	Max. RF	Mean RF	%Correct
GenX	250	0.001	500	0.033	0.467	0.168	0
STEM	250	0.001	500	0.433	0.967	0.675	0
GenX	500	0.001	500	0.000	0.300	0.118	3
STEM	500	0.001	500	0.333	0.900	0.657	0
GenX	1000	0.001	500	0.000	0.300	0.084	10
STEM	1000	0.001	500	0.400	0.900	0.662	0
GenX	250	0.005	500	0.000	0.300	0.110	2
STEM	250	0.005	500	0.100	0.767	0.384	0
GenX	500	0.005	500	0.000	0.200	0.077	10
STEM	500	0.005	500	0.133	0.633	0.380	0
GenX	1000	0.005	500	0.000	0.233	0.048	23
STEM	1000	0.005	500	0.067	0.667	0.372	0
GenX	250	0.010	500	0.000	0.333	0.105	6
STEM	250	0.010	500	0.033	0.567	0.307	0
GenX	500	0.010	500	0.000	0.200	0.066	16
STEM	500	0.010	500	0.100	0.667	0.296	0
GenX	1000	0.010	500	0.000	0.167	0.049	23
STEM	1000	0.010	500	0.067	0.633	0.282	0

Table 2.8: Results for GenX vs STEM, 100 trees with 32 species: scaled RF distances from true ST assuming rooted trees; % correct topology.

2.3.3 Application to Empirical Datasets

Application to Empirical Dataset with 10706 5-Taxon Gene Trees

A set of 10706 gene trees with five taxa *Hylobates moloch* (HMO), *Hylobates pileatus* (HPL), *Nomascus leucogenys* (NLE), *Hoolock leuconedys* (HLE) and *Symphalangus syndactylus* (SSY) that passed the molecular clock was used to infer a species tree using STEM 2.0, genX, ASTRAL (Zhang et al., 2018) and the Bayesian methods described earlier. This set that passed the molecular clock included 10706 out of 12413 total trees (Kim and Degnan, 2020). The species HMO and HPL represent the genus *Hylobates* (H), while NLE, HLE, and SSY represent the genera *Nomascus* (N), *Hoolock* (B) and *Symphalangus* (S), respectively. The gene trees were estimated from the gibbon non-coding sequence data from Shi and Yang (2018) and Carbone et al. (2014) as described in Kim and Degnan (2020) Section 3.

The parameter θ was estimated for the full data and a subset of each size shown below using the `pegas` package version 1.0-1 in R, and all estimates were approximately 0.009. We compared the trees inferred by ASTRAL (Zhang et al., 2018), STEM, and genX for the full data and the subsets of it as well as trees inferred by Shi and Yang (2018) (they infer the phylogeny of these four genera to be (H, (N, (B, S))), called Tree 1 in their paper). Shi and Yang (2018) include in their Table 1 a list of 15 species trees ordered according to the frequency which with they were inferred in earlier work by Carbone et al. (2014). Hereafter these are referred to as "Tree 1", "Tree 2", etc., with the topology given.

Using all 10706 of the gene trees, and the estimated value of 0.009 for θ, genX inferred the topology (((S, B), H), N), which corresponds to Tree 2 in Shi and Yang (2018), while ASTRAL inferred Tree 1, (((S, B), N), H). Additionally, we randomly chose subsets from these 10706 gene trees, of size 100, 500, 1000, 2000 and 8000 gene trees, and replicated these subsets 100 times each. GenX inferred Tree 1 and

Tree 2 each approximately 50% of the time for the smaller datasets, down to size 500. For size 100, genX inferred more of trees other than 1 or 2. ASTRAL most commonly inferred these two trees as well, but with increasing proportion of Tree 1 as the sample size increased (Figure 2.14).

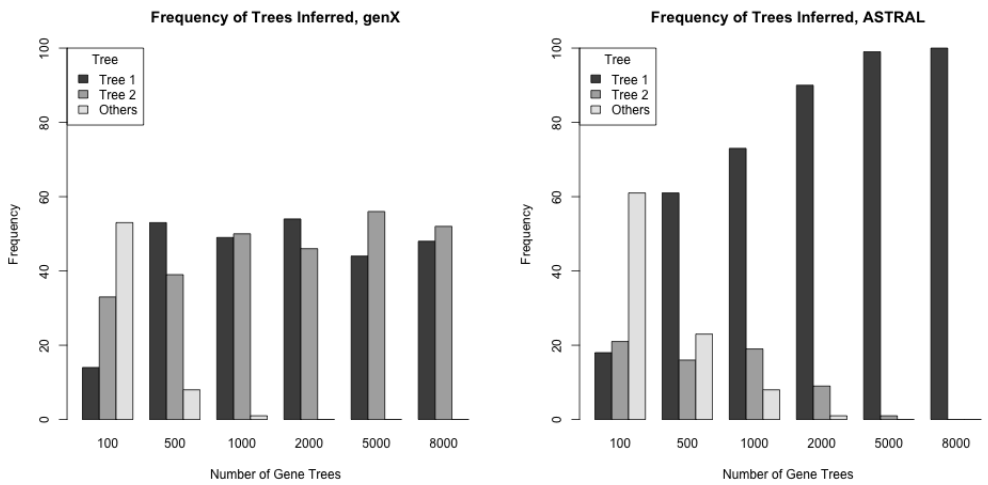

Figure 2.14: Frequency of species trees inferred by genX and ASTRAL from 5-taxon gibbon data.

STEM 2.0 inferred the tree (((H, B), N), S) - Tree 8 in the Shi and Yang paper - for the full dataset, with Tree 8 or Tree 6 (((H, B), S), N) the most commonly inferred tree for smaller datasets, except for datasets of size 100. For these, STEM most commonly inferred Trees 1 and 3, which was (((N, B), S), H). These were inferred 12 times each out of 100 while Trees 8 and 6 were inferred six and nine times respectively. These results are summarized in Figure 2.15.

Additionally, a consensus tree was obtained for both ASTRAL and genX via the bootstrap. We resampled the 10706 gene trees with replacement to obtain the same size dataset (10706 gene trees) 100 times, inferred a species tree with each method from each of the 100 bootstrap datasets, and then obtained a consensus tree with the `consense` program in PHYLIP (Felsenstein, 2009). For both methods, the

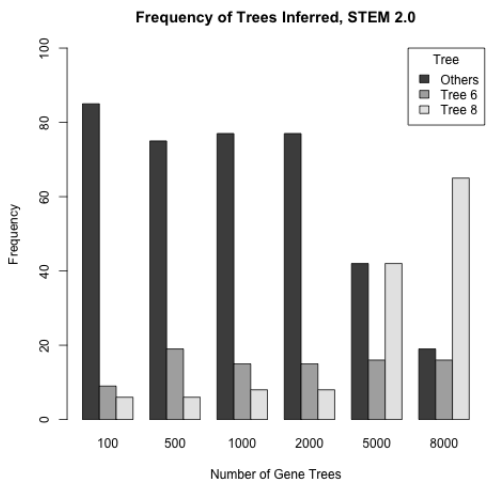

Figure 2.15: Frequency of species trees inferred by STEM 2.0 from 5-taxon gibbon data.

clades (HLE, SSY) and (HMO, HPL) were obtained for all 100 bootstrap datasets. For ASTRAL, the clade (NLE, (HLE, SSY)) was obtained 99 times and the clade ((HLE, SSY), (HMO, HPL)) obtained once. For genX, the clade (NLE, (HLE, SSY)) was obtained 39 times, and the clade ((HLE, SSY), (HMO, HPL)) obtained 61 times. Tree 1 was the consensus tree for ASTRAL, with strong bootstrap support for the involved clades, and Tree 2 was the consensus tree for genX, but without strong bootstrap support for the involved clades.

The Bayesian methods used on this dataset showed convergence issues for the parameter τ more severely than in the simulated data. Neither X nor β showed convergence issues. Convergence was again assessed by examining trace plots and considering $\widehat{R}$. Unlike the results from the simulated data, many models for the empirical dataset had $\widehat{R}$ well above 1.1, with many of the trace plots for the non-converging models showing obvious separation of the two chains. We attempted to improve the convergence of the models by increasing the number of iterations and trying both of the priors for τ discussed previously. The results of these attempts are

shown in Table 2.9. Note that to infer a tree from these methods, a distance matrix is constructed from the minimums of the posterior iterates of X or the means of those of τ, so a non-converging model would create a missing entry in the distance matrix, as no valid inference can be made when a Bayesian model does not converge. However, with only some of the entries missing (non-converging model for a particular pair of species), a set of possible trees could be inferred, with the set narrowed down when more of the models have converged. Table 2.9 includes the trees obtained by clustering the minimums of the posterior distribution of X or the means for τ, but these are not reliable estimates with the majority of the models not converging. Some improvement in convergence was noted with these attempts, in that more of the models converged when more iterations were performed (recall that a separate model is fit for each pairwise distance for this Bayesian method). Thus, it is possible that increasing the number of iterations even further would eventually obtain convergence for all ten models.

Prior for τ	Iterations	NC Models	NC $\widehat{R}$ Range	Bayes X	Bayes τ
$N(9, 14^2)$	40000	7/10	1.29–7.36	Tree 1	Tree 9
$N(2, 1)$	40000	7/10	1.11–3.94	Tree 1	Tree 14
$N(9, 14^2)$	100000	6/10	1.14–2.63	Tree 1	Tree 11

Table 2.9: Results of Bayesian Methods on 5-Taxon Gibbon Data.
NC indicates non-converging. Since many of the models still did not converge even with the larger number of iterations attempted, the trees listed are stated for methodological purposes and do not necessarily reflect valid inference from the posterior distributions.

Application to Empirical Dataset with 10631 8-Taxon Gene Trees

A similar analysis was performed with an additional subset of the gibbons data using 10631 gene trees that passed the molecular clock. This dataset contained eight taxa plus an outgroup. The outgroup was removed after inference using ASTRAL and

before inference using the genX methods and STEM to aid in comparison of the resulting trees. The eight taxa consisted of one individual from each species HMO and HPL (genera H), and two from each species SSY (genera S), HLE (genera B) and NLE (genera N). With this dataset, an estimate for θ of 0.006 was obtained using the `pegas` package in R.

In this case, genX inferred Tree 1 with the full data, while ASTRAL inferred Tree 2. For the larger data subsets, genX most commonly inferred Tree 1, with Tree 2 the second most common, with increasing proportion of Tree 1 as the sample size increased. For the smallest subset, size 100, Tree 2 was inferred 33 times and Tree 1 was inferred 24 times using genX. For ASTRAL, Tree 2 was the most commonly inferred in all subsets, and Tree 1 the second most common in the subsets down to size 1000, with increasing proportion of Tree 2 as the sample size increased. For size 500 and 100, Tree 9 was the second most commonly inferred by ASTRAL rather than Tree 2. These results are summarized in Figure 2.16.

STEM 2.0 inferred Tree 10 (((N, S), H), B) for the full dataset. For the data subsets of size 8000, 5000 and 500, Tree 10 was the most common for STEM, with Tree 7 ((H, N), (S, B)) or Tree 4 (((N, S), B), H) also commonly inferred. For the subsets of size 1000 and 2000, Trees 1 and 2 were the most commonly inferred for STEM, while Trees 2 and 3 were the most commonly inferred for the subsets of size 100. These results are summarized in Figure 2.17. Note that STEM has the option to specify when multiple individuals per species are represented in the data. That option was not used in this analysis, but STEM still correctly grouped such individuals together in all trees inferred from this dataset.

As with the first subset of the gibbon data, a consensus tree was obtained for both ASTRAL and genX via the bootstrap using the 10631 gene trees. We resampled the 10631 gene trees with replacement to obtain the same size dataset (10631 gene trees) 100 times, inferred a species tree with each method from each of the 100 bootstrap

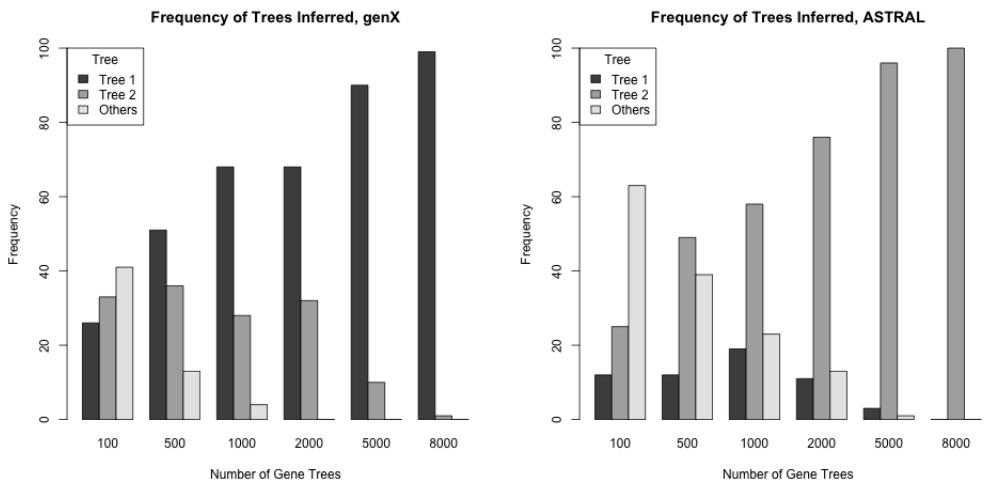

Figure 2.16: Frequency of species trees inferred by genX and ASTRAL from 8-taxon gibbon data.

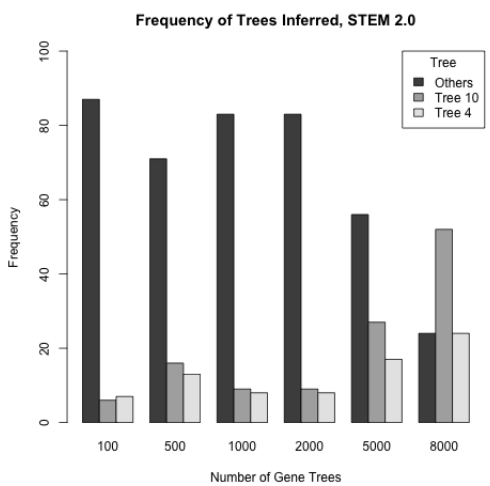

Figure 2.17: Frequency of species trees inferred by STEM 2.0 from 8-taxon gibbon data

datasets, and then obtained a consensus tree with the `consense` program in PHYLIP (Felsenstein, 2009).

For both methods, the H, B, S and N clades were obtained for all 100 bootstrap

datasets, i.e. the two species or two individuals from each of these genera were always grouped together. For genX, the (B, S) clade was inferred 100 times, the ((B, S), N) clade was inferred 91 times, and the ((B, S), H) clade 9 times. For ASTRAL, the clade (B, S) was obtained 99 times, and the clade containing B, S and H 99 times. The ((B, S), N) clade was inferred once, and the clade (H, S) was inferred once. With this dataset, Tree 1 was the consensus tree for genX, and Tree 2 was the consensus tree for ASTRAL.

As with the 5-taxon empirical dataset, the Bayesian methods used on this dataset showed convergence issues for the parameter τ more severely than in the simulated data. Neither X nor β showed convergence issues. Again, many models for this empirical dataset had $\widehat{R}$ well above 1.1, with several trace plots for the non-converging models showing obvious separation of the two chains. The results of attempts to improve convergence are shown in Table 2.10. The table includes the trees obtained by clustering the minimums of the posterior distribution of X or the means for τ, but as with the 5-taxon empirical dataset, these are not reliable estimates with the majority of the models not converging. Some improvement in convergence was noted with these attempts, similarly to the previously mentioned dataset. Thus, it is possible that increasing the number of iterations even further would eventually obtain convergence for all 28 models.

Prior for τ	Iterations	NC Models	NC $\widehat{R}$ Range	Bayes X	Bayes τ
$N(9, 14^2)$	40000	23/28	1.11–9.11	Tree 1	Tree 2
$N(2, 1)$	40000	18/28	1.12–4.06	Tree 1	Tree 12
$N(9, 14^2)$	100000	21/28	1.11–4.91	Tree 1	Tree 2

Table 2.10: Results of Bayesian Methods on 8-Taxon Gibbon Data.
NC indicates non-converging. Since many of the models still did not converge even with the larger number of iterations attempted, the trees listed are stated for methodological purposes and do not necessarily reflect valid inference from the posterior distributions.

2.4 Discussion

The genX method and the Bayesian methods outlined here show improvement over the GLASS/MT as implemented in STEM 2.0 with accuracy of species tree inference, in terms of RF distance and frequency of matching the topology of the true species tree, as shown in Figures 2.3 – 2.13. As expected, the percent of inferred trees with topology matching the true species trees goes down with the number of taxa for STEM and genX, but the mean scaled RF distance remains similar or even smaller with larger trees for some settings. The Bayesian methods are more accurate than genX in some cases, e.g. as shown in Figure 2.5. Additionally, in the simulated data with eight taxa, Bayes X appeared to be more accurate than genX for 250 loci with the higher value of either θ or sequence length. In general, Bayes τ performed comparably to or worse than genX, but still showed improvement over STEM 2.0. The Bayesian methods do require substantially more computation time and can be prone to convergence issues as demonstrated here with the gibbon datasets.

GenX is statistically consistent when the additive errors (difference between pairwise distances in the estimated and true gene trees) have mean 0 or multiplicative errors (ratio of pairwise distances from estimated and true gene trees) have a mean of 1. This assumption only appeared to be approximately true for the largest value of θ (0.01) in our simulations, and in practice the additive or multiplicative errors would be unknown and therefore the assumption not verifiable. However, the method still showed improved accuracy over STEM even when this assumption was violated. As shown in the figures depicting RF distances, such as Figures 2.3 and 2.4, the mean scaled RF distance appears to decrease with increasing loci for genX, but to stay fairly constant for STEM, particularly with larger trees.

One reason for the improved performance of these methods over STEM may be that the simulated distributions for X — the pairwise distances calculated from true

gene trees — are less likely to contain zeros than a set of W, the pairwise distances calculated from maximum likelihood gene trees estimated from DNA sequence data. In the Bayesian approach, the posterior distribution of X does not contain zeros. The issue of the estimated pairwise distances containing zeros was explored in detail in DeGiorgio and Degnan (2014). In summary, this issue can either result in the true minimum distance being ignored in the case of STEM 2.0, as it takes the minimum non-zero pairwise distance as the entry for the distance matrix, or unresolved trees in the case of STEM 1.1, which adds 10^{-6} to any observed distance of zero.

Some limitations of the methods developed here include that they return only an estimated tree, where STEM performs several other functions, such as computing the likelihood and returning a set of the highest likelihood trees. Since the new methods here include a random component, they will not always return the same tree (but usually return the same topology) if run multiple times. The genX method is relatively fast but becomes a bit slower than STEM with increasing loci and number of species. Also, it was noted to be less accurate than ASTRAL for the 5-taxon gibbons dataset, and did not converge to one tree in that case (as shown in Figure 2.14). With the full 5-taxon data, its consensus tree was incorrect, but without strong bootstrap support for the clades in that consensus tree. For the 8-taxon gibbon data, genX did perform better, which may indicate sensitivity to taxon sampling as described, for example, in Nabhan and Sarkar (2012). While the methods here showed improvement over STEM, they still are less accurate for lower values of θ and shorter sequence lengths. Thus, the new methods may still be subject to some GTEE despite the improvement noted. This issue may be expected, since as discussed above and in section 2.1, the assumption about the mean of the errors in the measurement error model is not always correct.

Areas for future research could include testing the method on a wider variety of trees; e.g. the simulated species trees were only simulated with the speciation

rate $\lambda = 1$, and the simulated sequences only under the one substitution model discussed above. Thus, it is unknown how the methods here would perform under different settings, other than as illustrated with the gibbon datasets. It may also be worthwhile to try additional priors for the τ parameter in the Bayesian models to try to improve the convergence issue, to try clustering the median of the posterior iterates of τ rather than the mean, and to try other hierarchical clustering approaches with the posterior estimate of τ. Additionally, it is unknown whether a measurement error modeling approach could also improve other distance matrix based species tree inference methods.

In summary, despite the above limitations, the genX method and the Bayesian methods developed here for modeling the measurement error between estimated and true gene trees improve the performance of the GLASS/MT/STEM tree when the input gene trees are estimated from DNA sequences. Since the genX method is relatively fast, it could be used to obtain starting trees for other methods that may be slower but more accurate. The Bayesian method developed here using the minimum values of X appears more accurate in many cases, but was tested in an even more limited number of scenarios due to longer computation times.

Chapter 3

PB Approach to Multi-factor heteANOVA Models

3.1 Introduction

Consider the multi-factor ANOVA problem of $abcd\ldots$ normal populations with unequal population variances $\sigma^2_{ijkm\ldots}$, $i = 1, ..., a$, $j = 1, ..., b$, $k = 1, ..., c$, $m = 1, ..., d$, ... and let $Y_{ijkm\ldots 1}, Y_{ijkm\ldots 2}, ..., Y_{ijkm\ldots n_{ijkm\ldots}}$ be the observations from each group. The full ANOVA model, hereafter called heteANOVA model, is

$$Y_{ijkm\ldots} = G + A_i + B_j + C_k + D_m + ... + AB_{ij} + AC_{ik} + AD_{im} + BC_{jk} + BD_{jm} + CD_{km} + ... + ABC_{ijk} + ABD_{ijm} + ACD_{ikm} + BCD_{jkm} + ... + ABCD_{ijkm} + ... + e_{ijkm\ldots}$$

where $e_{ijkm\ldots} \sim N(0, \sigma^2_{ijkm\ldots})$. The usual F-tests for main and interaction effects in these models assume equal group variances, and can be smaller or larger in size than the nominal level when this assumption is violated (Weerahandi, 1995; Bao and Ananda, 2001; Scheffe, 1959). Weerahandi (1995) showed examples of this: when there was no particular relationship between sample size and group variance, the p-

value for the conventional F-test was too large (Type II error); however, they provide an additional example where sample sizes were negatively correlated with the group variances, and the p-value of the conventional F-test was shown to be too small in this case. Transformed data, such as the log or square root of observed values, may in some cases meet the equal variance assumption. However, this method does not always work and can make the results more difficult to interpret. Other approaches such as the generalized F-test have been proposed (Weerahandi, 1995; Ananda and Weerahandi, 1997) for one-way and two-way models, but may not perform well with larger numbers of treatment levels (Xu et al., 2013).

As described in Christensen (2016) section 4.3, caution is needed when making practical decisions based on differences in means between groups with unequal variances. For example, if a lower value of a response is desired, such as blood pressure, a treatment group with a smaller mean and smaller variance may have a smaller probability of achieving the desired outcome than a treatment group with a larger mean and also larger variance. Thus, additional consideration of implications for the practical issue being studied is warranted. Nevertheless, the problem of unequal variance does arise in practice, so methods of dealing with the problem are desirable.

The parametric bootstrap (PB) approach has been shown to work well for one-way and two-way heteANOVA models, including cases with unbalanced data (Xu et al., 2013; Krishnamoorthy and Lu, 2007; Zhang, 2015a,b). Rather than resampling from the data with replacement, as in the perhaps more traditional bootstrap procedures, parametric bootstrap involves simulating data from distributions with estimated parameters (Efron and Tibshirani, 1993); this procedure will be discussed in detail in upcoming sections. We would expect the test statistics developed here to generalize to higher-way heteANOVA models. In practice, interpretation of more than three factors with interactions can become quite complicated. Therefore, we illustrate this generalization with a three-factor model and use simulations to compare

the performance of the PB method with the usual F-tests.

Another problem in ANOVA models is multiple comparison procedures (MCP's): simultaneous comparisons of multiple factor levels. The PB approach has been shown to work well for MCPs in one-way and two-way heteANOVA cases (Zhang, 2015a,b). We again generalize this to the three-factor case, and use simulations to compare the performance of the PB methods to Tukey's test.

