IMPROVING CNV DETECTION EFFICACY IN A NONMODEL ORGANISM THROUGH SIMULATIONS AND OPTIMIZATION OF PARAMETERS IN THE EXOMEDEPTH PROGRAM

A Thesis

by

WEIXI ZHU

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Claudio Casola
Committee Members,	Alan Dabney
	Carol A. Loopstra
Head of Department,	Dirk B. Hays

August 2020

Major Subject: Molecular and Environmental Plant Sciences

ABSTRACT

Copy number variants (CNVs) represent changes in the number of DNA segments from 50 bp to several millions of nucleotides that often include genic sequences. CNVs play a critical role in evolution and are related to disease in humans. Increasingly, genome and exome resequencing efforts have been used to identify CNVs. Whole exome sequencing (WES) data provide the advantage of informing on polymorphisms, including CNVs, in genic regions at a fraction of the cost necessary for whole genome sequencing (WGS). However, the performance of current CNV detection tools using WES data in species with genomic architecture different from model organisms has yet to be determined. In this research, I investigated the ability of a widespread CNV detector relying on WES data, ExomeDepth, to accurately identify CNVs in loblolly pine (*Pinus taeda* L.), a major forest species in the U.S. that is characterized by a large genome size (>20 Gbp) and by available WES data. Using CNV simulations, I first determined the sensitivity and false discovery rate of ExomeDepth, which showed high sensitivity and low false discovery rate for deletions but performed relatively poorly with duplications. The detection of duplications is especially affected by ExomeDepth’s main parameter, transition probability. Importantly, intersecting detected CNVs from multiple resampled runs of ExomeDepth significantly decreases the false discovery rate for duplication, but it might be challenging to apply to large datasets because of the required computational power. Overall, this project has laid the foundations for the accurate detection of CNVs based on WES data in loblolly pine, which might be useful on other nonmodel organisms.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my committee chair, Dr. Casola, for his patience, warmth, continuous guidance and support throughout the course of this research.

I would like to thank my committee members, Dr. Dabney and Dr. Loopstra, for their valuable advice on my study.

I am grateful to Yue Xing for assistance with the use of SECNVs.

I appreciate Texas A&M high performance research computing center providing me with a great opportunity to use computer clusters.

Thanks also go to my friends and colleagues and the department faculty and staff for making my time at Texas A&M University a great experience.

Finally, thanks to my mother for her encouragement.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supervised by a thesis committee consisting of Dr. Claudio Casola, Dr. Carol A. Loopstra of the Department of Ecology and Conservation Biology and Dr. Alan Dabney of the Department of Statistics.

All work conducted for the thesis was completed by the student independently.

Funding Sources

This graduate study was supported by a fellowship from the Molecular and Environmental Plant Sciences Graduate Interdisciplinary Program at Texas A&M University. This work has been supported by the National Institute of Food and Agriculture, U.S. Department of 559 Agriculture, under award number TEX0-1-9599, Texas A&M AgriLife Research, and the Texas A&M Forest Service.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

1. INTRODUCTION

Copy number variants, or CNVs, are a common type of polymorphic structural variation in genomes. CNVs are often defined as DNA segments between 50 base pairs (bp) and several million nucleotides that differ in their copy number between individuals of the same species (Zarrei et al. 2015). A significant proportion of CNVs encompass gene regions (Locke et al. 2006). CNVs are estimated to occur in 4.8-9.5% of the human genome, and are known to be responsible for several genetic diseases (Zarrei et al. 2015, Lupski 2007). CNVs thus play a critical role in generating genetic variation in a population, and some CNVs have been found to be associated with adaptation (Hull et al. 2017). Therefore, investigating CNVs is essential to our understanding of genome evolution, adaptation and the onset of disease. Because of their role in pathogenicity, CNVs have been first and more extensively investigated in humans and a few other model species (Makino et al. 2013).