This chapter is organized as follows. In Section 2 we describe the overall PB method and show relationships between PB methods and conventional F-tests, as well as develop an overall procedure for analyzing data under these models, analogous to conventional methods. In Section 3 the procedure is illustrated for a three-way ANOVA model and performance of the PB tests compared through simulations with that of the usual F-tests for each term in the model. Section 4 describes MCP using PB, and again, simulation results are presented to compare the PB method for MCP with Tukey's test. Section 5 gives an illustration of the PB method with a real example and compares results to those from traditional tests. Section 6 includes discussion of our results, limitations and areas for further research.

3.2 General PB Method for ANOVA Models

The overall process for analyzing multi-factor data using PB methods is similar to the usual ANOVA approach, such as in Christensen (2016) and Kutner et al. (2005), and is shown in Figure 3.1. For the PB method, a PB test rather than an F-test is used at each step of testing to determine the terms to be included in the final model, and PB tests rather than traditional MCP's are used to examine factor level means. In usual ANOVA models where the equal variance assumption is met, for testing $H_0 : Par = 0$, where the parameter of interest (Par) is a main effects term or an interaction term, the usual F-test statistic, or general linear test (Kutner et al.,

2005), takes the form

$$\frac{(SSE(R) - SSE(F))/(dfE(R) - dfE(F))}{MSE(F)},$$

where $SSE(R)$ indicates the sum of square for error (SSE) from the reduced model, $SSE(F)$ indicates SSE from the full model, dfE indicates the degrees of freedom for error for the respective models, and $MSE(F)$ indicates the mean squared error (MSE) from the full model.

In the following sections, we develop PB algorithms for use at each level of testing that are analogous to the F-test (general linear test for a three-way ANOVA model). Algorithm 1 will be used for testing the three-way interaction term, Algorithm 2 for the two-way interaction terms, and Algorithm 3 for testing main effects when no interaction terms have significant effects. Algorithm 4 is used when only one two-way interaction term is significant and we want to test the remaining main effect term that is not involved in the significant interaction. Algorithms 5 and 6 will pertain to MCP's. Algorithms 1-4 are the same at each step other than the design matrix specific to the reduced model being tested. For each of algorithms 1-4, the test statistic is based on the standardized sum of squares for the term under investigation, that is, a function of the numerator of the F-test shown above. As discussed in Christensen (2018), the test statistic S_I can be written in matrix form as:

$$S_I = Y'(A - A_0)'\Sigma_*^{-1}(A - A_0)Y = SSE(R) - SSE(F),$$

where $\Sigma_* = \text{diag}(\sigma^2_{111}, ..., \sigma^2_{111}, \sigma^2_{112}, ..., \sigma^2_{abc})$, (i.e. each σ^2_{ijk} is repeated n_{ijk} times along the diagonal) and Y is the response vector. Under the equal variance assumption, Σ_* reduces to $\sigma^2 I$ where I is an $n \times n$ identity matrix. In S_I above,
$A = X(X'\Sigma_*^{-1}X)^- X'\Sigma_*^{-1}$ and $A_0 = X_0(X_0'\Sigma_*^{-1}X_0)^- X_0'\Sigma_*^{-1}$ are the projection operators onto the column spaces of the design matrices X and X_0 for the full and reduced

models respectively, where X' indicates the transpose of a matrix X. If variances are known, $S_I \sim \chi^2(r(X)-r(X_0))$, as shown in Christensen (2018). We discuss this idea more specifically to each parameter of interest in the following sections. In general, variances are unknown, so the true distribution of this test statistic is also unknown. When group variances are equal (but unknown), the usual F-test statistic follows an F distribution since it is equivalent to:

$$\frac{\frac{Y'(A-A_0)Y}{\sigma^2}/[r(X)-r(X_0)]}{\frac{[N-r(X)]MSE(F)}{\sigma^2}/[N-r(X)]} = \frac{\chi^2_{r(X)-r(X_0)}/[r(X)-r(X_0)]}{\chi^2_{N-r(X)}/[N-r(X)]} \sim F_{[r(X)-r(X_0),N-r(X)]},$$

where $r(X)$ refers to the rank of the X matrix and N is the total number of observations for all groups. When variances are equal so that $\Sigma_* = \sigma^2 I$, A and A_0 reduce to $X(X'X)^-X'$ and $X_0(X_0'X_0)^-X_0'$, respectively, so in this case, $Y'(A-A_0)'\Sigma_*^{-1}(A-A_0)Y = \frac{1}{\sigma^2}Y'(A-A_0)'(A-A_0)Y = \frac{Y'(A-A_0)Y}{\sigma^2}$, since $(A-A_0)$, reduced as above with $\Sigma_* = \sigma^2 I$, is a perpendicular projection operator as shown in Christensen (2018) Theorem B.47, and is thus idempotent and symmetric. In the above F-statistic equation, the σ^2 cancel since they are equal, so unknown σ^2 is not a problem. However, as shown in our simulation results, the pooled variance estimate used for the MSE will not be accurate for all groups and can lead to test statistics being too large or too small, and thus decisions to reject or not reject hypotheses can be too liberal or too conservative, similar to the results of Weerahandi (1995).

The X and X_0 matrices above are the design matrices corresponding to a Y vector with all responses. We will be working with design matrices corresponding to the vector of group means, e.g for a three-way ANOVA model with a=3, b=2 and c=2, $\bar{Y} = (\bar{y}_{111}, \bar{y}_{112}, \bar{y}_{121}, ..., \bar{y}_{322})$, where $\bar{y}_{ijk} = \sum_{m=1}^{n_{ijk}} y_{ijk}/n_{ijk}$. It can be shown that $S_I = Y'(A-A_0)'\Sigma_*^{-1}(A-A_0)Y = SSE(R) - SSE(F) = \bar{Y}'\Sigma^{-1}\bar{Y} - \bar{Y}'\Sigma^{-1}X_*(X_*'\Sigma^{-1}X_*)^-X_*'\Sigma^{-1}\bar{Y}$, where $\Sigma = \text{diag}(\sigma^2_{111}/n_{111}, \sigma^2_{112}/n_{112}, ...\sigma^2_{abc}/n_{abc})$ and X_* is a matrix of indicators corresponding to each group mean, discussed further for each parameter in the upcoming sections.

For a three factor ANOVA model, if σ^2_{ijk}'s are known, $\Sigma = \text{diag}(\sigma^2_{111}/n_{111}, \sigma^2_{112}/n_{112}, ...\sigma^2_{abc}/n_{abc})$, and the null hypothesis $H_0 : Par = 0$ is true (under the null hypothesis, the χ^2 non-centrality parameter is 0), then a natural test statistic for testing H_0 is S_I, the standardized sum of squares for the term being tested, which as discussed above, follows a χ^2 distribution with $r(X)-r(X_0)$ degrees of freedom. In general, variances are unknown, so we replace S_I with the test statistic $\tilde{S}_I = \bar{Y}'S^{-1}\bar{Y} - \bar{Y}'S^{-1}X_*(X'_*S^{-1}X_*)^-X'_*S^{-1}\bar{Y}$, where $S = \text{diag}(s^2_{111}/n_{111}, s^2_{112}/n_{112}, ...s^2_{abc}/n_{abc})$, and $s^2_{ijk} = \frac{1}{n_{ijk}-1}\sum_{m=1}^{n_{ijk}}(y_{ijkm} - \bar{y}_{ijk})^2$.

In this case, since the variances are unequal and unknown, the test statistic no longer follows a known distribution. The overall idea of a PB approach to this problem is to simulate a distribution for $\tilde{S}_I$ under the null hypothesis.

Each of Algorithms 1 – 4 follows the same procedure for testing each null hypothesis $H_0 : Par = 0$, with Par the applicable parameter. This procedure involves (1) calculate the test statistic $\tilde{S}_I$ above, (2) simulate a distribution for $\tilde{S}_I$ under H_0, and (3) calculate a Monte Carlo estimate of a p-value: the proportion of the simulated null distribution that is at least as extreme as the test statistic. This p-value can be used in the typical manner to reject or not reject the null hypothesis pertaining to the model term (parameter) we are investigating. In each algorithm 1 – 4, the X_* matrix in $\tilde{S}_I$ changes to reflect each reduced model; otherwise these algorithms are the same at each step.

For multiple comparisons of levels of a factor, Algorithms 5 and 6 are analogous to Tukey's test, but Tukey's test is intended for cases where the equal variance assumption is met and group sizes are equal. The Tukey-Kramer procedure does allow for different sample sizes (Kutner et al., 2005; Kramer, 1956; Hayter, 1984), and the documentation for the `'TukeyHSD'` procedure in R (R Core Team, 2021) states that the results are valid for mildly unbalanced data (use of this R function for the data given in Kutner et al. (2005), Example 2 of Section 17.5, gives very

similar results to those shown for the Tukey-Kramer procedure used for the same data). When the equal variance assumption is met, Tukey's test statistic can be compared to the studentized range distribution, but if not, we no longer have a standard distribution for comparison of the test statistic, so the PB method is used to simulate a null distribution. Figure 3.1 depicts the overall procedure for a three-factor heteANOVA problem using these PB algorithms. R code for Algorithms 1 – 6 is shown in the Appendices.

3.3 Illustration Of PB for Three-Factor ANOVA

Consider the three factor ANOVA full model, with all interactions and main effects:

$$y_{ijkm} = G + A_i + B_j + C_k + [AB]_{ij} + [AC]_{ik} + [BC]_{jk} + [ABC]_{ijk} + e_{ijkm}, \quad (3.1)$$

where G indicates the grand mean, A, B, and C indicate main effects, $[AB]$, $[AC]$, and $[BC]$ indicate two-way interaction terms, and $[ABC]$ indicates the three-way interaction term. Also, we assume $e_{ijkm} \overset{ind}{\sim} N(0, \sigma^2_{ijk})$, and for identifiability, we assume the constraints $\sum_i w_i A_i = 0$, $\sum_j v_j B_j = 0$, $\sum_k u_k C_k = 0$, ...,$\sum_i w_i [ABC]_{ijk} = 0$, $\sum_j v_j [ABC]_{ijk} = 0$, $\sum_k u_k [ABC]_{ijk} = 0$, where the w's, v's, and u's are non-negative weights; for example, as discussed in Section 4.5 of Scheffe (1959) or Chapter 7 of Arnold (1981).

Define the vector of means, $\bar{Y} = (\bar{y}_{111}, \bar{y}_{112}, ..., \bar{y}_{121}, \bar{y}_{122}, ..., \bar{y}_{abc})'$, indicating the sample means of the observations from each factor level and combination of factor levels. Define the vector of sample variances for each combination of factor levels to be $\boldsymbol{s}^2_{ijk} = (s^2_{111}, s^2_{112}, ...s^2_{abc})'$, and the matrix

$S_{abc \times abc} = \text{diag}(s^2_{111}/n_{111}, s^2_{112}/n_{112}, ...s^2_{abc}/n_{abc})$.

Following the procedure in Figure 3.1, we test each term in model 3.1, from

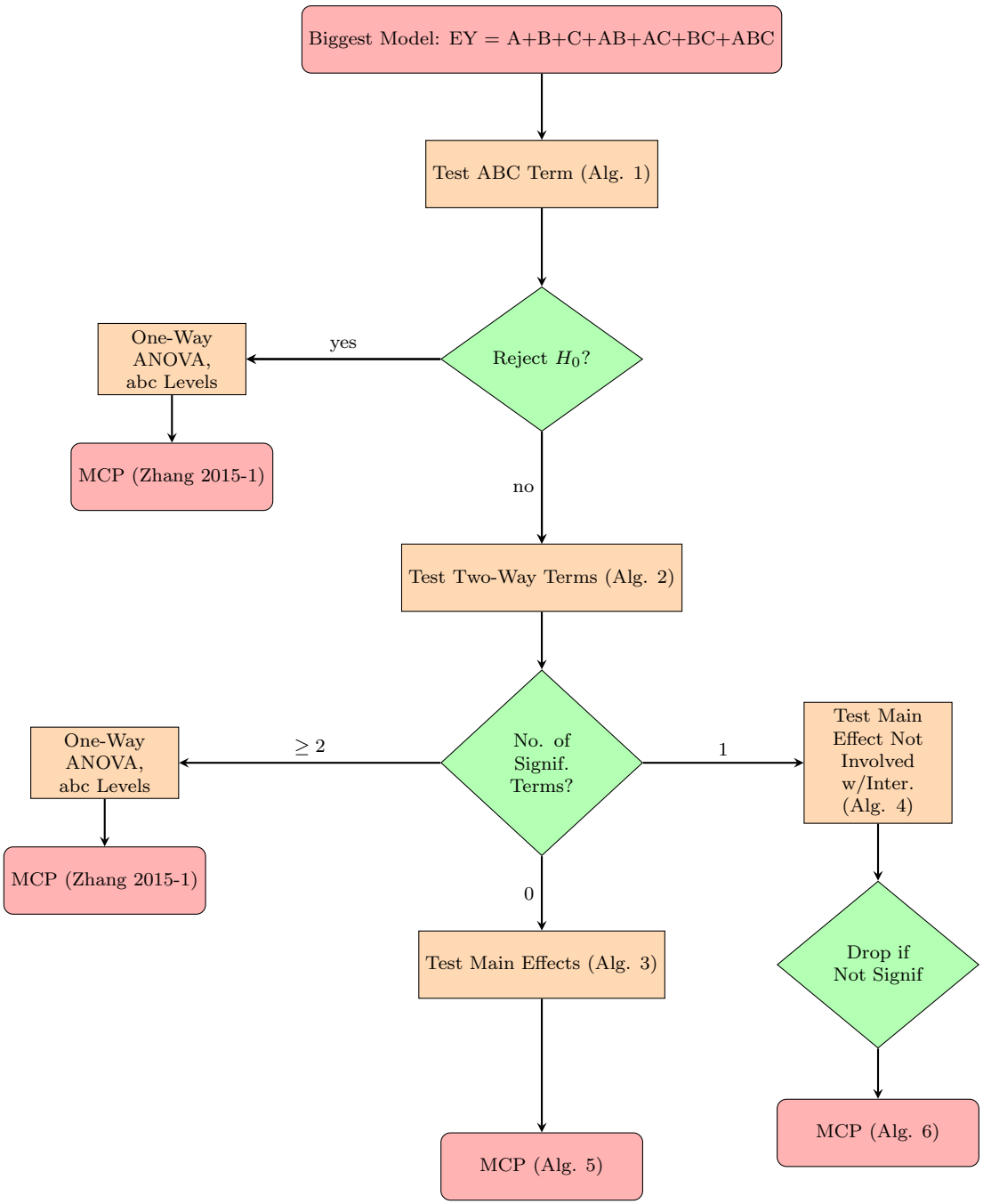

Figure 3.1: Overall Process: Three-Way ANOVA Using Parametric Bootstrap.

highest order to lowest order. Reduced models for each stage of testing are shown in corresponding subsections. For each term in the model, if σ^2_{ijk}'s are known, $\Sigma = \text{diag}(\sigma^2_{111}/n_{111}, \sigma^2_{112}/n_{112}, ...\sigma^2_{abc}/n_{abc})$, and the null hypothesis is true (under the null hypothesis, the χ^2 non-centrality parameter is 0), then a natural test statistic for testing H_0 is the standardized sum of squares for the term being tested (and higher order terms):

$\bar{Y}'\Sigma^{-1}\bar{Y} - \bar{Y}'\Sigma^{-1}X_*(X_*'\Sigma^{-1}X_*)^-X_*'\Sigma^{-1}\bar{Y} \sim \chi^2_{abc-r(X_*)}$, where X_* refers to e.g., X_{ABC} for the three-way interaction term, X_{BC} for the BC interaction term, and X_C for the main effects for factor C as described below.

The matrix X_* consists of a column of 1's for the grand mean and $(0, 1)$ indicators for membership in each factor level and combination of factor levels. Note that this matrix is indicating the levels for the group means, not each observation, so it should not be confused with the design matrix for the full data, which would include replications for each group. X_* can be expressed using Kronecker products. Let J_n indicate a column vector of n 1's, and I_n indicate an $n \times n$ identity matrix. Then, for example, $X_{ABC} = ([J_{abc}, I_a \otimes J_{bc}, J_a \otimes (I_b \otimes J_c), J_a \otimes (J_b \otimes I_c)], [I_{ab} \otimes J_c], [I_a \otimes (J_b \otimes I_c)], [J_a \otimes I_{bc}])$.

For example, suppose there are $a = 4$ levels for factor A, $b = 3$ levels for factor B and $c = 2$ levels for factor C, so $i = 1, 2, 3, 4$, $j = 1, 2, 3$, and $k = 1, 2$. The index m corresponds to the observations within each group, so $m = 1, ..., n_{ijk}$, with n_{ijk} observations in each factor level combination. Then the vector of means, $\bar{Y}$, has $abc = 24$ entries:

$\bar{Y} = (\bar{y}_{111}, \bar{y}_{112}, \bar{y}_{121}, \bar{y}_{122}, \bar{y}_{131}, \bar{y}_{132}, \bar{y}_{211}, \bar{y}_{212}, \bar{y}_{221}, \bar{y}_{222}, \bar{y}_{231}, \bar{y}_{232}, \bar{y}_{311}, \bar{y}_{312}, \bar{y}_{321}, \bar{y}_{322}, \bar{y}_{331}, \bar{y}_{332}, \bar{y}_{411}, \bar{y}_{412}, \bar{y}_{421}, \bar{y}_{422}, \bar{y}_{431}, \bar{y}_{432})'$,

indicating the sample means of the observations from all factor level combinations.

As before, the sample variance of each combination of factor levels is denoted $s^2_{ijk} = (s^2_{111}, s^2_{112}, ...s^2_{432})'$, and the matrix $S_{24\times 24}$ has diagonal entries $(s^2_{111}/n_{111}, s^2_{112}/n_{112}, ...s^2_{432}/n_{432})$, zeros elsewhere.

Structure of X_{ABC}

As noted before, X_{ABC} can be expressed using Kronecker products. X_{ABC} has a relatively large number of columns for this example: the dimensions of X_{ABC} would be $abc \times (1 + a + b + c + ab + ac + bc) = 24 \times 36$, so we will break it into parts starting with main effects and followed by each two-way interaction.

The columns of X_{ABC} corresponding to the grand mean and the main effects terms, the first $1 + a + b + c$ columns, can be written as $[J_{abc}, I_a \otimes J_{bc}, J_a \otimes (I_b \otimes J_c), J_a \otimes (J_b \otimes I_c)]$. For this example with $a = 4,\ b = 3,\ c = 2$, this part of the matrix would be:

ijk	G	A_1	A_2	A_3	A_4	B_1	B_2	B_3	C_1	C_2
111	1	1	0	0	0	1	0	0	1	0
112	1	1	0	0	0	1	0	0	0	1
121	1	1	0	0	0	0	1	0	1	0
122	1	1	0	0	0	0	1	0	0	1
131	1	1	0	0	0	0	0	1	1	0
132	1	1	0	0	0	0	0	1	0	1
211	1	0	1	0	0	1	0	0	1	0
212	1	0	1	0	0	1	0	0	0	1
221	1	0	1	0	0	0	1	0	1	0
222	1	0	1	0	0	0	1	0	0	1
231	1	0	1	0	0	0	0	1	1	0
232	1	0	1	0	0	0	0	1	0	1
311	1	0	0	1	0	1	0	0	1	0
312	1	0	0	1	0	1	0	0	0	1
321	1	0	0	1	0	0	1	0	1	0
322	1	0	0	1	0	0	1	0	0	1
331	1	0	0	1	0	0	0	1	1	0
332	1	0	0	1	0	0	0	1	0	1
411	1	0	0	0	1	1	0	0	1	0
412	1	0	0	0	1	1	0	0	0	1
421	1	0	0	0	1	0	1	0	1	0
422	1	0	0	0	1	0	1	0	0	1
431	1	0	0	0	1	0	0	1	1	0
432	1	0	0	0	1	0	0	1	0	1

Similarly, the columns of X_{ABC} corresponding to the AB interaction effects, columns $1+a+b+c+1, ..., 1+a+b+c+ab$, can be written as $I_{ab} \otimes J_c$. For this example with $a = 4,\ b = 3,\ c = 2$, this part of the matrix would be:

ijk	AB_{11}	AB_{12}	AB_{13}	AB_{21}	AB_{22}	AB_{23}	AB_{31}	AB_{32}	AB_{33}	AB_{41}	AB_{42}	AB_{43}
111	1	0	0	0	0	0	0	0	0	0	0	0
112	1	0	0	0	0	0	0	0	0	0	0	0
121	0	1	0	0	0	0	0	0	0	0	0	0
122	0	1	0	0	0	0	0	0	0	0	0	0
131	0	0	1	0	0	0	0	0	0	0	0	0
132	0	0	1	0	0	0	0	0	0	0	0	0
211	0	0	0	1	0	0	0	0	0	0	0	0
212	0	0	0	1	0	0	0	0	0	0	0	0
221	0	0	0	0	1	0	0	0	0	0	0	0
222	0	0	0	0	1	0	0	0	0	0	0	0
231	0	0	0	0	0	1	0	0	0	0	0	0
232	0	0	0	0	0	1	0	0	0	0	0	0
311	0	0	0	0	0	0	1	0	0	0	0	0
312	0	0	0	0	0	0	1	0	0	0	0	0
321	0	0	0	0	0	0	0	1	0	0	0	0
322	0	0	0	0	0	0	0	1	0	0	0	0
331	0	0	0	0	0	0	0	0	1	0	0	0
332	0	0	0	0	0	0	0	0	1	0	0	0
411	0	0	0	0	0	0	0	0	0	1	0	0
412	0	0	0	0	0	0	0	0	0	1	0	0
421	0	0	0	0	0	0	0	0	0	0	1	0
422	0	0	0	0	0	0	0	0	0	0	1	0
431	0	0	0	0	0	0	0	0	0	0	0	1
432	0	0	0	0	0	0	0	0	0	0	0	1

The columns of X_{ABC} corresponding to the AC interaction effects, columns $1+a+b+c+ab+1, ..., 1+a+b+c+ab+ac$, can be written as $I_a \otimes (J_b \otimes I_c)$. For this example with $a = 4,\ b = 3,\ c = 2$, this part of the matrix would be:

ijk	AC_{11}	AC_{12}	AC_{21}	AC_{22}	AC_{31}	AC_{32}	AC_{41}	AC_{42}
111	1	0	0	0	0	0	0	0
112	0	1	0	0	0	0	0	0
121	1	0	0	0	0	0	0	0
122	0	1	0	0	0	0	0	0
131	1	0	0	0	0	0	0	0
132	0	1	0	0	0	0	0	0
211	0	0	1	0	0	0	0	0
212	0	0	0	1	0	0	0	0
221	0	0	1	0	0	0	0	0
222	0	0	0	1	0	0	0	0
231	0	0	1	0	0	0	0	0
232	0	0	0	1	0	0	0	0
311	0	0	0	0	1	0	0	0
312	0	0	0	0	0	1	0	0
321	0	0	0	0	1	0	0	0
322	0	0	0	0	0	1	0	0
331	0	0	0	0	1	0	0	0
332	0	0	0	0	0	1	0	0
411	0	0	0	0	0	0	1	0
412	0	0	0	0	0	0	0	1
421	0	0	0	0	0	0	1	0
422	0	0	0	0	0	0	0	1
431	0	0	0	0	0	0	1	0
432	0	0	0	0	0	0	0	1

The columns of X_{ABC} corresponding to the BC interaction effects, columns $1+a+b+c+ab+ac+1, ..., 1+a+b+c+ab+ac+bc$, can be written as $J_a \otimes I_{bc}$. For this example with $a = 4,\ b = 3,\ c = 2$, this part of the matrix would be:

ijk	BC_{11}	BC_{12}	BC_{21}	BC_{22}	BC_{31}	BC_{32}
111	1	0	0	0	0	0
112	0	1	0	0	0	0
121	0	0	1	0	0	0
122	0	0	0	1	0	0
131	0	0	0	0	1	0
132	0	0	0	0	0	1
211	1	0	0	0	0	0
212	0	1	0	0	0	0
221	0	0	1	0	0	0
222	0	0	0	1	0	0
231	0	0	0	0	1	0
232	0	0	0	0	0	1
311	1	0	0	0	0	0
312	0	1	0	0	0	0
321	0	0	1	0	0	0
322	0	0	0	1	0	0
331	0	0	0	0	1	0
332	0	0	0	0	0	1
411	1	0	0	0	0	0
412	0	1	0	0	0	0
421	0	0	1	0	0	0
422	0	0	0	1	0	0
431	0	0	0	0	1	0
432	0	0	0	0	0	1

Putting all of the above sub-matrices together,
$X_{ABC} = ([J_{abc}, I_a \otimes J_{bc}, J_a \otimes (I_b \otimes J_c), J_a \otimes (J_b \otimes I_c)], [I_{ab} \otimes J_c], [I_a \otimes (J_b \otimes I_c)], [J_a \otimes I_{bc}])$.
Returning to discussion of the test statistic $\bar{Y}'\Sigma^{-1}\bar{Y} - \bar{Y}'\Sigma^{-1}X_*(X_*'\Sigma^{-1}X_*)^-X_*'\Sigma^{-1}\bar{Y}$, note that in general, variances are unknown, so we replace Σ with S to form the test statistic introduced earlier: $\tilde{S}_I = \bar{Y}'S^{-1}\bar{Y} - \bar{Y}'S^{-1}X_*(X_*'S^{-1}X_*)^-X_*'S^{-1}\bar{Y}$. This idea will be shown more specifically for each parameter tested in later sections.

The test statistic $\tilde{S}_I$ is location invariant (Xu et al., 2013), so without loss of generality, take $E(\boldsymbol{Y}) = \mathbf{0}$. The PB variable can then be developed as follows. For a given $(\bar{y}_{111}, \bar{y}_{112}, ..., \bar{y}_{abc}; s^2_{111}, s^2_{112}, ..., s^2_{abc}))$, $\bar{y}_{Bijk} \sim N(0, s^2_{ijk}/n_{ijk})$, and $s^2_{Bijk} \sim \left(\frac{s^2_{ijk}}{n_{ijk}-1}\right)\chi^2_{n_{ijk}-1}$, $i = 1, ...a$, $j = 1, ..., b$, $k = 1, ...c$.

Let $\bar{Y}_B = (\bar{y}_{B111}, \bar{y}_{B112}, ..., \bar{y}_{Babc})'$ and $S_B = \text{diag}(s^2_{B111}/n_{111}, s^2_{B112}/n_{112}, ...s^2_{Babc}/n_{abc})$.

Then we can construct the PB pivot variable based on the test statistic $\tilde{S}_I$, replacing $\bar{Y}$ with $\bar{Y}_B$ and S with S_B:

$\tilde{S}_{IB} = \bar{Y}'_B S_B^{-1} \bar{Y}_B - \bar{Y}'_B S_B^{-1} X_* (X'_* S_B^{-1} X_*)^- X'_* S_B^{-1} \bar{Y}_B$. For a given level α, there is evidence that the main effects or interaction effects exist when $P(\tilde{S}_{IB} > \tilde{s}_I) < \alpha$, where $\tilde{s}_I$ is an observed value of $\tilde{S}_I$. This probability can be estimated by Algorithms 1 – 4 depending on the term being tested.

3.3.1 Testing Three-Way Interaction

For the three-way interaction term, consider model 3.1 and:

$H_{0ABC} : [ABC]_{ijk} = 0$ for $i = 1, ...a$, $j = 1, ..., b$, $k = 1, ...c$ vs

$H_{\alpha ABC} : [ABC]_{ijk} \neq 0$ for some i, j, k.

If σ^2_{ijk}'s are known, as discussed previously, a natural test statistic for testing H_0 is the standardized sum of squares for the three way interaction, a function of $(\bar{Y} - \hat{G} - \hat{A} - \hat{B} - \hat{C} - \hat{AB} - \hat{AC} - \hat{BC})$, where the terms $\hat{G}, ..., \hat{BC}$ are the parameter estimates from fitting all terms from model 3.1 other than the ABC term:

$$\bar{Y}'\Sigma^{-1}\bar{Y} - \bar{Y}'\Sigma^{-1} X_{ABC} (X'_{ABC}\Sigma^{-1} X_{ABC})^- X'_{ABC}\Sigma^{-1}\bar{Y} \sim \chi^2_{abc-r(X_{abc})}. \tag{3.2}$$

In general, variances are unknown, so we replace 3.2 with the following test

statistic:

$$\tilde{S}_I = \bar{Y}'S^{-1}\bar{Y} - \bar{Y}'S^{-1}X_{ABC}(X'_{ABC}S^{-1}X_{ABC})^{-}X'_{ABC}S^{-1}\bar{Y}. \tag{3.3}$$

This test statistic is location invariant (Xu et al., 2013), so without loss of generality, take $E(\boldsymbol{Y}) = \mathbf{0}$. We construct the PB pivot variable based on test statistic 3.3 , replacing $\bar{Y}$ with $\bar{Y}_B$ and S with S_B:

$$\tilde{S}_{IB} = \bar{Y}'_B S_B^{-1}\bar{Y}_B - \bar{Y}'_B S_B^{-1}X_{ABC}(X'_{ABC}S_B^{-1}X_{ABC})^{-}X'_{ABC}S_B^{-1}\bar{Y}_B \tag{3.4}$$

For a given level α, the test rejects H_{0ABC} when $P(\tilde{S}_{IB} > \tilde{s}_I) < \alpha$, where $\tilde{s}_I$ is an observed value of $\tilde{S}_I$ in 3.3. This probability can be estimated by Algorithm 1.

Algorithm 1:
For a given $(n_{111}, n_{112}, ..., n_{abc})$, $(\bar{y}_{111}, \bar{y}_{112}, ..., \bar{y}_{abc})$, and $(s^2_{111}, s^2_{112}, ...s^2_{abc})$, compute $\tilde{S}_I = \bar{Y}'S^{-1}\bar{Y} - \bar{Y}'S^{-1}X_{ABC}(X'_{ABC}S^{-1}X_{ABC})^{-}X'_{ABC}S^{-1}\bar{Y}$ and call it $\tilde{s}_I$.

For $l = 1, ..., L$:

Generate $\bar{y}_{Bijk} \sim N(0, s^2_{ijk}/n_{ijk})$, and

$s^2_{Bijk} \sim \left(\frac{s^2_{ijk}}{n_{ijk}-1}\right)\chi^2_{n_{ijk}-1}$, $i = 1, ...a$, $j = 1, ..., b$, $k = 1, ...c$,

Compute $\tilde{S}_{IB} = \bar{Y}'_B S_B^{-1}\bar{Y}_B - \bar{Y}'_B S_B^{-1}X_{ABC}(X'_{ABC}S_B^{-1}X_{ABC})^{-}X'_{ABC}S_B^{-1}\bar{Y}_B$,

If $\tilde{S}_{IB} > \tilde{s}_I$, set $Q_l = 1$,
(end loop)

$\frac{1}{L}\sum_{l=1}^{L} Q_l$ is a Monte Carlo estimate of the p-value $P(\tilde{S}_{IB} > \tilde{s}_I)$.

Hence, reject H_{0ABC} if $\frac{1}{L}\sum_{l=1}^{L} Q_l < \alpha$.

3.3.2 Testing Two-Way Interaction Terms

For the two-way interaction terms, if we do not reject H_0 for the ABC interaction term, we may drop this term and consider the model:

$$y_{ijkm} = G + A_i + B_j + C_k + [AB]_{ij} + [AC]_{ik} + [BC]_{jk} + e_{ijkm} \tag{3.5}$$

Note that if the three-way interaction term $[ABC]_{ijk}$ is equal to zero for all i, j, k, this model 3.5 is equivalent to the full model 3.1. Additionally, if we do not reject H_0 for the ABC interaction term, it would not be significantly different from zero, but weak/non-significant effects could be present. As discussed by Xu et al. (2013), when the three-way interaction is present, each two-way effect alone, for example the BC interaction, cannot reflect the effects of B and C because it depends on the level of the ABC interaction. So rather than testing $H_{0BC} : [BC]_{jk} = 0$, we are actually testing $H_{0BC} : [BC]_{jk} + [ABC]_{ijk} = 0$ as discussed below. When the interaction effects are zero, the tests for main effects do not depend on chosen weights; see Arnold (1981) for a discussion of this issue. In testing the two-way interaction term $[BC]$, the sum of squares for the BC and ABC interaction will be a function of $(\bar{Y} - \hat{G} - \hat{A} - \hat{B} - \hat{C} - \hat{AB} - \hat{AC})$, where the terms $\hat{G}, ..., \hat{AC}$ are the parameter estimates from fitting all terms from model 3.5 other than the BC term, i.e. from fitting the reduced model:

$$y_{ijkm} = G + A_i + B_j + C_k + [AB]_{ij} + [AC]_{ik} + e_{ijkm} \tag{3.6}$$

Similarly to the three-way interaction case, a natural test statistic for testing

$H_{0BC} : [BC]_{jk} + [ABC]_{ijk} = 0$ for $i = 1, ..., a,\ j = 1, ..., b,\ k = 1, ...c$ vs.
$H_{\alpha BC} : [BC]_{jk} + [ABC]_{ijk} \neq 0$ for some i, j, k

is the standardized sum of squares for the BC and ABC interaction term:

$\bar{Y}'\Sigma^{-1}\bar{Y} - \bar{Y}'\Sigma^{-1}X_{BC}(X'_{BC}\Sigma^{-1}X_{BC})^{-}X'_{BC}\Sigma^{-1}\bar{Y} \sim \chi^2_{abc-r(X_{BC})}$, where

$X_{BC} = ([J_{abc}, I_a \otimes J_{bc}, J_a \otimes (I_b \otimes J_c), J_a \otimes (J_b \otimes I_c)], [I_{ab} \otimes J_c], [I_a \otimes (J_b \otimes I_c)])$

For unknown Σ, the test statistic will be:

$$\tilde{S}_I = \bar{Y}'S^{-1}\bar{Y} - \bar{Y}'S^{-1}X_{BC}(X'_{BC}S^{-1}X_{BC})^{-}X'_{BC}S^{-1}\bar{Y} \tag{3.7}$$

The test statistic 3.7 is analogous to the general linear test of the reduced model 3.6 above, vs. the biggest model 3.1. The PB pivot variable for H_{0BC} is constructed based on test statistic 3.7, replacing $\bar{Y}$ with $\bar{Y}_B$ and S with S_B:

$$\tilde{S}_{IB} = \bar{Y}'_B S_B^{-1}\bar{Y}_B - \bar{Y}'_B S_B^{-1}X_{BC}(X'_{BC}S_B^{-1}X_{BC})^{-}X'_{BC}S_B^{-1}\bar{Y}_B \tag{3.8}$$

For a given level α, the test rejects H_{0BC} when $P(\tilde{S}_{IB} > \tilde{s}_I) < \alpha$, where $\tilde{s}_I$ is an observed value of $\tilde{S}_I$ in 3.7. This probability can be estimated by Algorithm 2. Algorithm 2 should be used three times to test each two-way interaction term and is similar for each term. The X-matrix in 3.7 and 3.8 should be replaced to reflect the term under testing as follows:

$X_{AC} = [J_{abc}, I_a \otimes J_{bc}, J_a \otimes (I_b \otimes J_c), J_a \otimes (J_b \otimes I_c), I_{ab} \otimes J_c, J_a \otimes I_{bc}]$.

$X_{AB} = [J_{abc}, I_a \otimes J_{bc}, J_a \otimes (I_b \otimes J_c), J_a \otimes (J_b \otimes I_c), I_a \otimes (J_b \otimes I_c), J_a \otimes I_{bc}]$.

Algorithm 2 is identical to Algorithm 1 except that X_{BC}, X_{AC} or X_{AB} replaces X_{ABC} in the calculation of $\tilde{S}_I$ and $\tilde{S}_{IB}$. Note that it is also possible to specify the X matrix to perform sequential testing of the two-way interaction terms; e.g. the model with terms $G,\ A,\ B,\ C,\ AB,\ BC$ vs. the model with just the terms $G,\ A,\ B,\ C,\ AB$; however, we did not perform simulations for this procedure.

3.3.3 Testing Main Effects, No Significant Interaction Terms

If we do not reject H_0 for any of the interaction terms, we drop these terms and consider the model

$$y_{ijkm} = G + A_i + B_j + C_k + e_{ijkm} \tag{3.9}$$

In testing the main effect term C, the sum of squares for C and the interactions will be a function of $(\bar{Y} - \hat{G} - \hat{A} - \hat{B})$, where the terms $\hat{G},\ \hat{A}$ and $\hat{B}$ are the parameter estimates from fitting all terms from model 3.9 other than the C term, i.e. from fitting the reduced model:

$$y_{ijm} = G + A_i + B_j + e_{ijm} \tag{3.10}$$

A natural test statistic for testing

$H_{0C} : C_k + [AB]_{ij} + [AC]_{ik} + [BC]_{jk} + [ABC]_{ijk} = 0$
for $i = 1, ..., a,\ j = 1, ..., b,\ k = 1, ...c$ vs
$H_{\alpha C} : C_k + [AB]_{ij} + [AC]_{ik} + [BC]_{jk} + [ABC]_{ijk} \neq 0$ for some i, j, k

is the standardized sum of squares for C and the interaction terms:
$\bar{Y}'\Sigma^{-1}\bar{Y} - \bar{Y}'\Sigma^{-1}X_C(X_C'\Sigma^{-1}X_C)^{-}X_C'\Sigma^{-1}\bar{Y} \sim \chi^2_{abc-r(X_C)}$

For unknown Σ, the test statistic will be:

$$\tilde{S}_I = \bar{Y}'S^{-1}\bar{Y} - \bar{Y}'S^{-1}X_C(X_C'S^{-1}X_C)^{-}X_C'S^{-1}\bar{Y}, \tag{3.11}$$

where $X_C = [J_{abc}, I_a \otimes J_{bc}, J_a \otimes (I_b \otimes J_c)]$.

The PB pivot variable for H_{0C} is constructed based on the test statistic 3.11, replacing $\bar{Y}$ with $\bar{Y}_B$ and S with S_B:

$$\tilde{S}_{IB} = \bar{Y}_B'S_B^{-1}\bar{Y}_B - \bar{Y}_B'S_B^{-1}X_C(X_C'S_B^{-1}X_C)^{-}X_C'S_B^{-1}\bar{Y}_B \tag{3.12}$$

For a given level α, the test rejects H_{0BC} when $P(\tilde{S}_{IB} > \tilde{s}_I) < \alpha$, where $\tilde{s}_I$ is an observed value of $\tilde{S}_I$ in 3.11. This probability can be estimated by Algorithm 3. Algorithm 3 should be used three times to test each main effect term and is similar for each term. The X-matrix in 3.11 and 3.12 should be replaced to reflect the term under testing as follows:

$X_A = J_{abc},\ J_a \otimes (I_b \otimes J_c),\ J_a \otimes (J_b \otimes I_c)$

$X_B = J_{abc},\ I_a \otimes J_{bc},\ J_a \otimes (J_b \otimes I_c)$

Algorithm 3 is identical to Algorithm 1 except that we use X_A, X_B or X_C in place of X_{ABC} in the calculation of $\tilde{S}_I$ and $\tilde{S}_{IB}$.

3.3.4 Testing One Main Effect in Presence of One Significant Two-Way Interaction

If we do not reject H_0 for two of the interaction terms, but do reject for one of them, say AB, we drop the non-significant terms and consider the model

$$y_{ijkm} = G + A_i + B_j + C_k + [AB]_{ij} + e_{ijkm}, \tag{3.13}$$

which would be equivalent to model 3.1 if all interaction terms other than AB are zero. In testing the main effect term C when the AB interaction term is significantly different from 0, the sum of squares for C and the remaining interactions will be a function of $(\bar{Y} - \hat{G} - \hat{A} - \hat{B} - \hat{AB})$, where the terms $\hat{G}$, $\hat{A}$, $\hat{B}$ and $\hat{AB}$ are the parameter estimates from fitting all terms from model 3.13 other than the C term, i.e. from fitting the reduced model:

$$y_{ijkm} = G + A_i + B_j + [AB]_{ij} + e_{ijkm}. \tag{3.14}$$

Similarly to the previous cases, a natural test statistic for testing

$H_{0C*} : C_k + [AC]_{ik} + [BC]_{jk} + [ABC]_{ijk} = 0$
for $i = 1, ..., a,\ j = 1, ..., b,\ k = 1, ...c$ vs
$H_{\alpha C*} : C_k + [AC]_{ik} + [BC]_{jk} + [ABC]_{ijk} \neq 0$ for some i, j, k

is the standardized sum of squares for C and the interaction terms other than AB:

$\bar{Y}'\Sigma^{-1}\bar{Y} - \bar{Y}'\Sigma^{-1}X_{C*}(X'_{C*}\Sigma^{-1}X_{C*})^{-}X'_{C*}\Sigma^{-1}\bar{Y} \sim \chi^2_{abc-r(X_{C*})}$, where
$X_{C*} = [J_{abc}, I_a \otimes J_{bc}, J_a \otimes (I_b \otimes J_c), I_{ab} \otimes J_c]$

For unknown Σ, the test statistic will be:

$$\tilde{S}_I = \bar{Y}'S^{-1}\bar{Y} - \bar{Y}'S^{-1}X_{C*}(X'_{C*}S^{-1}X_{C*})^{-}X'_{C*}S^{-1}\bar{Y} \tag{3.15}$$

Similarly to the previous terms, for H_{0C*},we can construct the PB pivot variable based on test statistic 3.15, replacing $\bar{Y}$ with $\bar{Y}_B$ and S with S_B:

$$\tilde{S}_{IB} = \bar{Y}_B^T S_B^{-1}\bar{Y}_B - \bar{Y}_B^T S_B^{-1}X_{C*}(X_{C*}^T S_B^{-1}X_{C*})^{-}X_{C*}^T S_B^{-1}\bar{Y}_B \tag{3.16}$$

For a given level α, the test rejects H_{0C*} when $P(\tilde{S}_{IB} > \tilde{s}_I) < \alpha$, where $\tilde{s}_I$ is an observed value of $\tilde{S}_I$ in 3.15. This probability can be estimated by Algorithm 4. Algorithm 4 could be used to test any main effect term that is not involved in an interaction. To do so, the X-matrix in 3.15 and 3.16 should be replaced to reflect the term being tested, as follows:

$X_{A*} = [J_{abc}, J_a \otimes (I_b \otimes J_c), J_a \otimes (J_b \otimes I_c), J_a \otimes I_{bc}]$, where the reduced model is $y_{ijkm} = G + B_j + C_k + [BC]_{jk} + e_{ijkm}$;

$X_{B*} = [J_{abc}, I_a \otimes J_{bc}, J_a \otimes (J_b \otimes I_c), I_a \otimes (J_b \otimes I_c)]$, where the reduced model is $y_{ijkm} = G + A_i + C_k + [AC]_{ik} + e_{ijkm}$.

Algorithm 4 is identical to Algorithm 1 except that we use X_{A*}, X_{B*} or X_{C*} in place of X_{ABC} in the calculation of $\tilde{S}_I$ and $\tilde{S}_{IB}$.

3.3.5 Simulations for Testing Interaction and Main Effects Terms

For each term being tested, we consider model 3.1 and reduced models shown in the previous corresponding sections. For each simulation, datasets were generated under the reduced model with $e_{ijkm} \sim N(0, \sigma^2_{ijk})$, $i = 1, ..., a$, $j = 1, ..., b$, $k = 1, ..., c$, $G = 0$, and to meet the constraints $\sum_{i=1}^{a} A_i = 0$, $\sum_{j=1}^{b} B_j = 0$, $\sum_{k=1}^{c} C_k = 0$, $\sum_{j=1}^{b} AB_{ij} = 0$, $\sum_{k=1}^{c} AC_{ik} = 0$, and $\sum_{k=1}^{c} BC_{jk} = 0$. The sample mean and sample variance vectors $(\bar{y}_{111}, \bar{y}_{112}, ..., \bar{y}_{abc})$, and $(s^2_{111}, s^2_{112}, ...s^2_{abc})$ were calculated from each simulated dataset. The simulation was performed with:

(1) $a = b = c = 2$ to form 8 combinations;

(2) population standard deviation $\boldsymbol{\sigma}_i = (\sigma_{111}, \sigma_{112}, ..., \sigma_{222})$:

$\boldsymbol{\sigma}^{\mathbf{2}}_1$ = (1, 1, 1, 1, 1, 1, 1, 1), $\boldsymbol{\sigma}^{\mathbf{2}}_2$ = (0.1, 0.1, 0.1, 0.1, 0.5, 0.5, 0.5, 0.5), $\boldsymbol{\sigma}^{\mathbf{2}}_3$ = (1,1,1,1,0.5, 0.5, 0.5, 0.5), $\boldsymbol{\sigma}^{\mathbf{2}}_4$ = (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 1), $\boldsymbol{\sigma}^{\mathbf{2}}_5$ = (0.1, 0.3, 0.9, 0.4, 0.7, 0.5, 0.6, 1), $\boldsymbol{\sigma}^{\mathbf{2}}_6$ = (0.01, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 1);

(3) significance level $\alpha = 0.05$ and $\alpha = 0.1$;

(4) group sizes $\boldsymbol{n}_i = (n_{111}, n_{112}, ..., n_{222})$: $\boldsymbol{n}_1$ = (5,5,5,5,5,5,5,5), $\boldsymbol{n}_2$ = (10, 10, 10, 10, 10, 10, 10, 10), $\boldsymbol{n}_3$ = (3,3,4,5,4,5,6,6), $\boldsymbol{n}_4$ = (4, 6, 8,12, 14, 16, 18, 20).

For a given sample size and population variance configuration, we generated 2500 datasets, calculated the observed vectors $(\bar{y}_{111}, \bar{y}_{112}, ..., \bar{y}_{abc})$, and $(s^2_{111}, s^2_{112}, ...s^2_{abc})$ from the datasets, and used 5000 PB runs to estimate the p-value using Algorithms 1-4 as indicated. The p-value for the F-test (general linear test discussed in Section 3.2) for each term was also calculated for each simulated dataset using the '`lm`' function in R (R Core Team, 2021). The tests were considered to reject if the p-value was less than α, and the proportions rejected out of the 2500 datasets were

calculated for both the algorithm and the F-test, and shown in Tables 3.1 – 3.4.

For simulations for the three-way interaction, datasets were generated under the reduced model 3.5. Algorithm 1 was used to calculate the simulated p-value for the PB test, and the F-test comparing the reduced model with model 3.1 was calculated as described above; results shown in Table 3.1. Similarly: for the BC interaction term, model 3.6 was the reduced model and Algorithm 2 was used - results shown in Table 3.2; for the main effect C, model 3.10 was the reduced model and Algorithm 3 was used - results shown in Table 3.3; and for simulations of testing one main effect when one two-way term is significant, model 3.14 was the reduced model and Algorithm 4 was used - results shown in Table 3.4. We see from these tables that the F-test does not work well for some cases, but the PB test is robust; simulation results are discussed further in Section 3.6.

Table 3.1: Simulation Results for Testing ABC Interaction.

Numbers in the table are simulated p-values, with four different sizes and six different variance vectors: $\boldsymbol{n}_1 = (5,5,5,5,5,5,5,5); \boldsymbol{n}_2 = (10,10,10,10,10,10,10,10); \boldsymbol{n}_3 = (3,3,4,5,4,5,6,6);$ $\boldsymbol{n}_4 = (4,6,8,12,14,16,18,20); \boldsymbol{\sigma_1^2} = (1,1,1,1,1,1,1,1); \boldsymbol{\sigma_2^2} = (0.1,0.1,0.1,0.1,0.5,0.5,0.5,0.5);$ $\boldsymbol{\sigma_3^2} = (1,1,1,1,0.5,0.5,0.5,0.5); \boldsymbol{\sigma_4^2} = (0.1,0.2,0.3,0.4,0.5,0.6,0.7,1);$ $\boldsymbol{\sigma_5^2} = (0.1,0.3,0.9,0.4,0.7,0.5,0.6,1);$ $\boldsymbol{\sigma_6^2} = (0.01,0.1,0.1,0.1,0.1,0.1,0.1,1)$, and the two different α levels shown.