Hybridization-based microarray has traditionally been an important approach to detect CNVs. This method, although useful, has some important limitations, including higher costs compared to current DNA sequencing approaches and the inability to detect CNVs in genomic regions that are not covered by the array probes, and low sensitivity for duplications (Alkan et al. 2011). The development of high-throughput (next-generation) sequencing technology has made it possible to identify CNVs through the bioinformatic analysis of DNA sequencing fragments, or sequence “reads.” CNVs can be detected through both whole-genome sequencing (WGS) data and whole-exome sequencing (WES) data. As the names suggest, WGS represents the sequencing of the whole genome, whereas WES represents the sequencing of all or most known exons in a genome. Most functional DNA in a genome is present within exons and gene promoter regions, which tend to constitute a variable but generally small proportion in many eukaryotic genomes, e.g. ~1% for humans. Therefore, WES data provide the advantage of high sequencing coverage of most functional DNA at a fraction of the cost of WGS. This is especially important in species with a large genome, such as many vertebrates and land plants, which tend to share a comparable number of genes and exons but contain a higher proportion of non-genic DNA compared to species with smaller genomes.

Detecting CNVs through WES data is more challenging than through WGS data, because WES data represent short (from a few to a few thousand nucleotides) discrete regions in the genome. This fact rules out some strategies that can be applied to detect CNVs from WGS data. Typically, there are four methods for calling CNVs. All of them are based on the mapping of sequence reads to a reference genome and detect CNVs by identifying patterns that suggest the presence of deletions or duplications (Alkan et al. 2011). Read-pair methods compare the span and orientation of pair-ended reads and detect CNV by identifying the pair-ended reads that are inconsistent with the reference genome. Most types of structural variation can be detected by this method. The split-reads approach relies on the CNV breakpoints being present within a single read. The sequence assembly method consists in comparing de novo assemblies with the original reference genome and detecting changes in the distance between paired reads suggestive of either duplications or deletions in the sampled genome. The read-depth method investigates the divergence between the observed read depth to the expected read depth. If a target region duplicates, then a significantly higher read depth would be expected. Similarly, a deletion should show a reduced read depth. The read depth is the only approach that can be used to detect CNVs based on WES reads. Several software applications have been developed to detect CNVs in WES data following this approach. Almost all of these programs were originally designed for human sequencing data. The performance of these computer programs has been assessed in some studies (Tan et al. 2014, Zare et al. 2017), which also tend to be restricted to the human genome. In humans, false positive and false negative rates in CNV detection from WES data are usually estimated using array data (Tan et al. 2014, Zare et al. 2017). However, this information is rarely available in other species, especially nonmodel organisms. An alternative approach to determine the accuracy of CNV detection programs in the absence of array data consists of simulating and analyzing WES data using real genomic information from target species. However, no extensive study has been conducted so far on the ability of such programs to identify simulated CNVs. Given the paucity of information on CNV frequency and size distribution in most species, the optimization of existing CNV detection tools to individual reference genomes using simulations represents a critical step towards characterizing this type of genetic variants beyond a handful of model organisms.

With this goal in mind, my thesis project consisted of the implementation of CNV simulations and their analysis in nonmodel species. I specifically focused on the loblolly pine (*Pinus taeda*, L.), one of the most important forest species in the southern United States both commercially and ecologically (Turner et al. 1995, Smith et al. 2009) and a species with extensive genomic resources, including a reference genome and WES datasets. At ~22 giga base pairs (Gbp), the loblolly genome is one of the largest genomes ever sequenced and assembled (Neale et al. 2014; Zimin et al. 2017). It is also one of the very few genomes thus far available in gymnosperms, the sister lineage of all flowering plants. Importantly, large-scale WES data have been recently generated across multiple loblolly populations from the entire range of this species (Lu et al 2016). These data have led to the discovery of nearly 3 million single-nucleotide polymorphisms or SNPs, a common type of polymorphism, but the presence of CNVs in this dataset has not been assessed. Although CNVs are less common than SNPs, they can account for a significant proportion of the total genomic DNA and are thus an important source of genetic variation.

One of the goals of this research is to develop improved approaches to identify CNVs in the loblolly genome using WES data. This project consists of three parts. In the first part of this thesis, I have developed an error model for the loblolly WES data. Error models are required in order to perform CNV simulations with SECNVs, a program specifically developed to operate using real genome assemblies and generate both CNVs and reads simulating WES datasets (Xing et al. 2020). In the second part of the research, I have used SECNVs to simulate CNVs based on the loblolly genome sequence and determined the sensitivity and false discovery rate of ExomeDepth (Plagnol et al. 2012), a commonly used CNV detection program, under a variety of parameters. In the third part, I have conducted preliminary analyses to identify CNVs in loblolly WES data using ExomeDepth with optimized parameters for this species.