		$\boldsymbol{\alpha = 0.05}$		$\boldsymbol{\alpha = 0.1}$	
$\boldsymbol{\sigma_1^2}$		F-test	Algorithm	F-test	Algorithm
	$\boldsymbol{n}_1$	0.0576	0.0528	0.1104	0.1080
	$\boldsymbol{n}_2$	0.0504	0.0492	0.1096	0.1092
	$\boldsymbol{n}_3$	0.0556	0.0536	0.1068	0.0988
	$\boldsymbol{n}_4$	0.0528	0.0476	0.1056	0.1020
$\boldsymbol{\sigma_2^2}$		F-test	Algorithm	F-test	Algorithm
	$\boldsymbol{n}_1$	0.0412	0.0348	0.1088	0.1012
	$\boldsymbol{n}_2$	0.0568	0.0556	0.0936	0.0944
	$\boldsymbol{n}_3$	0.0248	0.0412	0.0572	0.0848
	$\boldsymbol{n}_4$	0.0088	0.0528	0.0256	0.0972
$\boldsymbol{\sigma_3^2}$		F-test	Algorithm	F-test	Algorithm
	$\boldsymbol{n}_1$	0.0492	0.0468	0.1060	0.1008
	$\boldsymbol{n}_2$	0.0504	0.0500	0.0992	0.0984
	$\boldsymbol{n}_3$	0.0688	0.0456	0.1252	0.0960
	$\boldsymbol{n}_4$	0.1060	0.0544	0.1400	0.0940
$\boldsymbol{\sigma_4^2}$		F-test	Algorithm	F-test	Algorithm
	$\boldsymbol{n}_1$	0.0556	0.0500	0.1028	0.0968
	$\boldsymbol{n}_2$	0.0504	0.0496	0.1044	0.1032
	$\boldsymbol{n}_3$	0.0232	0.0416	0.0580	0.0928
	$\boldsymbol{n}_4$	0.0100	0.0636	0.0252	0.1000
$\boldsymbol{\sigma_5^2}$		F-test	Algorithm	F-test	Algorithm
	$\boldsymbol{n}_1$	0.0552	0.0504	0.0900	0.0832
	$\boldsymbol{n}_2$	0.0560	0.0548	0.0960	0.0940
	$\boldsymbol{n}_3$	0.0340	0.0460	0.0816	0.1032
	$\boldsymbol{n}_4$	0.0176	0.0484	0.0464	0.1056
$\boldsymbol{\sigma_6^2}$		F-test	Algorithm	F-test	Algorithm
	$\boldsymbol{n}_1$	0.0696	0.0576	0.1180	0.1036
	$\boldsymbol{n}_2$	0.0564	0.0500	0.1192	0.1124
	$\boldsymbol{n}_3$	0.0244	0.0452	0.0472	0.0904
	$\boldsymbol{n}_4$	0.0060	0.0404	0.0160	0.1048

Table 3.2: Simulation Results for Testing BC + ABC Interaction.

Numbers in the table are simulated p-values, with four different sizes and six different variance vectors: $\boldsymbol{n}_1 = (5,5,5,5,5,5,5,5); \boldsymbol{n}_2 = (10,10,10,10,10,10,10,10); \boldsymbol{n}_3 = (3,3,4,5,4,5,6,6);$ $\boldsymbol{n}_4 = (4,6,8,12,14,16,18,20); \boldsymbol{\sigma_1^2} = (1,1,1,1,1,1,1,1); \boldsymbol{\sigma_2^2} = (0.1,0.1,0.1,0.1,0.5,0.5,0.5,0.5);$ $\boldsymbol{\sigma_3^2} = (1,1,1,1,0.5,0.5,0.5,0.5); \boldsymbol{\sigma_4^2} = (0.1,0.2,0.3,0.4,0.5,0.6,0.7,1);$ $\boldsymbol{\sigma_5^2} = (0.1,0.3,0.9,0.4,0.7,0.5,0.6,1)$; $\boldsymbol{\sigma_6^2} = (0.01,0.1,0.1,0.1,0.1,0.1,0.1,1)$, and the two different α levels shown.

		$\boldsymbol{\alpha = 0.05}$		$\boldsymbol{\alpha = 0.1}$	
$\boldsymbol{\sigma_1^2}$		F-test	Algorithm	F-test	Algorithm
	$\boldsymbol{n}_1$	0.0504	0.0440	0.1040	0.0864
	$\boldsymbol{n}_2$	0.0528	0.0496	0.1044	0.1052
	$\boldsymbol{n}_3$	0.0500	0.0452	0.1040	0.0920
	$\boldsymbol{n}_4$	0.0528	0.0512	0.1008	0.0920
$\boldsymbol{\sigma_2^2}$		F-test	Algorithm	F-test	Algorithm
	$\boldsymbol{n}_1$	0.0784	0.0464	0.1112	0.0936
	$\boldsymbol{n}_2$	0.0688	0.0456	0.1152	0.0948
	$\boldsymbol{n}_3$	0.0496	0.0452	0.0864	0.0832
	$\boldsymbol{n}_4$	0.0344	0.0492	0.0644	0.0900
$\boldsymbol{\sigma_3^2}$		F-test	Algorithm	F-test	Algorithm
	$\boldsymbol{n}_1$	0.0568	0.0416	0.0980	0.0908
	$\boldsymbol{n}_2$	0.0544	0.0536	0.1020	0.0956
	$\boldsymbol{n}_3$	0.0652	0.0424	0.1288	0.0932
	$\boldsymbol{n}_4$	0.0812	0.0468	0.1392	0.0996
$\boldsymbol{\sigma_4^2}$		F-test	Algorithm	F-test	Algorithm
	$\boldsymbol{n}_1$	0.0624	0.0480	0.1168	0.0960
	$\boldsymbol{n}_2$	0.0640	0.0424	0.0980	0.0908
	$\boldsymbol{n}_3$	0.0388	0.0468	0.0720	0.0852
	$\boldsymbol{n}_4$	0.0320	0.0420	0.0612	0.0948
$\boldsymbol{\sigma_5^2}$		F-test	Algorithm	F-test	Algorithm
	$\boldsymbol{n}_1$	0.0536	0.0392	0.0984	0.0844
	$\boldsymbol{n}_2$	0.0456	0.0420	0.1048	0.1040
	$\boldsymbol{n}_3$	0.0432	0.0460	0.0852	0.0972
	$\boldsymbol{n}_4$	0.0280	0.0464	0.0604	0.0976
$\boldsymbol{\sigma_6^2}$		F-test	Algorithm	F-test	Algorithm
	$\boldsymbol{n}_1$	0.0816	0.0500	0.1340	0.1008
	$\boldsymbol{n}_2$	0.0784	0.0496	0.1160	0.0996
	$\boldsymbol{n}_3$	0.0336	0.0448	0.0720	0.0932
	$\boldsymbol{n}_4$	0.0188	0.0512	0.0360	0.0972

Table 3.3: Simulation Results for Testing Main Effect C and Interactions.

Numbers in the table are simulated p-values, with four different sizes and six different variance vectors: $\boldsymbol{n}_1 = (5,5,5,5,5,5,5,5); \boldsymbol{n}_2 = (10,10,10,10,10,10,10,10); \boldsymbol{n}_3 = (3,3,4,5,4,5,6,6);$ $\boldsymbol{n}_4 = (4,6,8,12,14,16,18,20); \boldsymbol{\sigma_1^2} = (1,1,1,1,1,1,1,1); \boldsymbol{\sigma_2^2} = (0.1,0.1,0.1,0.1,0.5,0.5,0.5,0.5);$ $\boldsymbol{\sigma_3^2} = (1,1,1,1,0.5,0.5,0.5,0.5); \boldsymbol{\sigma_4^2} = (0.1,0.2,0.3,0.4,0.5,0.6,0.7,1);$ $\boldsymbol{\sigma_5^2} = (0.1,0.3,0.9,0.4,0.7,0.5,0.6,1)$; $\boldsymbol{\sigma_6^2} = (0.01,0.1,0.1,0.1,0.1,0.1,0.1,1)$, and the two different α levels shown.

		$\boldsymbol{\alpha = 0.05}$		$\boldsymbol{\alpha = 0.1}$	
$\boldsymbol{\sigma_1^2}$		F-test	Algorithm	F-test	Algorithm
	$\boldsymbol{n}_1$	0.0484	0.0404	0.0996	0.0920
	$\boldsymbol{n}_2$	0.0468	0.0460	0.1056	0.1016
	$\boldsymbol{n}_3$	0.0400	0.0376	0.1040	0.0900
	$\boldsymbol{n}_4$	0.0520	0.0524	0.1048	0.1084
$\boldsymbol{\sigma_2^2}$		F-test	Algorithm	F-test	Algorithm
	$\boldsymbol{n}_1$	0.0672	0.0420	0.1232	0.0984
	$\boldsymbol{n}_2$	0.0652	0.0480	0.1092	0.0872
	$\boldsymbol{n}_3$	0.0496	0.0400	0.0780	0.0876
	$\boldsymbol{n}_4$	0.0264	0.0508	0.0524	0.0956
$\boldsymbol{\sigma_3^2}$		F-test	Algorithm	F-test	Algorithm
	$\boldsymbol{n}_1$	0.0560	0.0424	0.1084	0.0936
	$\boldsymbol{n}_2$	0.0644	0.0540	0.1160	0.0972
	$\boldsymbol{n}_3$	0.0836	0.0400	0.1252	0.0872
	$\boldsymbol{n}_4$	0.0988	0.0460	0.1704	0.1008
$\boldsymbol{\sigma_4^2}$		F-test	Algorithm	F-test	Algorithm
	$\boldsymbol{n}_1$	0.0660	0.0432	0.1144	0.0820
	$\boldsymbol{n}_2$	0.0708	0.0480	0.1312	0.1084
	$\boldsymbol{n}_3$	0.0424	0.0440	0.0620	0.0792
	$\boldsymbol{n}_4$	0.0260	0.0532	0.0492	0.0988
$\boldsymbol{\sigma_5^2}$		F-test	Algorithm	F-test	Algorithm
	$\boldsymbol{n}_1$	0.0656	0.0388	0.1100	0.0880
	$\boldsymbol{n}_2$	0.0580	0.0460	0.1188	0.0964
	$\boldsymbol{n}_3$	0.0440	0.0460	0.1008	0.0920
	$\boldsymbol{n}_4$	0.0352	0.0460	0.0688	0.1036
$\boldsymbol{\sigma_6^2}$		F-test	Algorithm	F-test	Algorithm
	$\boldsymbol{n}_1$	0.1152	0.0524	0.1612	0.0968
	$\boldsymbol{n}_2$	0.1096	0.0504	0.1540	0.0988
	$\boldsymbol{n}_3$	0.0584	0.0488	0.0984	0.0968
	$\boldsymbol{n}_4$	0.0316	0.0460	0.0572	0.0896

Table 3.4: Simulation Results for Testing Main Effect C When AB Interaction Present.

Numbers in the table are simulated p-values, with four different sizes and six different variance vectors: $\boldsymbol{n}_1 = (5,5,5,5,5,5,5,5); \boldsymbol{n}_2 = (10,10,10,10,10,10,10,10); \boldsymbol{n}_3 = (3,3,4,5,4,5,6,6);$ $\boldsymbol{n}_4 = (4,6,8,12,14,16,18,20); \boldsymbol{\sigma_1^2} = (1,1,1,1,1,1,1,1); \boldsymbol{\sigma_2^2} = (0.1,0.1,0.1,0.1,0.5,0.5,0.5,0.5);$ $\boldsymbol{\sigma_3^2} = (1,1,1,1,0.5,0.5,0.5,0.5); \boldsymbol{\sigma_4^2} = (0.1,0.2,0.3,0.4,0.5,0.6,0.7,1);$ $\boldsymbol{\sigma_5^2} = (0.1,0.3,0.9,0.4,0.7,0.5,0.6,1);$ $\boldsymbol{\sigma_6^2} = (0.01,0.1,0.1,0.1,0.1,0.1,0.1,1)$, and the two different α levels shown.

		$\boldsymbol{\alpha = 0.05}$		$\boldsymbol{\alpha = 0.1}$	
$\boldsymbol{\sigma_1^2}$		F-test	Algorithm	F-test	Algorithm
	$\boldsymbol{n}_1$	0.0480	0.0404	0.1096	0.0888
	$\boldsymbol{n}_2$	0.0472	0.0468	0.0952	0.0972
	$\boldsymbol{n}_3$	0.0512	0.0412	0.0908	0.0812
	$\boldsymbol{n}_4$	0.0428	0.0404	0.1036	0.0940
$\boldsymbol{\sigma_2^2}$		F-test	Algorithm	F-test	Algorithm
	$\boldsymbol{n}_1$	0.0704	0.0396	0.1164	0.0832
	$\boldsymbol{n}_2$	0.0668	0.0504	0.1204	0.0956
	$\boldsymbol{n}_3$	0.0576	0.0344	0.0940	0.0756
	$\boldsymbol{n}_4$	0.0384	0.0568	0.0680	0.0948
$\boldsymbol{\sigma_3^2}$		F-test	Algorithm	F-test	Algorithm
	$\boldsymbol{n}_1$	0.0540	0.0472	0.1100	0.0908
	$\boldsymbol{n}_2$	0.0592	0.0528	0.1076	0.1032
	$\boldsymbol{n}_3$	0.0824	0.0376	0.1292	0.0768
	$\boldsymbol{n}_4$	0.0888	0.0472	0.1484	0.0904
$\boldsymbol{\sigma_4^2}$		F-test	Algorithm	F-test	Algorithm
	$\boldsymbol{n}_1$	0.0600	0.0400	0.1168	0.0908
	$\boldsymbol{n}_2$	0.0668	0.0424	0.1160	0.0972
	$\boldsymbol{n}_3$	0.0420	0.0428	0.0732	0.0840
	$\boldsymbol{n}_4$	0.0284	0.0472	0.0580	0.0956
$\boldsymbol{\sigma_5^2}$		F-test	Algorithm	F-test	Algorithm
	$\boldsymbol{n}_1$	0.0612	0.0408	0.1068	0.0916
	$\boldsymbol{n}_2$	0.0620	0.0520	0.1196	0.1080
	$\boldsymbol{n}_3$	0.0504	0.0392	0.0944	0.0888
	$\boldsymbol{n}_4$	0.0376	0.0436	0.0736	0.1016
$\boldsymbol{\sigma_6^2}$		F-test	Algorithm	F-test	Algorithm
	$\boldsymbol{n}_1$	0.1108	0.0452	0.1496	0.0988
	$\boldsymbol{n}_2$	0.0916	0.0572	0.1524	0.1064
	$\boldsymbol{n}_3$	0.0696	0.0500	0.1052	0.0948
	$\boldsymbol{n}_4$	0.0336	0.0444	0.0512	0.0984

3.4 Multiple Comparison Procedures

For the three-way ANOVA illustration in the previous sections, if the highest order term (i.e. the three-factor interaction term) is found to have a significant effect, or if two or more of the two-factor interaction terms are found to be significant, one can approach the problem as a one-way ANOVA problem with *abc* levels, and then perform multiple comparisons of factor level means. Approaching this problem using PB methods is described in detail by Zhang (2015b), which performs all pairwise comparisons of factor level means analogously to Tukey's test, but uses PB methods to allow for unequal variances. If there are no significant interaction terms but some main effects are found to be significant, all pairwise comparisons of the factor level means of the significant main effects may be of interest.

3.4.1 Multiple Comparisons for Main Effects Only

Consider simultaneous comparisons of the factor A level means when no interactions are present, i.e. in model 3.9. An estimator of the factor A level means, similar to the estimator described in Zhang (2015b) is a weighted average of the corresponding cell means:

$$\bar{Y}_{i\cdots} = \frac{\sum_j \sum_k v_{jk} \bar{Y}_{ijk}}{\sum_j \sum_k v_{jk}}, \text{ where } v_{jk} = \frac{\sum_i n_{ijk}}{N}, \tag{3.17}$$

with N the total number of observations.

The variance of these estimators is found to be $V(\bar{Y}_{i\cdots}) = \frac{1}{(\sum_{j,k} v_{jk})^2} \sum_j \sum_k v_{jk}^2 \frac{\sigma_{ijk}^2}{n_{ijk}}$ with the estimated variance

$$\hat{V}(\bar{Y}_{i\cdots}) = \frac{1}{(\sum_{j,k} v_{jk})^2} \sum_j \sum_k v_{jk}^2 \frac{s_{ijk}^2}{n_{ijk}}. \tag{3.18}$$

Similarly to Tukey's test, a test statistic for testing $H_0 : A_i = A_{i'}$ is

$q^A_{ii'} = \frac{|\bar{Y}_{i...} - \bar{Y}_{i'...}|}{\sqrt{\hat{V}(\bar{Y}_{i...}) + \hat{V}(\bar{Y}_{i'...})}}$

Since we have unequal variances and possibly also unbalanced data, the studentized range distribution typically used for Tukey's test is inappropriate. Thus, we use the PB method to simulate a distribution for the test statistic and for the confidence interval $\bar{y}_{i...} - \bar{y}_{i'...} \pm q^A_\alpha \sqrt{(\hat{V}(\bar{Y}_{i...}) + \hat{V}(\bar{Y}_{i'...}))}$, where q^A_α is the $1-\alpha$ percentile of the simulated distribution of q. The PB pivot variable for this procedure is based on the test statistic $q^A_{ii'}$, and can be developed as follows.

For a given $(\bar{y}_{111}, \bar{y}_{112}, ..., \bar{y}_{abc}, s^2_{111}, s^2_{112}, ..., s^2_{abc})$, $\bar{Y}_{Bijk} \sim N(0, s^2_{ijk}/n_{ijk})$, and $s^2_{Bijk} \sim \frac{s^2_{ijk}}{n_{ijk}-1}\chi^2_{(n_{ijk}-1)}$. In Algorithm 5 below, these variables are simulated. Then, $\bar{Y}_{Bi...}$ and $\bar{Y}_{Bi'...}$ can be calculated from $\bar{Y}_{Bijk}$ using 3.17, and the variances $V(\bar{Y}_{Bi...})$ and $V(\bar{Y}_{Bi'...})$ are as in 3.18 with s^2_{Bijk} taking the place of s^2_{ijk}. Thus, the PB pivot variable is:

$$q^A_{Bii'} = \frac{|\bar{Y}_{Bi...} - \bar{Y}_{Bi'...}|}{\sqrt{\hat{V}(\bar{Y}_{Bi...}) + \hat{V}(\bar{Y}_{Bi'...})}} \tag{3.19}$$

Algorithm 5

For a given $(\bar{y}_{111}, \bar{y}_{112}, ..., \bar{y}_{abc})$, $(s^2_{111}, s^2_{112}, ..., s^2_{abc})$, and $(n_{111}, n_{112}, ..., n_{abc})$:

For $l = 1, ..., L$

Generate $\bar{Y}_{Bijk} \sim N(0, s^2_{ijk}/n_{ijk})$ and $s^2_{Bijk} \sim \frac{s^2_{ijk}}{n_{ijk}-1}\chi^2_{n_{ijk}-1}$

Compute $q^A_{Bii'}$ using 3.19 for $i = 1, ..., a-1,\ i' = i+1, ...a$

Find $q_l = \max(q^A_{Bii'})$

(end loop)

q^A_α is the $1-\alpha$ percentile of the simulated distribution of q.

The procedure for simultaneous comparisons of the factor B or C level means,

when no interactions are present, is analogous to Algorithm 5.

3.4.2 Multiple Comparisons for Two-Way Interaction Term

Consider simultaneous comparisons of the levels of the AB interaction term in model 3.13. An estimator of the AB level means is a weighted average of the corresponding cell means, similar to the weights described in Zhang (2015b):

$$\bar{Y}_{ij..} = \sum_k v_k \bar{Y}_{ijk}, \text{ where } v_k = \frac{\sum_{i,j} n_{ijk}}{N}, \tag{3.20}$$

with N the total number of observations.

The variance of these estimators is found to be $V(\bar{Y}_{ij..}) = \sum_k v_k^2 \frac{\sigma_{ijk}^2}{n_{ijk}}$ with the estimated variance $\hat{V}(\bar{Y}_{ij..}) = \sum_k v_k^2 \frac{s_{ijk}^2}{n_{ijk}}$. Similarly to Tukey's test, a test statistic for testing $H_0 : AB_{ij} = AB_{i'j'}$ is

$$q_{iji'j'}^{AB} = \frac{|\bar{Y}_{ij..} - \bar{Y}_{i'j'..}|}{\sqrt{\hat{V}(\bar{Y}_{ij..}) + \hat{V}(\bar{Y}_{i'j'..})}} \tag{3.21}$$

Since we have unequal variances and possibly also unbalanced data, the studentized range distribution typically used for Tukey's test is inappropriate. Thus, the PB method is used to simulate a distribution for the test statistic and for the confidence interval $\bar{y}_{ij..} - \bar{y}_{i'j'..} \pm q_\alpha^{AB} \sqrt{(\hat{V}(\bar{Y}_{ij..}) + \hat{V}(\bar{Y}_{i'j'..}))}$, where q_α^{AB} is the $1-\alpha$ percentile of the simulated distribution of q. The PB pivot variable for this procedure is based on the test statistic $q_{iji'j'}^{AB}$, and can be developed as follows.

For a given $(\bar{y}_{111}, \bar{y}_{112}, ..., \bar{y}_{abc}, s_{111}^2, s_{112}^2, ..., s_{abc}^2)$, $\bar{Y}_{Bijk} \sim N(0, s_{ijk}^2/n_{ijk})$, and $s_{Bijk}^2 \sim \frac{s_{ijk}^2}{n_{ijk}-1} \chi_{(n_{ijk}-1)}^2$. In Algorithm 6 below, these variables are simulated. Then, $\bar{Y}_{Bij..}$ and $\bar{Y}_{Bi'j'..}$ can be calculated from $\bar{Y}_{Bijk}$ using 3.22 below, and the variances $V(\bar{Y}_{Bij..})$ and $V(\bar{Y}_{Bi'j'..})$ are as above with s_{Bijk}^2 taking the place of s_{ijk}^2. Thus, the PB pivot variable is:

$$q^{AB}_{Biji'j'} = \frac{|\bar{Y}_{Bij..} - \bar{Y}_{Bi'j'..}|}{\sqrt{\hat{V}(\bar{Y}_{Bij..}) + \hat{V}(\bar{Y}_{Bi'j'..})}} \tag{3.22}$$

Algorithm 6

For a given $(\bar{y}_{111}, \bar{y}_{112}, ..., \bar{y}_{abc})$, $(s^2_{111}, s^2_{112}, ..., s^2_{abc})$, and $(n_{111}, n_{112}, ..., n_{abc})$:

For $l = 1, ..., L$

Generate $\bar{Y}_{Bijk} \sim N(0, s^2_{ijk}/n_{ijk})$ and $s^2_{Bijk} \sim \frac{s^2_{ijk}}{n_{ijk}-1}\chi^2_{n_{ijk}-1}$

Compute $q^{AB}_{Biji'j'}$ using 3.22 for all pairs $(ij,\ i'j')$ where $ij \neq i'j'$.

Take q_l to be the max of the $q^{AB}_{Biji'j'}$ for the l^{th} run.

(end loop)

q^{AB}_{α} is the $1-\alpha$ percentile of the simulated distribution of q. Reject $H_0 : AB_{ij} = AB_{i'j'}$ if the test statistic 3.21 is greater than q^{AB}_{α}.

3.4.3 MCP Simulations

Datasets were generated under model 3.9 for simulating MCP for levels of Factor A and under model 3.13 for MCP for levels of the AB interaction term, assuming $E(Y) = 0$ for all factor levels (such that $H_0 : A_i = A_{i'}$ is true or $H_0 : AB_{ij} = AB_{i'j'}$ is true, respectively). For both simulations, the sample mean and sample variance vectors $(\bar{y}_{111}, \bar{y}_{112}, ..., \bar{y}_{abc})$, and $(s^2_{111}, s^2_{112}, ...s^2_{abc})$ were calculated from each simulated dataset. The simulations were performed with $a = 3,\ b = 2,\ c = 4$ to form 24 combinations, and the population variances and sample size scenarios as:

$\sigma^2_1 = (1, 1, ..., 1),\ \sigma^2_2 = (0.1, 0.1, ..., 0.1, 0.5, 0.5, ..., 0.5)$,

$\sigma_3^2 = (1, 1, ..., 1, 0.5, 0.5, ..., 0.5)$, $\sigma_4^2 = (0.1,\ 0.1,\ 0.1,\ 0.2,\ 0.2,\ 0.2,\ 0.3,\ 0.3,\ 0.3,\ 0.4,\ 0.4,\ 0.4,\ 0.5,\ 0.5,\ 0.5,\ 0.6,\ 0.6,\ 0.6,\ 0.7,\ 0.7,\ 0.7,\ 1,\ 1,\ 1)$, $\sigma_5^2 = (0.1,\ 0.1,\ 0.1,\ 0.3,\ 0.3,\ 0.3,\ 0.9,\ 0.9,\ 0.9,\ 0.4,\ 0.4,\ 0.4,\ 0.7,\ 0.7,\ 0.7,\ 0.5,\ 0.5,\ 0.5,\ 0.6,\ 0.6,\ 0.6,\ 1,\ 1,\ 1)$, $\sigma_6^2 = (0.01,\ 0.01,\ 0.01,\ 0.1,\ 0.1,\ 0.1,\ 0.1,\ 0.1,\ 0.1,\ 0.1,\ 0.1,\ 0.1,\ 0.1,\ 0.1,\ 0.1,\ 0.1,\ 0.1,\ 0.1,\ 0.1,\ 0.1,\ 0.1,\ 1,\ 1,\ 1)$, and $n_1 = (5,\ 5,\ ...,\ 5)$, $n_2 = (10,\ 10,\ ...,\ 10)$, $n_3 = (3,\ 3,\ 3,\ 3,\ 3,\ 3,\ 4,\ 4,\ 4,\ 5,\ 5,\ 5,\ 4,\ 4,\ 4,\ 5,\ 5,\ 5,\ 6,\ 6,\ 6,\ 6,\ 6,\ 6)$, $n_4 = (4,\ 4,\ 4,\ 6,\ 6,\ 6,\ 8,\ 8,\ 8,\ 12,\ 12,\ 12,\ 14,\ 14,\ 14,\ 16,\ 16,\ 16,\ 18,\ 18,\ 18,\ 20,\ 20,\ 20)$.

Each scenario was simulated for $\alpha = 0.05$ and $\alpha = 0.1$. For the factor A MCP simulation, Tukey's test was also performed on factor A for each dataset using the `'TukeyHSD'` function in R, and on factor AB for the simulation for the AB term MCP. The smallest p-value for Tukey's test was checked and the test was considered to reject if this p-value was less than α. For the algorithms, the $1 - \alpha$ percentile was taken from the simulated PB distribution, and the test was considered to reject if the test statistic for the simulated dataset was greater than this percentile. The proportions rejected out of the 2500 datasets were calculated for both the algorithm and Tukey's test, and shown in Table 3.5 for the factor A MCP simulation and Table 3.6 for the AB simulation. We see from these tables that Tukey's test does not work well for some cases, but the PB test is robust; simulation results are discussed further in Section 3.6.

Table 3.5: Results of Simulations For Testing Multiple Comparisons for Factor A.

Numbers in the table are simulated p-values. We consider four different sizes and six different variance vectors as shown in Section 3.4.3, with the two different α levels shown.

		$\boldsymbol{\alpha = 0.05}$		$\boldsymbol{\alpha = 0.1}$	
$\boldsymbol{\sigma_1^2}$		Tukey	Algorithm	Tukey	Algorithm
	$\boldsymbol{n}_1$	0.0564	0.0544	0.0968	0.0972
	$\boldsymbol{n}_2$	0.0448	0.0432	0.0972	0.1012
	$\boldsymbol{n}_3$	0.0584	0.0512	0.0892	0.0864
	$\boldsymbol{n}_4$	0.0488	0.0524	0.1016	0.1040
$\boldsymbol{\sigma_2^2}$		Tukey	Algorithm	Tukey	Algorithm
	$\boldsymbol{n}_1$	0.0564	0.0444	0.1028	0.0900
	$\boldsymbol{n}_2$	0.0564	0.0508	0.1008	0.0948
	$\boldsymbol{n}_3$	0.0332	0.0496	0.0616	0.0936
	$\boldsymbol{n}_4$	0.0240	0.0584	0.0428	0.0860
$\boldsymbol{\sigma_3^2}$		Tukey	Algorithm	Tukey	Algorithm
	$\boldsymbol{n}_1$	0.0476	0.0412	0.0944	0.0880
	$\boldsymbol{n}_2$	0.0544	0.0476	0.0996	0.0972
	$\boldsymbol{n}_3$	0.0680	0.0484	0.1220	0.0916
	$\boldsymbol{n}_4$	0.0792	0.0472	0.1512	0.1088
$\boldsymbol{\sigma_4^2}$		Tukey	Algorithm	Tukey	Algorithm
	$\boldsymbol{n}_1$	0.0580	0.0480	0.1008	0.0952
	$\boldsymbol{n}_2$	0.0564	0.0476	0.1008	0.0984
	$\boldsymbol{n}_3$	0.0340	0.0536	0.0732	0.0976
	$\boldsymbol{n}_4$	0.0224	0.0472	0.0520	0.1004
$\boldsymbol{\sigma_5^2}$		Tukey	Algorithm	Tukey	Algorithm
	$\boldsymbol{n}_1$	0.0524	0.0496	0.1004	0.0980
	$\boldsymbol{n}_2$	0.0496	0.0536	0.0980	0.0968
	$\boldsymbol{n}_3$	0.0448	0.0520	0.0800	0.0940
	$\boldsymbol{n}_4$	0.0292	0.0444	0.0656	0.0956
$\boldsymbol{\sigma_6^2}$		Tukey	Algorithm	Tukey	Algorithm
	$\boldsymbol{n}_1$	0.0736	0.0544	0.1100	0.0912
	$\boldsymbol{n}_2$	0.0612	0.0508	0.1084	0.0956
	$\boldsymbol{n}_3$	0.0276	0.0480	0.0616	0.0968
	$\boldsymbol{n}_4$	0.0180	0.0508	0.0368	0.1000

Table 3.6: Results of Simulations For Testing Multiple Comparisons for Levels of AB.

Numbers in the table are simulated p-values. We consider four different sizes and six different variance vectors as shown in Section 3.4.3, with the two different α levels shown.

		$\boldsymbol{\alpha = 0.05}$		$\boldsymbol{\alpha = 0.1}$	
$\boldsymbol{\sigma_1^2}$		Tukey	Algorithm	Tukey	Algorithm
	$\boldsymbol{n_1}$	0.0476	0.0368	0.0972	0.0888
	$\boldsymbol{n_2}$	0.0524	0.0452	0.0940	0.0932
	$\boldsymbol{n_3}$	0.0456	0.0432	0.1008	0.0784
	$\boldsymbol{n_4}$	0.0452	0.0444	0.1092	0.1044
$\boldsymbol{\sigma_2^2}$		Tukey	Algorithm	Tukey	Algorithm
	$\boldsymbol{n_1}$	0.0768	0.0420	0.1376	0.0920
	$\boldsymbol{n_2}$	0.0892	0.0500	0.1472	0.1020
	$\boldsymbol{n_3}$	0.0552	0.0408	0.0988	0.0888
	$\boldsymbol{n_4}$	0.0416	0.0508	0.0732	0.0944
$\boldsymbol{\sigma_3^2}$		Tukey	Algorithm	Tukey	Algorithm
	$\boldsymbol{n_1}$	0.0456	0.0372	0.1012	0.0900
	$\boldsymbol{n_2}$	0.0592	0.0504	0.1004	0.0920
	$\boldsymbol{n_3}$	0.0696	0.0376	0.1308	0.0804
	$\boldsymbol{n_4}$	0.1028	0.0420	0.1624	0.0900
$\boldsymbol{\sigma_4^2}$		Tukey	Algorithm	Tukey	Algorithm
	$\boldsymbol{n_1}$	0.0792	0.0460	0.1192	0.0936
	$\boldsymbol{n_2}$	0.0724	0.0456	0.1228	0.0820
	$\boldsymbol{n_3}$	0.0484	0.0388	0.0852	0.0896
	$\boldsymbol{n_4}$	0.0236	0.0488	0.0540	0.0936
$\boldsymbol{\sigma_5^2}$		Tukey	Algorithm	Tukey	Algorithm
	$\boldsymbol{n_1}$	0.0660	0.0468	0.1232	0.0980
	$\boldsymbol{n_2}$	0.0736	0.0560	0.1052	0.0952
	$\boldsymbol{n_3}$	0.0516	0.0456	0.0844	0.0796
	$\boldsymbol{n_4}$	0.0368	0.0500	0.0752	0.0976
$\boldsymbol{\sigma_6^2}$		Tukey	Algorithm	Tukey	Algorithm
	$\boldsymbol{n_1}$	0.1148	0.0400	0.1500	0.0856
	$\boldsymbol{n_2}$	0.0936	0.0464	0.1680	0.0996
	$\boldsymbol{n_3}$	0.0572	0.0384	0.0852	0.1012
	$\boldsymbol{n_4}$	0.0368	0.0448	0.0564	0.0920

3.5 Data Analysis Example

An example dataset is shown that considers the effect on potato plants of three factors with two levels each: variety of potato (1 or 2); cold acclimatization regimes (0 for room temperature or 1 for cold room); exposure to cold temperatures (-4 degrees C coded as 1; -8 degrees C coded as 2). The response of interest is ion leakage (a measure of damage to the plant). The data are unbalanced, as may be encountered if some plants were lost during an experiment to measure effects of this nature.

The data were analyzed using the `lm` function in R and PB Algorithms 1, 2, 4 and 6. Fitted-residual plots were examined for the biggest model, which includes all main effects, two-way interaction terms, and the three-way interaction term. The data were found to violate the equal variance assumption (see Figure 3.2). We also note that the data are unbalanced; see Table 3.7. A quantile-quantile plot was also examined for the residuals, and the normality assumption appeared to be violated. A square root transformation was used after adding a constant to the response variable to ensure positive values and the biggest model fit again. After transformation, the data no longer appeared to violate the equal variance or normality assumption. The fitted-residual plots before and after transformation are shown in Figure 3.2.

Variety	Regime	Temp	N	Leak ($\bar{y}_{ijk}$)	Std. Dev.
1	0	1	5	3.87	1.61
1	0	2	5	5.93	6.01
1	1	1	12	2.34	2.73
1	1	2	13	10.98	7.74
2	0	1	13	22.38	12.82
2	0	2	13	32.32	12.97
2	1	1	7	2.42	1.66
2	1	2	7	9.81	3.82

Table 3.7: Summary Statistics, Potato Data.

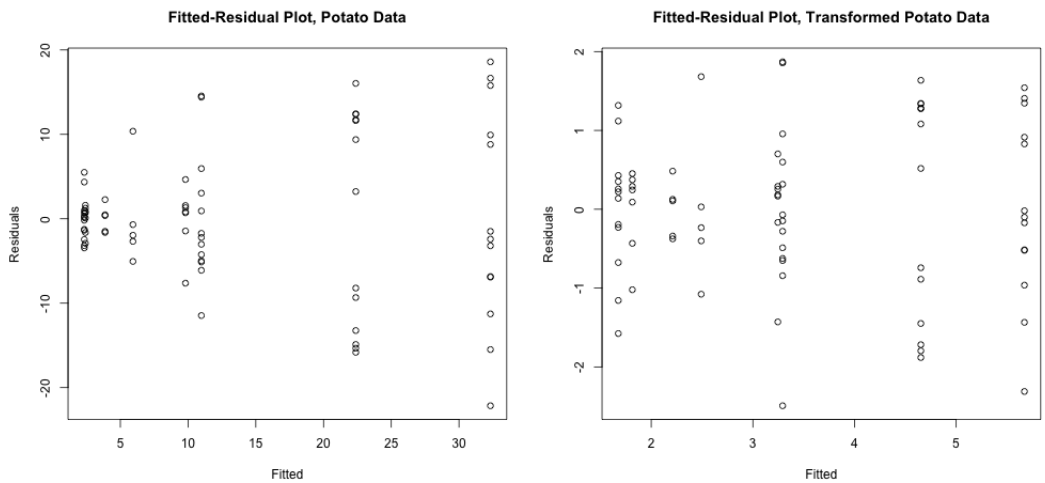

Figure 3.2: Fitted-Residual Plots, Potato Data.

The p-value for the three-way interaction term on the transformed data was 0.383, indicating no significant effect. Algorithm 1 was applied to the untransformed data and obtained a p-value of 0.163 for the three-way interaction term, also indicating no significant effect. The model was re-fit using the `lm` function in R, and the only two-way interaction term found to have significant effect was the variety:regime term with p-value near 0. Algorithm 2 was applied to the untransformed data three times (once for each two-way interaction term). Similarly, the only significant two-way term was variety:regime with p-value of 0. Finally, the model was fit again with only the main effects and the variety:regime interaction term; using the `lm` function found the main effect temp to be significant with p-value near 0. Algorithm 4 was applied to the untransformed data to test the main effect (temp) not involved with the interaction term and obtained the p-value 0.002, also indicating that the temp term should remain in the model.

MCP were applied to the model with Tukey's test and Algorithm 6; both found a significant difference, with p-values near 0, between the same levels of the variety:regime interaction ($\bar{y}_{10\cdot}$ and $\bar{y}_{20\cdot}$; $\bar{y}_{20\cdot}$ and $\bar{y}_{11\cdot}$; $\bar{y}_{20\cdot}$ and $\bar{y}_{21\cdot}$, recalling that levels

for variety were 1 and 2 and levels of regime were 0 and 1).

This analysis shows that the PB method applied to the untransformed data obtains the same conclusions as performing F-tests and Tukey's test on transformed data. Thus, the PB method avoids the need for transformation and simplifies interpretation of the results, as point estimates and standard errors for the differences between levels are on the original scale. Additionally, with sufficient sample size, the group means are approximately normal regardless of the distribution of the original observations by the Central Limit Theorem (Casella and Berger, 2002). In this example, despite smaller group sample sizes, the PB method appears to be robust to violation of the normality assumption.

3.6 Discussion

As shown in Table 3.1, under equal variances and equal sample sizes, the F-test and the algorithm perform similarly, with overall p-values near the nominal level. In particular, with equal variances for all groups ($\boldsymbol{\sigma_1^2}$), both tests are near the nominal level for all simulated sample sizes. However, for the other simulated (unequal) variances, the F-test begins to over-reject or under-reject the null hypothesis for those sample sizes with unbalanced data ($\boldsymbol{n_3}$ and $\boldsymbol{n_4}$).

For $\boldsymbol{\sigma_2^2}$, $\boldsymbol{\sigma_4^2}$, $\boldsymbol{\sigma_5^2}$, and $\boldsymbol{\sigma_6^2}$, the F-test rejects the null hypothesis less often than would be expected when we have unbalanced data, indicating the F-statistic is artificially small due to the pooled variance estimate being artificially large. This is particularly true for $\boldsymbol{n_4}$, where the largest group size has the largest variance (recall that calculating an estimate of pooled variance involves weighting each sample variance by the sample sizes of the respective groups). On the other hand, for $\boldsymbol{\sigma_3^2}$, the F-test rejects the null hypothesis more often than expected when we also have (pronounced) unbalanced data. In this case, the F-statistic is artificially large due to

the pooled variance estimate being artificially small. This is true for $\boldsymbol{n_4}$, where the largest group size has a smaller variance. This is not as pronounced for $\boldsymbol{n_3}$; although this group has unequal sample sizes, they are closer together than for $\boldsymbol{n_4}$. These trends appear to be true both for $\alpha = 0.05$ and $\alpha = 0.1$. In these cases where the F-test is either too conservative or too liberal, the algorithm still appears to give satisfactory results, rejecting the null hypothesis approximately at the nominal level α.

The simulation results for testing the BC (and ABC) interaction were similar to those for testing the ABC interaction. Again, under equal variances and equal sample sizes, the F-test and the algorithm perform similarly, with overall p-values near the nominal level. However, for the other simulated (unequal) variances, the F-test begins to over-reject or under-reject the null hypothesis for those sample sizes with unbalanced data ($\boldsymbol{n_3}$ and $\boldsymbol{n_4}$). For $\boldsymbol{\sigma_2^2}$, $\boldsymbol{\sigma_4^2}$, $\boldsymbol{\sigma_5^2}$, and $\boldsymbol{\sigma_6^2}$, the F-test rejects the null hypothesis less often than would be expected when we have unbalanced data. Also similar to the results for testing the three-way interaction, for $\boldsymbol{\sigma_3^2}$, the F-test rejects the null hypothesis more often than expected when we also have (pronounced) unbalanced data, both for $\alpha = 0.05$ and $\alpha = 0.1$. Again, in all simulated cases, the proportion rejected using the algorithm was fairly close to the nominal level. The results shown in Table 3.3 and Table 3.4, with the F-test rejecting more or less often than the nominal level in cases with both unequal variances and unbalanced data, while the algorithm performs satisfactorily in each case, are similar to results of the simulations shown in Tables 3.1 and 3.2 . While we illustrated this method with a three-way ANOVA model, the test statistic $\tilde{S}_I$ for each term in the model takes on the same form. Thus, models with additional factors should follow the same pattern, though interpretation becomes more complicated with additional factors.

Table 3.5 shows the results for comparing our MCP PB method (Algorithm 5 – pairwise comparisons of the levels of factor A) to Tukey's test. As with the other

simulations, the simulated p-values are near the specified α level for both methods when we have homoscedasticity and balanced data (σ_1^2 and n_1 or n_2). However, in cases with both unequal variances and unbalanced data, the simulated p-values for the algorithm are generally near the specified α level, whereas those for Tukey's test tend to be too conservative. An exception to this is with σ_3^2 and n_4, where Tukey's test rejected H_0 more often than the nominal level. Similarly to our comparisons between Algorithms 1–4 and the F-test, for σ_3^2 and n_4, smaller variances correspond to larger sample sizes, so the pooled variance estimate used for Tukey's test becomes artificially small, and thus the test statistic artificially large. Table 3.6 shows the results for comparing our MCP PB method (Algorithm 6 – pairwise comparisons of the levels of the AB interaction term) to the analogous version of Tukey's test. These results are very similar to the results shown in Table 3.5, multiple comparisons of the levels of A.

In this research, we looked at the multi-factor heteANOVA problem with unbalanced data, including MCP's analogous to Tukey's test from a parametric bootstrap view and proposed applicable PB tests. Simulation results show that traditional tests and the PB tests give acceptable results under the equal variance assumption. Additionally, when data are balanced, the classical F-tests and MCP's perform satisfactorily in most heteroscedastic cases. However, for heteANOVA problems when the equal variance assumption is violated and data are unbalanced, the traditional tests no longer provide reasonable nominal levels, while the proposed PB methods work well and are easy to implement.

A potential limitation of the proposed PB method is that it may still require the normality assumption when sample sizes are small, so if a particular dataset violates both the normality and homoscedasticity assumptions, a transformation may still be needed. However, according to the Central Limit Theorem (Casella and Berger, 2002), with large sample sizes, the groups means are approximately normal regardless

of the original distribution. In the potato data example, sample sizes were fairly small and the PB test appeared robust to violation of the normality assumption. Additionally, as discussed in the introduction and in Christensen (2016), we may need to exercise caution when making practical decisions based on differences in means between groups with unequal variances, carefully considering implications for the practical issue being studied. In this study, we only examined two levels for each factor in our simulations for Algorithms 1–4, for simplicity, so further simulations with additional levels may be warranted. Despite these limitations, the proposed PB tests provide viable methods for dealing with multi-factor heteANOVA problems and MCP. Further areas for research may include extending the procedures to more complicated models, such as additional factors/levels, models that include random effects, or more complex designed experiments.

Chapter 4

PB Analogy to Dunnett's Test

4.1 Introduction

Consider a one-way analysis of variance (ANOVA) problem with a treatment groups, where the first group is a control group. Let Y_{ij} be the value of the response variable in the jth trial for the ith factor level, $\mu + \alpha_i$ the mean for the ith factor level, $i = 1, 2, \cdots, a, j = 1, 2, \cdots, n_i$. The one-way ANOVA model is as follows:

$$Y_{ij} = \mu + \alpha_i + \varepsilon_{ij}, \tag{4.1}$$

where $\varepsilon_{ij} \overset{iid}{\sim} N(0, \sigma_i^2)$, and $\sum_i \alpha_i = 0$.

One may wish to perform multiple comparisons of the treatment groups with the control group, rather than performing all pairwise comparisons. In this case, procedures such as Tukey's test, which examines all pairwise comparisons among the treatment groups, can result in confidence intervals that are wider than necessary (Dunnett, 1955). Under the equal variance assumption, Dunnett's test (Dunnett, 1955; Kutner et al., 2005), which is similar to Tukey's test, can be used for comparing treatment groups only against the control, and is frequently used in clinical

or pharmacological studies (Tallarida and Murray, 1987; Strojek et al., 2011; Kutuk et al., 2019; Cheng et al., 1990). Dunnett's test compares $a-1$ pairs (each group with the control group), instead of the $\binom{a}{2}$ pairs involved in all pairwise comparisons. Dunnett's test uses the statistic

$$\frac{|\bar{Y}_1 - \bar{Y}_i|}{\sqrt{\hat{\sigma}^2(1/n_1 + 1/n_i)}} \tag{4.2}$$

where $\bar{Y}_i$ is the sample mean for group i, $i \neq 1$, with α_1 the parameter associated with the control group, and $\hat{\sigma}^2$ is the pooled variance estimate $\frac{\sum_{i=1}^{a}\sum_{n=1}^{n_i}(Y_{ij}-\bar{Y}_i)^2}{\sum_{i=1}^{a} n_i - a}$.

When variances are equal, so the pooled variance estimate is appropriate for all treatment groups, and data are balanced, the test statistic follows a special case of a multivariate analog of the t-distribution (Miller, 1981), the density of which was derived by Dunnett and Sobel (1954, 1955). Tables of critical values for this distribution in various practical scenarios were developed by Dunnett (1955) and are also available in software, e.g. the `DunnettTest` function in the R package `DescTools` (Signorell et mult al, 2020).

When the assumption of equal variance is violated, we can modify the test statistic to include the separate variance estimates as shown in the next section. However, the modified test statistic no longer follows a known distribution. When the variances are unequal and the data are also unbalanced (hereafter called HeteANOVA problem), the results of Dunnett's test are questionable. Many alternative methods were developed for the classical F-test and multiple comparisons for HeteANOVA problems (Krishnamoorthy and Lu, 2007; Zhang, 2015a; Xu et al., 2013). Among them, the parametric bootstrap (Krishnamoorthy and Lu, 2007) test is shown to be one of the best for testing equality of factor level means. Recently, Zhang (2015a,b) proposed PB multiple comparison tests for one-way and two-way ANOVA, which are shown to be competitive.

Inspired by Dunnett's test and PB tests, we develop a PB test that is similar to

the PB multiple comparison procedures described in Zhang (2015a,b). This modified PB test is analogous to Dunnett's test, which performs simultaneous multiple comparisons of the treatment groups with the control for the heteANOVA problem. This chapter is organized as follows: Section 4.2 proposes the methodology and presents the algorithm; Section 4.3 performs a simulation study; Section 4.4 gives two real examples; and Section 4.5 gives conclusions and discussion of the research.

4.2 Proposed PB Test and Algorithm

In this section, we develop a PB method for multiple comparisons of treatment groups with the control group for a heteANOVA problem, and present an algorithm to implement the test.

4.2.1 Proposed PB Test

Following the procedure from previous papers (Krishnamoorthy and Lu, 2007; Zhang, 2015a), consider the test statistic in equation 4.2. We modify this to include the different group variances:

$$T_i = \frac{|\bar{Y}_1 - \bar{Y}_i|}{\sqrt{(s_1^2/n_1 + s_i^2/n_i)}} \tag{4.3}$$

As noted previously, this test statistic no longer follows a known distribution for comparison; the aim of the PB method is to simulate this distribution. The test is location invariant, so we assume without loss of generality that the mean of $\bar{Y}_i$ is zero for all i. Then $\bar{Y}_i \sim N(0, \sigma_i^2/n_i)$ and the sample variance $S_i^2 \sim \frac{\sigma_i^2}{n-1}\chi^2_{(n_i-1)}$ (Casella and Berger, 2002). These can be approximately simulated by pivot variables $\bar{Y}_{Bi} \sim N(0, s_i^2/n_i)$, or equivalently, $\bar{Y}_{Bi} \sim N(0,1)\sqrt{s_i^2/n_i}$, and $S_{Bi}^2 \sim \frac{s_i^2}{n-1}\chi^2_{(n_i-1)}$.

We can replace $\bar{Y}_i$ and s_i^2 in equation 4.3 with $\bar{Y}_{Bi}$ and S_{Bi}^2 to obtain a PB pivot variable:

$$T_{PB_i} = \frac{|\bar{Y}_{B1} - \bar{Y}_{Bi}|}{\sqrt{(S_{B1}^2/n_1 + S_{Bi}^2/n_i)}} \tag{4.4}$$

As described in Christensen (1996), Dunnett's test is based on knowing the distribution of the maximum over i of the test statistic in equation 4.2 when the null hypothesis of equality of all means is true; i.e. if H_0 is not rejected for the maximum difference, it would not be rejected for any of the differences. For the PB method, we simulate a distribution for the test statistic 4.3, using 4.4. With this simulated distribution, we can estimate the p-value or obtain a critical value which can be used to construct confidence intervals. The procedure is shown in the following algorithm, and example code for this algorithm is shown in Appendix C.

4.2.2 PB Algorithm for Comparing Multiple Treatment Groups with Control

Algorithm 7

For a given $(n_1, n_2, ..., n_a)$, $(\bar{y}_1, \bar{y}_2, ..., \bar{y}_a)$, and $(s_1^2, s_2^2, ..., s_a^2)$, compute the test statistic T_i in equation 4.3 for each group paired with the control group.

For $l = 1, ..., L$:

Generate $\bar{Y}_{Bi} \sim N(0,1)\sqrt{s_i^2/n_i}$, and $S_{Bi}^2 \sim \frac{s_i^2}{n-1}\chi^2_{(n_i-1)}$, $i = 1, ..., a$.

For each $i \neq 1$, compute the PB pivot variable T_{PB_i} as in equation 4.4.

D_l = maximum over i of the results from the previous step.

(end loop).

$\boldsymbol{D} = (D_1, ..., D_L)$ is then a simulated distribution for the test statistic. One can use the $1 - \alpha$ quantile of $\boldsymbol{D}$, D_{crit}, as a critical value for a decision rule (i.e. reject $H_0 : \alpha_1 = \alpha_i$ in favor of $H_A : \alpha_1 \neq \alpha_i$ for some i, if the test statistic 4.3 is larger than D_{crit}) or construct a confidence interval using this critical value:

$$\bar{Y}_i - \bar{Y}_1 \pm D_{crit}\sqrt{(s_i^2/n_i + s_1^2/n_1)}.$$

As usual, if a p-value is desired, one can compute the proportion of values of D that are greater than the test statistic in 4.3.

The code shown in Appendix C is one way to program the PB test (Algorithm 7) to simulate a distribution for the PB test statistic. The output here is the test statistic and the p-value, but could be modified to return other values, for example, D_{crit} or confidence intervals. Processing time was checked for L=100000 for select scenarios ($\boldsymbol{n}_1$ and $\boldsymbol{n}_2$ with $\boldsymbol{\sigma}_1^{\boldsymbol{2}}$ and $\boldsymbol{n}_4$ with $\boldsymbol{\sigma}_1^{\boldsymbol{2}}$ and $\boldsymbol{\sigma}_5^{\boldsymbol{2}}$ from the simulation studies). For one simulated dataset with these scenarios, the maximum processing time was 2.559 seconds. With L=5000, the maximum processing time for one of these datasets was 0.164 seconds.

4.3 Simulations

4.3.1 Evaluation of Type I Error

To evaluate the performance of the algorithm in terms of Type I error, we simulated 2500 datasets with $\mu = 0$ and $\alpha_i = 0$ for all i, such that H_0 is true, and compared the rejection rate for both Dunnett's Test, using the `DunnettTest` function in the R package `DescTools` Signorell et mult al (2020) and the PB method (Algorithm 7) with $L = 5000$ bootstrap sample mean and variance vectors. We used $a = 6$ treatment groups including the control, with $\boldsymbol{\sigma}_1^{\boldsymbol{2}} = (1, 1, 1, 1, 1, 1)$, $\boldsymbol{\sigma}_2^{\boldsymbol{2}} =$

$(0.1, 0.1, 0.1, 0.5, 0.5, 0.5)$, $\boldsymbol{\sigma_3^2} = (1, 1, 1, 0.5, 0.5, 0.5)$, $\boldsymbol{\sigma_4^2} = (0.1, 0.2, 0.3, 0.4, 0.5, 1)$, $\boldsymbol{\sigma_5^2} = (0.3, 0.9, 0.4, 0.7, 0.5, 1)$, and $\boldsymbol{\sigma_6^2} = (0.01, 0.1, 0.1, 0.1, 0.1, 1)$. The sample size vectors used in the simulations were $\boldsymbol{n}_1 = (5, 5, 5, 5, 5, 5)$, $\boldsymbol{n}_2 = (10, 10, 10, 10, 10, 10)$, $\boldsymbol{n}_3 = (3, 3, 4, 5, 6, 6)$, and $\boldsymbol{n}_4 = (4, 6, 8, 12, 16, 20)$. The simulation settings follow from Zhang (2015b). All calculations, simulations and data analysis were performed using R (R Core Team, 2021).

Results are shown in Table 4.1. With the equal variance assumption, both Dunnett's test and the PB test give acceptable results. Additionally, when data are balanced, Dunnett's test performs satisfactorily in most heteroscedastic cases. The exception to this is for $\boldsymbol{\sigma_6^2}$. In this case, the simulated p-value for Dunnett's test is higher than the nominal level even with balanced data. This variance vector includes 0.01 which is small, likely leading to an artificially small pooled variance estimate and thus an artificially large test statistic, so the test rejects more often than the nominal level.

The PB test outperforms Dunnett's test, with simulated p-values close to the nominal level for all simulation settings including unequal variance and unbalanced data. In all heteroscedastic cases except $\boldsymbol{\sigma_3^2}$, the proportion rejected for Dunnett's test is too conservative (less than the nominal level) when the data are unbalanced. In these cases, the smaller variances in the simulations are for groups with smaller sample sizes, and larger variances for groups with larger sample sizes. For these settings, the pooled variance is artificially large, leading to a test statistic that is artificially small. The opposite is true for $\boldsymbol{\sigma_3^2}$, which assigns smaller variances to larger group sizes, so the pooled variance estimate is too small and the test statistic too large.

Table 4.1: Simulation Results: MCP of Treatment Group Means vs. Control – Type I Error.

Numbers in the table are simulated p-values. We consider four different sample sizes and six different variance vectors as shown in Section 4.3.1, with the two different α levels shown.

			$\boldsymbol{\alpha = 0.05}$		$\boldsymbol{\alpha = 0.1}$
$\boldsymbol{\sigma_1^2}$		Dunnett	PB	Dunnett	PB
	$\boldsymbol{n_1}$	0.0516	0.0420	0.1044	0.0824
	$\boldsymbol{n_2}$	0.0464	0.0360	0.1080	0.0980
	$\boldsymbol{n_3}$	0.0448	0.0384	0.1052	0.0876
	$\boldsymbol{n_4}$	0.0584	0.0592	0.0996	0.1052
$\boldsymbol{\sigma_2^2}$		Dunnett	PB	Dunnett	PB
	$\boldsymbol{n_1}$	0.0464	0.0444	0.0940	0.0872
	$\boldsymbol{n_2}$	0.0420	0.0460	0.0868	0.1056
	$\boldsymbol{n_3}$	0.0100	0.0364	0.0328	0.0836
	$\boldsymbol{n_4}$	0.0016	0.0540	0.0080	0.0996
$\boldsymbol{\sigma_3^2}$		Dunnett	PB	Dunnett	PB
	$\boldsymbol{n_1}$	0.0704	0.0404	0.1272	0.0788
	$\boldsymbol{n_2}$	0.0752	0.0364	0.1368	0.0976
	$\boldsymbol{n_3}$	0.1000	0.0412	0.1836	0.0956
	$\boldsymbol{n_4}$	0.1408	0.0592	0.2064	0.1068
$\boldsymbol{\sigma_4^2}$		Dunnett	PB	Dunnett	PB
	$\boldsymbol{n_1}$	0.0456	0.0472	0.0780	0.0928
	$\boldsymbol{n_2}$	0.0424	0.0424	0.0744	0.1092
	$\boldsymbol{n_3}$	0.0104	0.0376	0.0336	0.0844
	$\boldsymbol{n_4}$	0.0004	0.0496	0.0020	0.0944
$\boldsymbol{\sigma_5^2}$		Dunnett	PB	Dunnett	PB
	$\boldsymbol{n_1}$	0.0348	0.0436	0.0768	0.0892
	$\boldsymbol{n_2}$	0.0320	0.0412	0.0692	0.1056
	$\boldsymbol{n_3}$	0.0184	0.0372	0.0604	0.0872
	$\boldsymbol{n_4}$	0.0096	0.0556	0.0308	0.1016
$\boldsymbol{\sigma_6^2}$		Dunnett	PB	Dunnett	PB
	$\boldsymbol{n_1}$	0.0996	0.0540	0.1392	0.1060
	$\boldsymbol{n_2}$	0.1000	0.0416	0.1440	0.1044
	$\boldsymbol{n_3}$	0.0308	0.0412	0.0628	0.1056
	$\boldsymbol{n_4}$	0.0016	0.0460	0.0036	0.0884

4.3.2 Evaluation of Power

To evaluate the performance of the algorithm in terms of power, we simulated 2500 datasets for each combination of settings with $\mu = 0$ and
$\boldsymbol{\alpha}_1 = (0, 0, -0.2, 0.2, 0.4, 0.8)$ or $\boldsymbol{\alpha}_2 = (0, 0, -0.3, 0.3, 0.6, 1.2)$, such that H_0 is not true, and compared the rejection rate for both Dunnett's Test, using the `DunnettTest` function in the R package `DescTools` (Signorell et mult al, 2020) and the PB method (Algorithm 7) with $L = 5000$ bootstrap sample mean and variance vectors. We used $a = 6$ treatment groups including the control, with $\boldsymbol{\sigma}_1^{\mathbf{2}} = (1, 1, 1, 1, 1, 1)$, $\boldsymbol{\sigma}_2^{\mathbf{2}} = (0.1, 0.2, 0.3, 0.4, 0.5, 1)$, $\boldsymbol{\sigma}_3^{\mathbf{2}} = (0.3, 0.9, 0.4, 0.7, 0.5, 0.1)$. The sample size vectors used in the simulations were $\boldsymbol{n}_1 = (15, 15, 15, 15, 15, 15)$, $\boldsymbol{n}_2 = (15, 15, 20, 20, 25, 25)$, $\boldsymbol{n}_3 = (15, 18, 21, 24, 27, 30)$. The simulation settings follow from Xu et al. (2013). All calculations, simulations and data analysis were performed using R (R Core Team, 2021).

Results are shown in Table 4.2. With equal variance and balanced data ($\boldsymbol{\sigma}_1^{\mathbf{2}}$ and $\boldsymbol{n}_1$), power was similar between the two methods or somewhat lower for the PB version. However, with unequal variance and unbalanced data, power is similar between the two methods or somewhat higher for the PB version. As expected, the power for both tests is generally higher with the mean vector $\boldsymbol{\alpha}_2$ as this has a larger difference between groups, and with the sample size vectors $\boldsymbol{n}_2$ and $\boldsymbol{n}_3$, as these are larger sample sizes for most groups. Note that with $\boldsymbol{\sigma}_2^{\mathbf{2}}$ and both $\boldsymbol{n}_2$ and $\boldsymbol{n}_3$, the largest group has the largest variance. Thus, we would expect the pooled variance estimate to be too large, making Dunnett's test statistic too small, and the test thus being too conservative. We do see somewhat lower power for Dunnett's test in Table 4.2 with these settings, though it is still acceptable. The opposite would be expected for $\boldsymbol{\sigma}_3^{\mathbf{2}}$, where the largest group size has the smallest variance, and some increase in power for Dunnett's test with $\boldsymbol{\sigma}_3^{\mathbf{2}}$ over $\boldsymbol{\sigma}_2^{\mathbf{2}}$ is noted in Table 4.2.

Table 4.2: Simulation Results: MCP of Treatment Group Means vs. Control – Power.

Numbers in the table are simulated power, with three different sample sizes, three different variance vectors, and two different mean vectors as shown in Section 4.3.2.

		$\boldsymbol{\alpha}_1$		$\boldsymbol{\alpha}_2$	
$\boldsymbol{\sigma_1^2}$		Dunnett	PB	Dunnett	PB
	$\boldsymbol{n}_1$	0.4136	0.3668	0.8056	0.7560
	$\boldsymbol{n}_2$	0.5164	0.4532	0.9104	0.8740
	$\boldsymbol{n}_3$	0.5456	0.4960	0.9332	0.9016
$\boldsymbol{\sigma_2^2}$		Dunnett	PB	Dunnett	PB
	$\boldsymbol{n}_1$	0.8036	0.7008	0.9916	0.9736
	$\boldsymbol{n}_2$	0.8760	0.8928	0.9992	1.0000
	$\boldsymbol{n}_3$	0.9072	0.9468	0.9996	0.9996
$\boldsymbol{\sigma_3^2}$		Dunnett	PB	Dunnett	PB
	$\boldsymbol{n}_1$	0.8356	0.9872	0.9988	0.9996
	$\boldsymbol{n}_2$	0.9504	0.9904	1.0000	1.0000
	$\boldsymbol{n}_3$	0.9656	0.9948	1.0000	1.0000

4.4 Applications

4.4.1 Iron Data

An example of the method is shown by applying it to the data discussed by Sananman and Lear (1961) (data downloaded from website by Winner, University of Florida Winner (2020)). The data concerns iron content, in milligrams per liter, found in various depths of seawater. For this example, we considered surface water, where Depth=0, to be the control group. Summary statistics are shown in Table 4.3. We use five digits for the display as some of the sample variances are quite small. As shown in the table, the variance for 40 feet is somewhat larger than the others, and the variances at the shallower levels are somewhat smaller than those for the deeper

levels.

Table 4.3: Summary Statistics for Iron Data.

Depth	$\bar{y}_i$	s_i^2	s_i	n_i
0	0.04267	0.00001	0.00252	3
10	0.03967	0.00006	0.00757	3
30	0.04533	0.00001	0.00231	3
40	0.10867	0.00169	0.04105	3
50	0.10333	0.00020	0.01401	3
100	0.20520	0.00052	0.02282	5

We fit the one-way ANOVA model and then checked assumptions of normality and constant variance. By the Shapiro-Wilk test for normality using the `shapiro.test` function in R (W = 0.9394, p-value = 0.2334), and examination of a normal plot of the standardized residuals (residual plots shown below), the normality assumption was satisfied. For checking the constant variance assumption, we examined a plot of the standardized residuals against the fitted values from the ANOVA model (Figure 4.1, right panel). We also performed the Breusch-Pagan test using the function `bptest` from the R package `lmtest` (Zeileis and Hothorn, 2002). The p-value from the Breusch-Pagan test was 0.0596, between the commonly used alpha levels of 0.05 and 0.1, and the residual-fitted plot did appear to indicate non-constant variances.

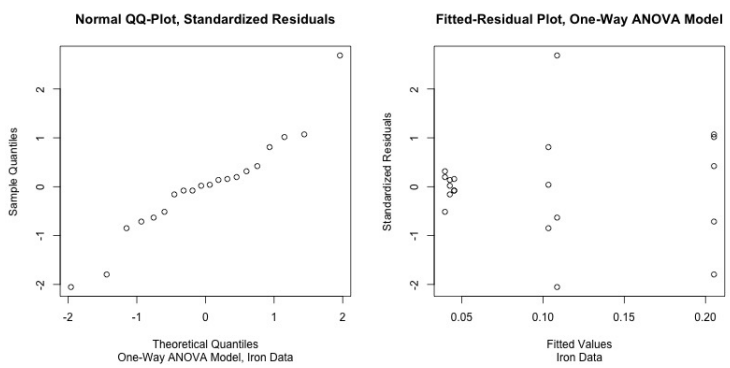

Figure 4.1: Verification of Assumptions, Iron Data.

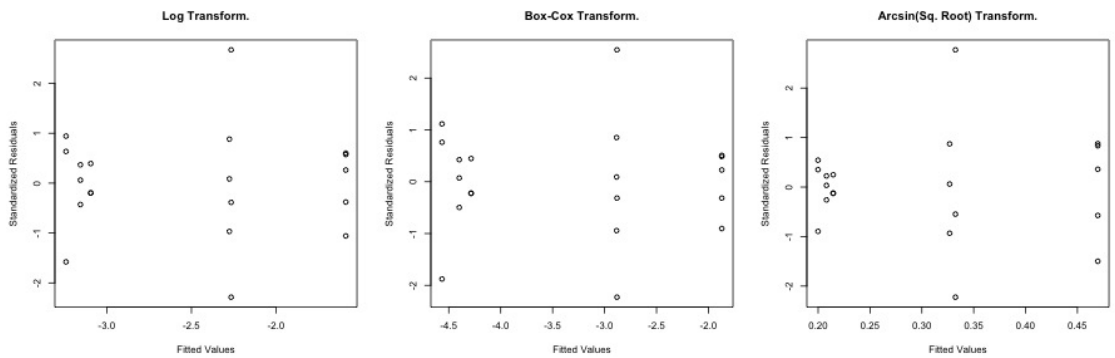

Figure 4.2: Fitted-Residual Plots after Transformations, Iron Data.

Several transformations were attempted to satisfy the non-constant variance assumption: log transformation; Box-Cox transformation using $\lambda = -0.2$; and since the units of measurement mg/L could be considered a proportion, the $\sin^{-1}\sqrt{y_{ij}}$ transformation. The λ value for the Box-Cox transformation was found using the `boxcox` function from the R package `MASS` (Venables and Ripley, 2002). While the log transformation and the Box-Cox transformation improved the appearance of the fitted-residual plots (shown in Figure 4.2), none of these improved the p-value from the Breusch-Pagan test.

We performed Dunnett's test, using the previously mentioned function in R, on both the untransformed data and the Box-Cox transformed data. Of note, the normality assumption was still satisfied after the Box-Cox transformation, with W = 0.9554 and p-value = 0.4556 according to the Shapiro-Wilk test. The Dunnett's tests found a significant difference between the iron content of water from the surface (treated as control) and all depths of 40 feet or greater. We then performed the analogous PB test. This test only found a significant difference between the surface and depths of 50 feet or greater. The differences between means, confidence intervals and p-values are shown in Tables 4.4 and 4.5 for Dunnett's test and Table 4.6 for the PB test.

Table 4.4: Results from Dunnett's Test, Iron Data.

	Diff	Lower CI	Upper CI	p-value
10-0	-0.0030	-0.0508	0.0448	0.9998
30-0	0.0027	-0.0451	0.0504	0.9999
40-0	0.0660	0.0182	0.1138	0.0065
50-0	0.0607	0.0129	0.1084	0.0118
100-0	0.1625	0.1198	0.2053	0.0000

Table 4.5: Results from Dunnett's Test, Box-Cox Iron Data.

	Diff	Lower CI	Upper CI	p-value
10-0	-0.1659	-0.8462	0.5143	0.9275
30-0	0.1140	-0.5663	0.7943	0.9835
40-0	1.5183	0.8380	2.1985	0.0001
50-0	1.5144	0.8341	2.1946	0.0004
100-0	2.5269	1.9185	3.1354	0.0000

Table 4.6: Results from PB Test, Iron Data.

	Diff	Lower CI	Upper CI	p-value
10-0	-0.0030	-0.0315	0.0255	0.9852
30-0	0.0027	-0.0095	0.0149	0.8072
40-0	0.0660	-0.0810	0.2130	0.3210
50-0	0.0607	0.0098	0.1115	0.0290
100-0	0.1625	0.0987	0.2263	0.0024

Recall from Table 4.3 that the measurements taken at 40 feet have a larger variance than the other depths. Thus, the pooled variance estimate could be too small for this group and lead to an artificially large test statistic in the traditional Dunnett's test. In fact, the mean squared error from the ANOVA model for the untransformed data is 0.0004 and the sample variance of the 40-foot depth measurements is 0.0017. A possible practical issue with these results is that if the goal was to get the most iron-rich water from as shallow depth as possible, knowing that the surface was not rich enough, obtaining the water from 40 feet deep could still yield samples that are not as high in iron as desired.

4.4.2 Elephant Ivory Data

An additional example of the method is shown by applying it to the data found in the supplementary material of Ziegler et al. (2016). The data concerns isotope levels in elephant tusks from different geographical areas. Summary statistics are shown in Table 4.7. We considered Asia to be the "control" group as the other regions were in Africa. While Ziegler et al. (2016) examined all pairwise comparisons of the different regions using the Games-Howell post-hoc test (Games and Howell, 1976), another possible question of interest could be whether any of the African regions differ from Asia (rather than additionally comparing all of the African regions with each other). The data are very unbalanced, and we can see from Table 4.7 and Figure 4.3 that the variances appear unequal for the δ^{15}N isotope ratio (nitrogen stable isotope ratios expressed in δ units). Ziegler et al. (2016) looked at several other isotopes and performed additional classification procedures, but we limited the analysis in this study to one isotope simply to illustrate the method.

Region	n_i	$\bar{Y}_i$	s_i^2
Asia	8	8.49	1.62
Central Africa	120	9.37	3.71
East Africa	37	9.78	6.15
Southern Africa	261	8.93	2.78
West Africa	69	5.85	1.40

Table 4.7: Summary Statistics, δ^{15}N, Elephant Tusk Data.

We fit the one-way ANOVA model and then checked assumptions of normality and constant variance. By the Shapiro-Wilk test for normality using the `shapiro.test` function in R (W = 0.985, p-value near 0), and examination of a normal plot of the residuals, the normality assumption was violated. The fitted-residual plot from the ANOVA model indicated violation of the equal variance assumption. We also performed the Breusch-Pagan (BP) test using the function `bptest` from the R package `lmtest` (Zeileis and Hothorn, 2002) and Levene's test using the `leveneTest` function

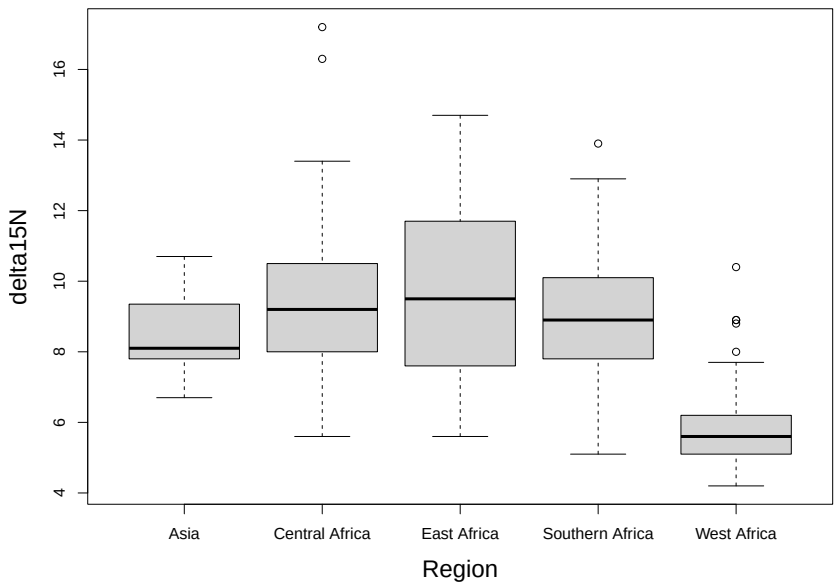

Figure 4.3: δ^{15}N by Region, Elephant Tusk Data.

from the R package `car` (Fox and Weisberg, 2019). The p-value from both formal tests for equal variance were near 0 and the fitted-residual plot indicated unequal group variances.

A log transformation was attempted to satisfy assumptions. The normality assumption was then satisfied by appearance of the normal plot and formally by the Shapiro-Wilk test, with W = 0.996 and p-value = 0.333. The fitted-residual plot was somewhat improved after transformation but still appeared to violate the equal variance assumption. The p-values for the BP test and Levene's test were 0.031 and 0.013, respectively, for the transformed data. The fitted-residual plots before and after transformation are shown in Figure 4.4.

We performed Dunnett's test, using the previously mentioned function in R, on the untransformed and the log transformed data; both found a significant difference in δ^{15}N isotope levels (nitrogen stable isotope ratios expressed in δ units) between

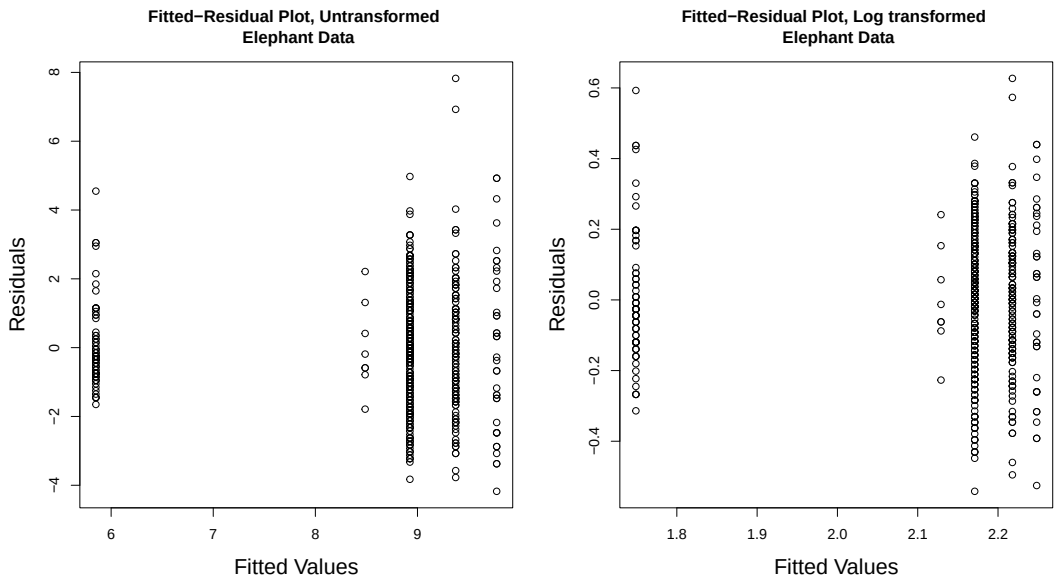

Figure 4.4: Fitted-Residual Plots Before/After Transformation, Elephant Data.

Asia and West Africa, but not the other African regions studied. We then performed the analogous PB test, which came to the same conclusion. The differences between means, confidence intervals and p-values are shown in Table 4.8 and 4.9 for Dunnett's test and Table 4.10 for the PB test. An illustration of the method is depicted in Figure 4.5, as a histogram of the PB simulated null distribution with its critical value and the test statistic for comparing Asia to West Africa shown. We note that Ziegler et al. (2016) also found a significant difference between Asia and East and Central Africa for this isotope (see Table A4 in their supplementary material). The data they report in their results (see Table 1 of Ziegler et al. (2016)) contained 507 observations including 20 from Asia, while the data we used from their supplementary material contained 495 observations, with only 8 observations from Asia. Additionally, in the supplementary material data, Rwanda appears to be classified as part of Central Africa, but in their Table 1, it is classified as East Africa. So, it is possible that the difference between our findings and those of Ziegler et al. (2016) could be due to these differences in sample sizes, with the missing observations coming from Asia.

Finally, Ziegler et al. (2016) state that, with some exceptions, single isotope markers alone are of little usefulness for forensic purposes, so we emphasize that findings from this study for one particular isotope would be best combined with other results for practical application. They also state several biological and environmental factors to consider when interpreting findings. However, in the interest of brevity, we chose to illustrate the proposed method with only one of the isotope ratios.

	Diff	Lower CI	Upper CI	p-value
Central Africa-Asia	0.89	-0.53	2.31	0.27
East Africa-Asia	1.29	-0.23	2.80	0.11
Southern Africa-Asia	0.44	-0.96	1.83	0.69
West Africa-Asia	-2.64	-4.09	-1.19	0.00

Table 4.8: Results from Dunnett's Test, Elephant Data.

	Diff	Lower CI	Upper CI	p-value
Central Africa-Asia	0.09	-0.07	0.25	0.35
East Africa-Asia	0.12	-0.05	0.29	0.21
Southern Africa-Asia	0.04	-0.12	0.20	0.77
West Africa-Asia	-0.38	-0.54	-0.22	0.00

Table 4.9: Results from Dunnett's Test, Log Elephant Data.

	Diff	Lower CI	Upper CI	p-value
Central Africa-Asia	0.89	-0.42	2.19	0.19
East Africa-Asia	1.29	-0.35	2.93	0.12
Southern Africa-Asia	0.44	-0.81	1.69	0.61
West Africa-Asia	-2.64	-3.91	-1.36	0.00

Table 4.10: Results from PB Test, Elephant Data.

In this case, while the log transformation corrected the violation of the normality assumption, it could not completely correct the violation of the equal variance assumption though the fitted-residual plot was somewhat improved. Although both

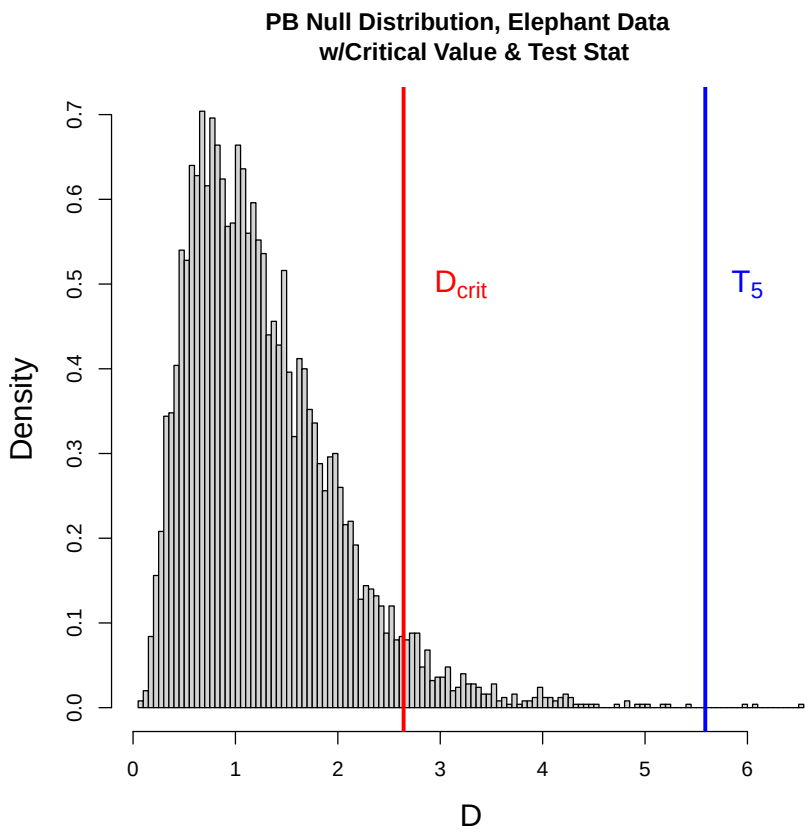

Figure 4.5: PB Distribution, Elephant Tusk Data.

methods came to the same conclusion, a researcher could have more confidence in the result from the PB method since it does not require the equal variance assumption, and avoids the need for transformation. Additionally, for the difference between Asia and West Africa, the PB method produced narrower confidence intervals. In this case, the mean squared error from the ANOVA model, which would be used as a pooled variance estimate in the traditional Dunnett's test, was 3.046, which is much larger than the variance for Asia and West Africa in particular, so would produce a test statistic that is smaller than necessary (and thus less likely to reject the null hypothesis of no difference between these two groups). This is somewhat similar to the simulation setting (for evaluating Type I error) of $\boldsymbol{n_4}$ with $\boldsymbol{\sigma_5^2}$, although in that

simulation setting the largest group had the largest variance; in the elephant dataset, the largest group size had the third largest variance. Still, in that simulation setting, Dunnett's test was too conservative due to the pooled variance estimate being too large for some groups, which is similar to the findings here of confidence intervals for Dunnett's test being wider than those of the PB test for the Asia and West Africa comparison.

While the PB method uses the normality assumption, it uses group means in its calculations, which should be approximately normal regardless of the distribution of the individual observations, at least for large samples, by the Central Limit Theorem (Casella and Berger, 2002), so it is plausible that the PB test could also be robust to violations of the normality assumption.

4.5 Conclusions and Discussion

In this research, we looked at Dunnett's test from a parametric bootstrap view and proposed a PB test for comparing treatment groups with the control. Simulation results show that both Dunnett's test and the PB test give acceptable results under the equal variance assumption. Additionally, when data are balanced, Dunnett's test performs satisfactorily in most heteroscedastic cases. However, for heteANOVA problems, where the equal variance assumption is violated and data are unbalanced, Dunnett's test no longer provides reasonable nominal levels, while the proposed PB method works well. The two real examples illustrate that the classical way of transformation to deal with unequal variance is not guaranteed and interpretation of the results after transformation could be difficult. The proposed PB test is robust to violation of equal variance even when data are unbalanced, and it is easy to implement.

While Dunnett's test performed satisfactorily with most balanced data cases in

simulations, the rejection rate can be much higher or lower than the nominal level for the heteANOVA problem. One reason for this is that if one group's variance is much smaller than the others, the pooled variance estimate will be too large, leading to an artificially small test statistic. On the other hand, if one group's variance is much larger than the others, the pooled variance estimate will be too small, leading to an artificially large test statistic.

Some limitations of the proposed PB method are that it may require the normality assumption for small sample sizes, so if a particular dataset violates both assumptions, a transformation may still be needed. Additionally, as described in Christensen (2016) section 4.3, caution should be exercised when making practical decisions based on differences in means between groups with unequal variances. For example, if a lower value of a response is desired, such as blood pressure, a treatment group with a smaller mean and smaller variance may have a smaller probability of achieving the desired outcome than a treatment group with a larger mean but also larger variance. Thus, additional consideration of implications for the practical issue being studied is warranted. This issue is illustrated in the iron data example. Despite these limitations, the proposed PB test is a viable method for performing multiple comparisons of treatment vs control for the heteANOVA problem.

Chapter 5

Conclusions and Future Work

5.1 Phylogenetic Inference

The methods developed in Chapter 2 show that robustness of the GLASS/STEM tree to gene tree estimation error (GTEE) can be improved through measurement error modeling. Limitations to these new methods include, as discussed in Chapter 2, that they perform fewer functions than the STEM software, which allows for multiple individuals of the same species, returns the likelihood, and handles missing data, while genX and the Bayesian method simply return an inferred species tree. While it was not designed to take the place of other methods such as ASTRAL (Zhang et al., 2018), it could provide starting trees for other possibly slower methods. Additionally, while the genX method is statistically consistent when the expected value of the errors between true and estimated gene trees is zero, this assumption does not appear to be true in most simulated cases. The Bayesian approach discussed in Chapter 2 can be more accurate than the genX method for some settings, but can be prone to convergence issues and requires more computation time. A possible area for future development of the Bayesian method could explore use of other priors elicited

from a wider variety of simulated trees or perhaps additional empirical datasets. Since the new methods still show lower accuracy with lower values of the population scaled mutation rate θ and shorter DNA sequence lengths, they do not appear to be completely correcting for GTEE. This could be explored further by inferring species trees from known gene trees using STEM, inferring species trees from estimated gene trees (estimated from DNA sequences simulated from the known ones) using genX, and then comparing the RF distances of both to the true species tree. The improvement noted is limited to the scenarios simulated in this study; it is unknown how the new methods would perform outside of these. However, the genX method did perform fairly well on one of the empirical datasets shown here, and the Bayes X method inferred the correct tree from both empirical datasets. Areas for future research in this area could include studying the method on a wider variety of trees, which may allow for further correction for GTEE, exploring whether other distance matrix methods of species tree inference could also be improved with measurement error modeling, and extending the method to perform additional functions.

5.2 PB Methods for heteANOVA Data

The methods shown in Chapters 3 and 4 extend previously developed methods for handling violation of the equal variance assumption in one- and two-way ANOVA models to the three-way ANOVA model, and we expect that they could be further extended to higher-way models, though interpretation could be complicated (as with standard higher-way models). A major benefit of these methods is that they help avoid the need for transforming data to meet assumptions, which can also make interpretation more complicated. A potential limitation of these methods is that they may still require the normality assumption for small sample sizes, so transformation could still be needed in this case. However, with large sample sizes, the group means

used in these methods are approximately normal regardless of the distribution of the individual observations according to the Central Limit Theorem. In the potato dataset example used in Chapter 3, where the normality assumption was violated and sample sizes fairly small, the method appeared robust to this violation. One area for future research would be to try these methods on additional real datasets. The computer code shown in the appendix only returns p-values and/or critical values, but is easily modified for the MCP to return additional output such as confidence intervals (as shown with the PB version of Dunnett's test applied to the iron data). While the PB methods of testing main and interaction effects in a heteANOVA model do not depend on the identifiability constraints chosen, these constraints would need to be specified to obtain parameter estimates, and the code as written does not calculate parameter estimates. This function would be desirable if the algorithms here were compiled into a software package. Additionally, future research could involve extending the methods to more complicated models. Finally, as discussed in earlier chapters, caution is needed with practical application of results when the treatment groups considered have unequal variance, as a group with a mean closer to a desired level could have a lower probability of reaching that level than a group with a mean slightly further from the desired level, but with larger variance. Nevertheless, since the unequal variance issue does arise in practice, and data may be unbalanced for reasons out of an experimenter's control, methods for dealing with the issue are useful, and the methods here provide a viable option that avoids the need for transformation of the data.

Appendices

Appendix A

R Code for genX

```
library(ape)
library(gdata)
pwd1 <- function(tree){
nt <- length(tree$tip.label) #count number of taxa
dm <- matrix(0, nt, nt)  #set up matrix to store pwdist for one tree
    taxa <- as.character(1:nt)  #taxa need to be named 1:4 etc. in order.
    for (k in 1:(nt-1)){
for (i in 1:(nt-k)){
dm[i,i+k] <- max(drop.tip(tree, taxa[taxa!=taxa[i] & taxa!=taxa[i+k]])$edge.length)
#pwdist bw taxa (i, i+k)
} #end i loop
} #end k loop
pwd1 <- t(dm)[lower.tri(t(dm))] #transpose to get the values in the "right" order
return(pwd1)
}

#get a matrix of pairwise distances from multiple trees
pwdists <- function(in.file){ #start function
trees <- read.tree(in.file) #read in trees
est_gt <- data.frame(matrix(unlist(lapply(trees, pwd1)), byrow=TRUE, nrow=length(trees)))
return(est_gt)
}

dist.mat <- function(pwdist.obj, nt, theta){
dm <- matrix(0, nt, nt)
mins<- apply(pwdist.obj,2, function(x) min(x[x>0]))/theta
```

```
upperTriangle(dm, byrow=TRUE) <- mins #needs library(gdata)
return(dm)
}


genX <- function(W, nloci, nspecies, theta) {  #W is output from pwdists function
W <- W/(theta/2)
est.tau <- apply(W, 2, function(x) mean(x)-1)
expos <-  matrix(replicate(nloci*choose(nspecies,2), rexp(1) ), nrow=nloci, ncol=choose(nspecies,2))
#just take one of the exponentials for each entry, not the mean of 30 of them
X.tilde <- matrix(NA, nloci, choose(nspecies,2))
for (j in 1:choose(nspecies,2)){
X.tilde[,j] <- expos[,j] + est.tau[j]
}
dm <- dist.mat(X.tilde, nt=nspecies, theta=1)  #already accounted for theta
hc <- hclust(as.dist(t(dm)), method="single")
estST <- as.phylo(hc)
return(estST)
}

##load the above functions, assign W <- pwdists("genetrees.txt")
##where "genetrees.txt" is file of trees in Newick format
##then run genX function with the indicated arguments
```

Appendix B

R Code for PB Algorithms 1 – 6

```
####Algorithm 1
alg.ABC <- function(ns, ybars, s2, a, b, c, L){
S <- diag(s2/ns) ##make S matrix

##make terms for X matrix
J.abc <- rep(1, a*b*c)
I.a <- diag(a)
I.b <- diag(b)
I.c <- diag(c)
J.bc <- rep(1, b*c)
J.a <- rep(1, a)
J.b <- rep(1,b)
J.c <- rep(1,c)
I.ab <- diag(a*b)
I.bc <- diag(b*c)

X <- as.matrix(cbind(
J.abc, kronecker(I.a, J.bc), kronecker(J.a, kronecker(I.b, J.c)),
kronecker(J.a, kronecker(J.b, I.c)), kronecker(I.ab, J.c), kronecker(I.a, kronecker(J.b, I.c)),
kronecker(J.a, I.bc)))

#test statistic
library(MASS)
SI <- t(ybars)%*%solve(S)%*%ybars -
t(ybars)%*%solve(S)%*%X%*%ginv(t(X)%*%solve(S)%*%X)%*%t(X)%*%solve(S)%*%ybars
```

```
##Q, counts how many times test stat is less than PB pivot variable
Q <- NULL
for(j in 1:L) {
ybar.B <- NULL
S2B <-NULL
for (i in 1:length(ybars)) {
ybar.B[i] <- rnorm(1, mean=0, sd=sqrt(s2/ns)[i])  ##create bootstrap mean vector
S2B[i] <- rchisq(1, df=(ns[i]-1)) * s2[i]/(ns[i]-1) ##create bootstrap variances vector
}
SB <- diag(S2B/ns)

##PB variable:
SIB <-  t(ybar.B)%*%solve(SB)%*%ybar.B -
t(ybar.B)%*%solve(SB)%*%X%*%ginv(t(X)%*%solve(SB)%*%X)%*%t(X)%*%solve(SB)%*%ybar.B

Q[j] <- ifelse(SIB>SI, 1, 0)
}
return(sum(Q)/length(Q))  ##p-value
}



#Algorithm 2
alg.BC <- function(ns, ybars, s2, a, b, c, L){
S <- diag(s2/ns) ##make S matrix

##make terms for X matrix
J.abc <- rep(1, a*b*c)
I.a <- diag(a)
I.b <- diag(b)
I.c <- diag(c)
J.bc <- rep(1, b*c)
J.a <- rep(1, a)
J.b <- rep(1,b)
J.c <- rep(1,c)
I.ab <- diag(a*b)
I.bc <- diag(b*c)

X <- as.matrix(cbind(
J.abc, kronecker(I.a, J.bc), kronecker(J.a, kronecker(I.b, J.c)),
kronecker(J.a, kronecker(J.b, I.c)), kronecker(I.ab, J.c), kronecker(I.a, kronecker(J.b, I.c))))

#test statistic
```

```
library(MASS)
SI <- t(ybars)%*%solve(S)%*%ybars -
t(ybars)%*%solve(S)%*%X%*%ginv(t(X)%*%solve(S)%*%X)%*%t(X)%*%solve(S)%*%ybars

##Q, counts how many times test stat is less than PB pivot variable
Q <- NULL
for(j in 1:L) {
ybar.B <- NULL
S2B <-NULL
for (i in 1:length(ybars)) {
ybar.B[i] <- rnorm(1, mean=0, sd=sqrt(s2/ns)[i])  ##create bootstrap mean vector
S2B[i] <- rchisq(1, df=(ns[i]-1)) * s2[i]/(ns[i]-1) ##create bootstrap variances vector
}
SB <- diag(S2B/ns)

##PB variable:
SIB <-  t(ybar.B)%*%solve(SB)%*%ybar.B -
t(ybar.B)%*%solve(SB)%*%X%*%ginv(t(X)%*%solve(SB)%*%X)%*%t(X)%*%solve(SB)%*%ybar.B

Q[j] <- ifelse(SIB>SI, 1, 0)
}
return(sum(Q)/length(Q))  ##p-value
}
```

```
############################
#Algorithm 3
alg.C <- function(ns, ybars, s2, a, b, c, L){
S <- diag(s2/ns) ##make S matrix

##make terms for X matrix
J.abc <- rep(1, a*b*c)
I.a <- diag(a)
I.b <- diag(b)
I.c <- diag(c)
J.bc <- rep(1, b*c)
J.a <- rep(1, a)
J.b <- rep(1,b)
J.c <- rep(1,c)
I.ab <- diag(a*b)
I.bc <- diag(b*c)
```

```
X <- as.matrix(cbind( J.abc, kronecker(I.a, J.bc), kronecker(J.a, kronecker(I.b, J.c))))

#test statistic
library(MASS)
SI <- t(ybars)%*%solve(S)%*%ybars -
t(ybars)%*%solve(S)%*%X%*%ginv(t(X)%*%solve(S)%*%X)%*%t(X)%*%solve(S)%*%ybars

##Q, counts how many times test stat is less than PB pivot variable
Q <- NULL
for(j in 1:L) {
ybar.B <- NULL
S2B <-NULL
for (i in 1:length(ybars)) {
ybar.B[i] <- rnorm(1, mean=0, sd=sqrt(s2/ns)[i])  ##create bootstrap mean vector
S2B[i] <- rchisq(1, df=(ns[i]-1)) * s2[i]/(ns[i]-1) ##create bootstrap variances vector
}
SB <- diag(S2B/ns)

##PB variable:
SIB <-  t(ybar.B)%*%solve(SB)%*%ybar.B -
t(ybar.B)%*%solve(SB)%*%X%*%ginv(t(X)%*%solve(SB)%*%X)%*%t(X)%*%solve(SB)%*%ybar.B

Q[j] <- ifelse(SIB>SI, 1, 0)
}
return(sum(Q)/length(Q))  ##p-value
}
```

```
#Algorithm 4
alg.C.AB <- function(ns, ybars, s2, a, b, c, L){
S <- diag(s2/ns) ##make S matrix

##make terms for X matrix
J.abc <- rep(1, a*b*c)
I.a <- diag(a)
I.b <- diag(b)
I.c <- diag(c)
J.bc <- rep(1, b*c)
J.a <- rep(1, a)
J.b <- rep(1,b)
```

```
J.c <- rep(1,c)
I.ab <- diag(a*b)
I.bc <- diag(b*c)

X <- as.matrix(cbind(J.abc, kronecker(I.a, J.bc), kronecker(J.a, kronecker(I.b, J.c)),
kronecker(I.ab, J.c)))

#test statistic
library(MASS)
SI <- t(ybars)%*%solve(S)%*%ybars -
t(ybars)%*%solve(S)%*%X%*%ginv(t(X)%*%solve(S)%*%X)%*%t(X)%*%solve(S)%*%ybars

##Q, counts how many times test stat is less than PB pivot variable
Q <- NULL
for(j in 1:L) {
ybar.B <- NULL
S2B <-NULL
for (i in 1:length(ybars)) {
ybar.B[i] <- rnorm(1, mean=0, sd=sqrt(s2/ns)[i])  ##create bootstrap mean vector
S2B[i] <- rchisq(1, df=(ns[i]-1)) * s2[i]/(ns[i]-1) ##create bootstrap variances vector
}
SB <- diag(S2B/ns)

##PB variable:
SIB <-  t(ybar.B)%*%solve(SB)%*%ybar.B -
t(ybar.B)%*%solve(SB)%*%X%*%ginv(t(X)%*%solve(SB)%*%X)%*%t(X)%*%solve(SB)%*%ybar.B

Q[j] <- ifelse(SIB>SI, 1, 0)
}
return(sum(Q)/length(Q))  ##p-value
}
```

```
##Algorithm 5
#######ALGORITHM for PB mult comparisons of the levels of factor A
#make a PB "Q" distrib for the multiple comparisons and calculate a test stat

Q.test.dist <- function(L=5000, ns, means, s2, alpha=0.05, a, b, c){
##Calculate weights for actual test stat and the PB pivot variable
library(plyr)
```

```
ns.ind <- arrange(expand.grid(A=1:a, B=1:b, C=1:c), A,B)
n.grp <- array(0, c(a,b,c))  ##array does not fill entries in the desired order.
for(i in 1:a){
for(j in 1:b){
for(k in 1:c)
n.grp[i,j,k] = ns[which(ns.ind$A==i & ns.ind$B==j & ns.ind$C==k)]
}
}
v.weight <- matrix(0, b, c)
for(j in 1:b){
for(k in 1:c){
v.weight[j,k] <- sum(n.grp[,j,k])
}
}
vjk <- as.vector(t(v.weight/sum(ns)))  ##the weights in order of the j,k index

#calculate factor level estimated means (using the weights) for the test statistic
ybari <- rep(0,a)
ni <- rep(0,a)
var.YA <- rep(0, a)
ni[1] <- sum(ns[1:(b*c)])
ybari[1] <- sum(vjk*means[1:(b*c)])
var.YA[1] <- sum(vjk^2 * (s2/ns)[1:(b*c)])
for(i in 2:a){
ybari[i] <- sum(vjk*means[(b*c*(i-1)+1):(i*b*c)])
ni[i] <- sum(ns[(b*c*(i-1)+1):(i*b*c)])
var.YA[i] <- sum(vjk^2 * (s2/ns)[(b*c*(i-1)+1):(i*b*c)])
}

Qtest.mat <- matrix(0,a,a)
#we just fill in upper triangular part
for (r in 1: (a -1))
for (s in (r+1):(a)){
Qtest.mat[r,s]<- abs(ybari[r] - ybari[s])/sqrt(var.YA[r] + var.YA[s])
}
   Q.test <- max(Qtest.mat)

##calculate the parts of the PB pivot variable
Q <- rep(0, L)
for(i in 1:L){  ##calculate the bootstrap means and sample variances

y.B <- rep(0, length(means))
s2.B <- rep(0, length(s2))
```

```
for (j in 1:length(means)){
y.B[j]<- rnorm(1, 0, sqrt(s2[j]/ns[j]))
s2.B[j] <- rchisq(1, df=(ns[j]-1))*s2[j]/(ns[j]-1)
}#end the j loop

#now Q will be the PB analogy of the Q.test above. we use the same ni's
yB.bari <- rep(0,a)
var.YBA <- rep(0, a)
yB.bari[1] <- sum(vjk*y.B[1:(b*c)])
var.YBA[1] <- sum(vjk^2 * (s2.B/ns)[1:(b*c)])
for(m in 2:a){
yB.bari[m] <- sum(vjk*y.B[(b*c*(m-1)+1):(m*b*c)])
var.YBA[m] <- sum(vjk^2 * (s2.B/ns)[(b*c*(m-1)+1):(m*b*c)])
} #end m loop

Qmat <- matrix(0,a,a)
#we just fill in upper triangular part
for (r in 1: (a -1))
for (s in (r+1):a){
Qmat[r,s]<- abs(yB.bari[r] - yB.bari[s])/sqrt(var.YBA[r] + var.YBA[s])
}

   Q[i] <-max(Qmat)
} #end i loop that has L reps
Q.crit <-quantile(Q, 1-alpha)
list(Q.crit = Q.crit, Q.test = Q.test)
}
```

```
##Algorithm 6
#######ALGORITHM for PB mult comparisons of the levels of AB int. term
#make a PB "Q" distrib for the multiple comparisons and calculate a test stat

Q.ABmc <- function(L=5000, ns, means, s2, alpha=0.05, a, b, c){
  ##get the ns, means and s2 in an array so we can identify the indices
  library(plyr)
  ns.ind <- arrange(expand.grid(A=1:a, B=1:b, C=1:c), A,B)
  n.grp <- array(0, c(a,b,c))  ##array does not fill entries in the desired order.
```

```
  s2.grp <- array(0, c(a,b,c))
  means.grp <- array(0, c(a,b,c))
  for(i in 1:a){
    for(j in 1:b){
      for(k in 1:c){
        n.grp[i,j,k] = ns[which(ns.ind$A==i & ns.ind$B==j & ns.ind$C==k)]
        s2.grp[i,j,k] = s2[which(ns.ind$A==i & ns.ind$B==j & ns.ind$C==k)]
        means.grp[i,j,k] = means[which(ns.ind$A==i & ns.ind$B==j & ns.ind$C==k)]
      }
    }
  }

  ##Calculate weights vk for actual test stat and the PB pivot variable
  vk <- rep(0, c)
  for(k in 1:c){
    vk[k] <- sum(n.grp[,,k])
  }
  v.wt.k <- vk/sum(ns) ##the weights in order of the k index

  #calculate estimated means (using the weights) for each level of AB for the test statistic
  ybarij <- matrix(0, a, b)
  var.YAB <- matrix(0, a, b)

  for(i in 1:a){
    for(j in 1:b){
        ybarij[i,j] <- sum(v.wt.k*means.grp[i,j,])
        var.YAB[i,j] <- sum(v.wt.k^2 * s2.grp[i,j,]/n.grp[i,j,])
    }
}

 ybarijVect <- as.vector(ybarij)
 var.YABvect <- as.vector(var.YAB)

Qtest.mat <- matrix(0,a*b,a*b)
#we just fill in upper triangular part
for (r in 1: ((a*b) -1))
for (s in (r+1):(a*b)){
Qtest.mat[r,s]<- abs(ybarijVect[r] - ybarijVect[s])/sqrt(var.YABvect[r] + var.YABvect[s])
}
   Q.test <- max(Qtest.mat)


  ##calculate the parts of the PB pivot variable
```

```
  Q <- rep(0, L)
  for(l in 1:L){  ##calculate the bootstrap means and sample variances
    y.B <- rep(0, length(means))
    s2.B <- rep(0, length(s2))

    for (j in 1:length(means)){
      y.B[j]<- rnorm(1, 0, sqrt(s2[j]/ns[j]))
      s2.B[j] <- rchisq(1, df=(ns[j]-1))*s2[j]/(ns[j]-1)
    }#end the j loop

#put the bootstrap means and s2's in indexed arrays
    s2B.grp <- array(0, c(a,b,c))
    meansB.grp <- array(0, c(a,b,c))
    for(i in 1:a){
      for(j in 1:b){
        for(k in 1:c){
          s2B.grp[i,j,k] = s2.B[which(ns.ind$A==i & ns.ind$B==j & ns.ind$C==k)]
          meansB.grp[i,j,k] = y.B[which(ns.ind$A==i & ns.ind$B==j & ns.ind$C==k)]
          n.grp[i,j,k] = ns[which(ns.ind$A==i & ns.ind$B==j & ns.ind$C==k)]
        }
      }
    }

    #now Q will be the PB analogy of the Q.test above, use same weights
    yB.barij <- matrix(0, a, b)
    varB.YAB <- matrix(0, a, b)

    for(i in 1:a){
      for(j in 1:b){
        yB.barij[i,j] <- sum(v.wt.k*meansB.grp[i,j,])
        varB.YAB[i,j] <- sum(v.wt.k^2 * s2B.grp[i,j,]/n.grp[i,j,])
      }
    }

    yB.barijVect <- as.vector(yB.barij)
 varB.YABvect <- as.vector(varB.YAB)

Qmat <- matrix(0,a*b,a*b)
#we just fill in upper triangular part
for (r in 1: ((a*b) -1))
for (s in (r+1):(a*b)){
Qmat[r,s]<- abs(yB.barijVect[r] - yB.barijVect[s])/sqrt(varB.YABvect[r] + varB.YABvect[s])
}
```

```
    Q[l] <- max(Qmat)
  } #end l loop that has L reps

  Q.crit <-quantile(Q, 1-alpha)
# list(Q=Q, Q.crit = Q.crit, Q.test = Q.test) #this return list for testing function
  list(Q.crit = Q.crit, Q.test = Q.test)
}
```

Appendix C

R Code for Dunnett's Test PB Algorithm

```
#ns, means, s2: vectors of group sample sizes, means and variances; alpha: desired alpha level
dunnett.PB <- function(L, ns, means, s2, alpha){ #L is #bootstrap runs
D <- rep(0, L)
r <- length(ns) #number of groups
pairs.data <- rep(0, r)
diffs <- rep(0, r)
for(j in 1:r){
diffs[j] <- means[1]-means[j]
pairs.data[j] <- abs(means[1]-means[j])/sqrt( (s2[1]/ns[1]) + (s2[j]/ns[j]))
#the first 'pairs' will be 0
}
test.stat <- max(pairs.data)
pairs <- rep(0, r)
##storage vector for the differences between group means for bootstrap data
for(i in 1:L){
y.B <- rep(0, r)
s2.B <- rep(0, r)
for (j in 1:r){
y.B[j]<- rnorm(1)*sqrt(s2[j]/ns[j])
s2.B[j] <- rchisq(1, df=(ns[j]-1))*s2[j]/(ns[j]-1)
pairs[j] <- abs(y.B[1]-y.B[j])/sqrt( (s2.B[1]/ns[1]) + (s2.B[j]/ns[j]))
#the first one will be 0
}
```

```
D[i]<- max(pairs)
}
pvals <- rep(0, r)
for(j in 1:r){
pvals[j] <- length(which(D>pairs.data[j]))/L
}
D.crit <- quantile(D, 1-alpha)
list(result = data.frame(diffs=diffs, test.stats=pairs.data, pvals=pvals), D.crit = D.crit)
}
```

References

M. M. Ananda and S. Weerahandi. Two-way ANOVA with unequal cell frequencies and unequal variances. *Statistica Sinica*, 7(3):631–646, July 1997.

S. F. Arnold. *The Theory of Linear Models and Multivariate Analysis*. John Wiley and Sons, 1981.

P. Bao and M. M. Ananda. Performance of two-way ANOVA procedures when cell frequencies and variances are unequal. *Communications in Statistics - Simulation and Computation*, 30(4):805–829, 2001.

L. Carbone, R. Alan Harris, S. Gnerre, et al. Gibbon genome and the fast karyotype evolution of small apes. *Nature*, 513(7517):195–201, September 2014.

R. J. Carroll, D. Ruppert, L. A. Stefanski, and C. M. Crainiceanu. *Measurement Error in Nonlinear Models: A Modern Perspective*. Chapman & Hall/CRC, Boca Raton, FL, second edition, 2006.

G. Casella and R. L. Berger. *Statistical Inference*. Duxbury, Pacific Grove, CA, Second edition, 2002.

W. S. C. Cheng, T. L. Murphy, M. T. Smith, W. G. E. Cooksley, J. W. Halliday, and L. W. Powell. Dose-dependent pharmacokinetics of caffeine in humans: Relevance as a test of quantitative liver function. *Clinical Pharmacology & Therapeutics*, 47 (4):516–524, 1990. doi: https://doi.org/10.1038/clpt.1990.66.

R. Christensen. *Analysis of Variance, Design, and Regression: Applied Statistical Methods*. Chapman and Hall/CRC, Boca Raton, FL, 1996.

R. Christensen. *Analysis of Variance, Design, and Regression: Linear Modeling for Unbalanced Data*. CRC Press, Boca Raton, FL, 2nd edition, 2016.

R. Christensen. *Plane Answers to Complex Questions: The Theory of Linear Models*. Springer, 5th edition, 2018.

R. Christensen, W. Johnson, A. Branscum, and T. Hanson. *Bayesian Ideas and Data Analysis*. Chapman and Hall/CRC, Boca Raton, FL, 2011.

M. DeGiorgio and J. Degnan. Robustness to divergence time underestimation when inferring species trees from estimated gene trees. *Syst. Biol.*, 63(1):66–82, 2014.

C. Dunnett and M. Sobel. Approximations to the probability integral and certain percentage points of a multivariate analogue of Student's t-distribution. *Biometrika*, 42(1/2):258–260, June 1955.

C. W. Dunnett. A multiple comparison procedure for comparing several treatments with a control. *Journal of the American Statistical Association*, 50(272):1096–1121, 1955. ISSN 01621459. URL `http://www.jstor.org/stable/2281208`.

C. W. Dunnett and M. Sobel. A bivariate generalization of Student's t-distribution, with tables for certain special cases. *Biometrika*, 41(1/2):153–169, June 1954.

B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, New York, NY, 1993.

J. Felsenstein. *Inferring Phylogenies*. Sinauer Associates, Inc, Sunderland, MA, 2004.

J. Felsenstein. PHYLIP(Phylogeny Inference Package) Version 3.7a. Distributed by the author, 2009.

T. S. Ferguson. *A Course in Large Sample Theory*. Chapman & Hall, London, UK, 1996.

J. Fox and S. Weisberg. *An R Companion to Applied Regression*. Sage, Thousand Oaks CA, third edition, 2019. URL `https://socialsciences.mcmaster.ca/jfox/Books/Companion/`.

P. A. Games and J. F. Howell. Pairwise multiple comparison procedures with unequal N's and/or variances: A monte carlo study. *Journal of Educational Statistics*, 1 (2):113–125, 1976.

A. J. Hayter. A proof of the conjecture that the Tukey-Kramer multiple comparisons procedure is conservative. *The Annals of Statistics*, 12(1):61–75, 1984.

H. Huang, Q. HE, L. S. Kubatko, and L. L. Knowles. Sources of error inherent in species-tree estimation: Impact of mutational and coalescent effects on accuracy and implications for choosing among different methods. *Syst. Biol.*, 59(5):573–583, 2010.

G. S. James. The comparison of several groups of observations when the ratios of the population variances are unknown. *Biometrika*, 38(3/4):324–329, 1951. ISSN 00063444. URL `http://www.jstor.org/stable/2332578`.

E. M. Jewett and N. A. Rosenberg. iGLASS: An improvement to the GLASS method for estimating species trees from gene trees. *Journal of Computational Biology*, 19 (3):293–315, 2012.

A. Kim and J. H. Degnan. PRANC: ML species tree estimation from the ranked gene trees under coalescence. *Bioinformatics*, 36(18):4819–4821, 2020.

C. Y. Kramer. Extension of multiple range tests to group means with unequal numbers of replications. *Biometrics*, 12(3):307–310, September 1956.

K. Krishnamoorthy and F. Lu. A parametric bootstrap approach for ANOVA with unequal variances: Fixed and random models. *Computational Statistics and Data Analysis*, 51:5731–5742, 08 2007. doi: 10.1016/j.csda.2006.09.039.

L. Kubatko, B. Carstens, and L. Knowles. STEM: Species tree estimation using maximum likelihood for gene trees under coalescence. *Bioinformatics*, 25(7):971–973, 2009.

M. H. Kutner, C. J. Nachtsheim, J. Neter, and W. Li. *Applied Linear Statistical Models*. McGraw-Hill Irwin, New York, NY, 5th edition, 2005.

Z. B. Kutuk, E. Ergin, F. Y. Cakir, and S. Gurgan. Effects of in-office bleaching agent combined with different desensitizing agents on enamel. *Journal of Applied Oral Science*, 27, 00 2019. ISSN 1678-7757.

A. D. Leaché and B. Rannala. The Accuracy of Species Tree Estimation under Simulation: A Comparison of Methods. *Syst. Biol.*, 60(2):126–137, 2011.

E. Lehman. *Elements of Large-Sample Theory*. Springer-Verlag, LLC, New York, NY, 1999.

L. Liu, L. Yu, and S. V. Edwards. A maximum pseudo-likelihood approach for estimating species trees under the coalescent model. *BMC Evolutionary Biology*, 10(1):302, 2010a.

L. Liu, L. Yu, and D. K. Pearl. Maximum tree: a consistent estimator of the species tree. *J. Math. Biol.*, 60:95–106, 2010b.

W. P. Maddison. Gene trees in species trees. *Syst. Biol.*, 46(3):523–536, 1997.

E. K. Malloy and T. Warnow. To include or not to include: The impact of gene filtering on species tree estimation methods. *Syst. Biol.*, 67:285–303, 2018.

R. G. J. Miller. *Simultaneous Statistical Inference*. Springer-Verlag, New York; Heidelberg; Berlin, second edition, 1981.

S. Mirarab, M. S. Bayzid, B. Boussau, and T. Warnow*. Statistical binning enables an accurate coalescent-based estimation of the avian tree. *Science*, 346(6215): 1250463, 2014.

S. Mirarab, L. Nakhleh, and T. Warnow. Multispecies coalescent: Theory and applications in phylogenetics. *The Annual Review of Ecology, Evolution, and Systematics*, 52:247–268, 2021.

E. Mossel and S. Roch. Incomplete lineage sorting: Consistent phylogeny estimation from multiple loci. *IEEE/ACM transactions on computational biology and bioinformatics*, 7(1):166–171, 2010.

A. R. Nabhan and I. N. Sarkar. The impact of taxon sampling on phylogenetic inference: a review of two decades of controversy. *Briefings in Bioinformatics*, 13 (1):122–134, January 2012.

E. Paradis and K. Schliep. ape 5.0: an environment for modern phylogenetics and evolutionary analyses in R. *Bioinformatics*, 35:526–528, 2019.

M. Plummer. JAGS: A Program for Analysis of Bayesian Graphical Models using Gibbs Sampling. *3rd International Workshop on Distributed Statistical Computing (DSC 2003); Vienna, Austria*, 124, 04 2003.

M. Plummer. *rjags: Bayesian Graphical Models using MCMC*, 2021. URL `https://CRAN.R-project.org/package=rjags`. R package version 4-12.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2021. URL `https://www.R-project.org/`.

A. Rambut and N. Grassly. Seq-Gen: An application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. *Comput. Appl. Biosci.*, 13: 235–238, 1997.

B. Rannala and Z. Yang. Bayes estimation of species divergence times and ancestral population sizes using DNA sequences from multiple loci. *Genetics*, 164:1645–1656, 2003.

D. Robinson and L. Foulds. Comparison of phylogenetic trees. *Mathematical Biosciences*, 53(1-2):131–147, 1981.

S. Roch and T. Warnow. On the robustness to gene tree estimation error (or lack thereof) of coalescent-based species tree methods. *Syst. Biol.*, 64(4):663–676, 2015.

S. Roch, M. Nute, and T. Warnow. Long-branch attraction in species tree estimation: Inconsistency of partitioned likelihood and topology-based summary methods. *Syst. Biol.*, 68(2):281–297, 2019.

M. Sananman and D. W. Lear. Iron in chesapeake bay waters. *Chesapeake Science*, 2(3/4):207–209, 1961. ISSN 00093262. URL `http://www.jstor.org/stable/1351176`.

H. Scheffe. *The Analysis of Variance*. John Wiley and Sons, New York, 1959.

C.-M. Shi and Z. Yang. Coalescent-based analyses of genomic sequence data provide a robust resolution of phylogenetic relationships among major groups of gibbons. *Mol. Biol. Evol.*, 35(1):159–179, 2018.

A. Signorell et mult al. *DescTools: Tools for Descriptive Statistics*, 2020. URL `https://cran.r-project.org/package=DescTools`. R package version 0.99.38.

T. Stadler. *TreeSim: Simulating Phylogenetic Trees*, 2019. URL `https://CRAN.R-project.org/package=TreeSim`. R package version 2.4.

K. Strojek, K. H. Yoon, V. Hruba, M. Elze, A. M. Langkilde, and S. Parikh. Effect of dapagliflozin in patients with type 2 diabetes who have inadequate glycaemic control with glimepiride: a randomized, 24-week, double-blind, placebo-controlled trial. *Diabetes, Obesity and Metabolism*, 13(10):928–938, 2011. doi: https://doi.org/10.1111/j.1463-1326.2011.01434.x.

Y. Su, J. Reedy, and R. J. Carroll. Clustering in general measurement error models. *Statistica Sinica*, 28(4):2337–2351, 2018.

R. Tallarida and R. Murray. *Dunnett's Test (Comparison with a Control). In: Manual of Pharmacologic Calculations*. Springer, New York, NY, 1987.

W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, fourth edition, 2002. URL `http://www.stats.ox.ac.uk/pub/MASS4/`. ISBN 0-387-95457-0.

J. Wakeley. *Coalescent Theory: An Introduction*. Roberts & Company, Greenwood Village, CO, 2009.

G. R. Warnes, B. Bolker, G. Gorjanc, G. Grothendieck, A. Korosec, T. Lumley, D. MacQueen, A. Magnusson, J. Rogers, et al. *gdata: Various R Programming Tools for Data Manipulation*, 2017. URL `https://CRAN.R-project.org/package=gdata`. R package version 2.18.0.

S. Weerahandi. ANOVA under unequal error variances. *Biometrics*, 51(2):589–599, 1995.

B. L. Welch. On the comparison of several mean values: An alternative approach. *Biometrika*, 38(3/4):330–336, 1951. ISSN 00063444. URL `http://www.jstor.org/stable/2332579`.

L. Winner. University of Florida, Dept of Statistics. Miscellaneous

Datasets: One-Way ANOVA/Independent Samples t-test, 2020. URL `http://users.stat.ufl.edu/ winner/data/ironwater.dat`.

Y. Wu. Coalescent-based species tree inference from gene tree topologies under incomplete lineage sorting by maximum likelihood. *Evolution*, 66(3):763–775, 2012. doi: https://doi.org/10.1111/j.1558-5646.2011.01476.x.

L.-W. Xu, F.-Q. Yang, A. Abula, and S. Qin. A parametric bootstrap approach for two-way ANOVA in presence of possible interactions with unequal variances. *Journal of Multivariate Analysis*, 115:172–180, 2013.

E. Yigit and F. Gökpınar. A simulation study on tests for one-way anova under the unequal variance assumption. *Communications, Faculty of Science, University of Ankara Series A1*, 59(2):15–34, 2010.

A. Zeileis and T. Hothorn. Diagnostic checking in regression relationships. *R News*, 2(3):7–10, 2002. URL `https://CRAN.R-project.org/doc/Rnews/`.

C. Zhang, M. Rabiee, E. Sayyari, and S. Mirarab. ASTRAL-III: Polynomial time species tree reconstruction from partially resolved gene trees. *BMC Bioinformatics*, 19(S6):153, 2018.

G. Zhang. A parametric bootstrap approach for one-way ANOVA under unequal variances with unbalanced data. *Communications in Statistics - Simulation and Computation*, 44(4):827–832, 2015a. doi: 10.1080/03610918.2013.794288. URL `https://doi.org/10.1080/03610918.2013.794288`.

G. Zhang. Simultaneous confidence intervals for pairwise multiple comparisons in a two-way unbalanced design with unequal variances. *Journal of Statistical Computation and Simulation*, 85 (13):2727–2735, 2015b. doi: 10.1080/00949655.2014.935735. URL `https://doi.org/10.1080/00949655.2014.935735`.

S. Zhu, J. H. Degnan, S. J. Goldstien, and B. Eldon. Hybrid-Lambda: simulation of multiple merger and Kingman gene genealogies in species networks and species trees. *BMC Bioinformatics*, 16(292), 2015.

S. Ziegler, S. Merker, B. Streit, M. Boner, and D. E. Jacob. Towards understanding isotope variability in elephant ivory to establish isotopic profiling and source-area determination. *Biological Conservation*, 197:154–163, 2016.