2. DETERMINING THE ERROR MODEL FOR THE LOBLOLLY WES DATA

2.1. Background

Sequencing errors are relatively common in next-generation and third-generation sequencing technologies (Ma et al. 2019). However, error models are currently unavailable for many sequencing platforms. An accurate error model is a fundamental step in order to simulate CNVs using SECNVs, because this program can simulate SNPs within CNVs. General Error-Model Based SIMulator, or GemSIM (McElroy et al. 2012), is a next-generation sequencing simulator that is compatible with most sequencing platforms, such as Illumina. This software can build the empirically derived error model as well as create either single or pair-ended reads that realistically emulate sequencing runs. In this project, I used GemSIM to build the error model for the loblolly pine sequencing data and generate simulated reads for further use.

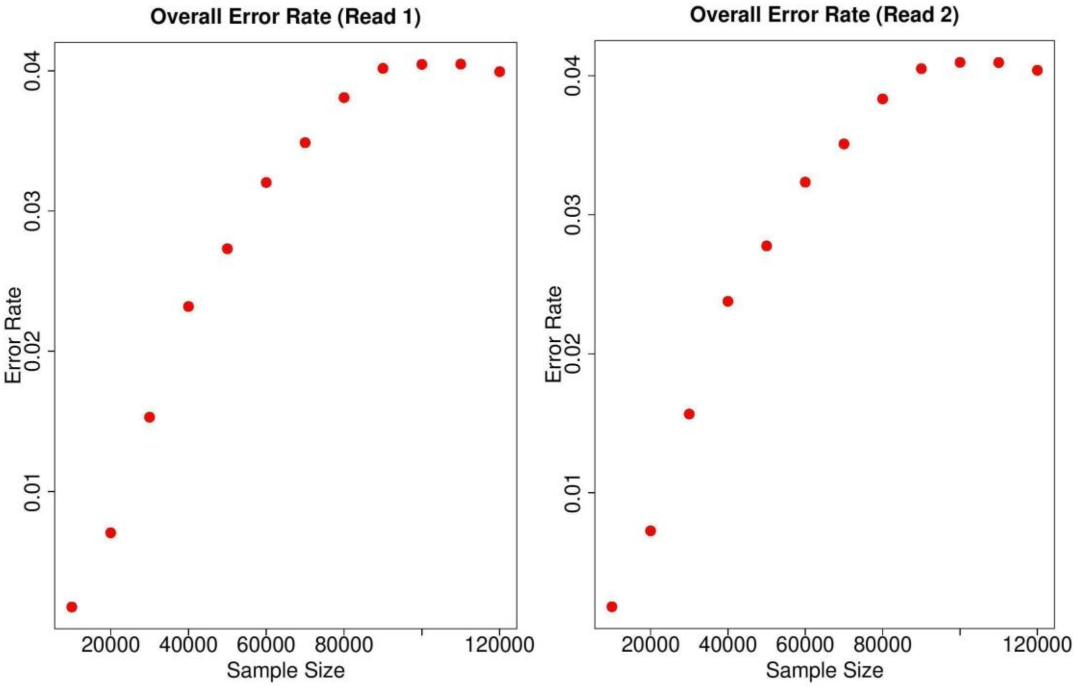
One important caveat of this analysis is that the reads used to calculate the error model were not obtained from the sample (tree) used to build the genome assembly. In the case of loblolly pine, the reference genome has been built on sequencing data obtained from a single tree, named 20-10-10. However, this tree has not been included in the WES dataset used here (Lu et al. 2016). Although other sequencing datasets are available on the Sequence Read Archive (SRA) hosted on the NCBI servers, they do not correspond to the specific sequencing platform (HiSeq 2500) used in the Lu et al. (2016) WES experiment. To correct for this, I modified the sequence of the WES sample used to generate the error model to account for known polymorphisms with the reference genome. As I explain below, this improved error model is still providing higher than expected estimates of sequencing error.

2.2. Materials and Methods

The loblolly pine reference genome applied in this study is PineRefSeq v. 1.01, with an estimated size of ~22 G bp. The DNA for WES analyses was obtained from 374 loblolly pine trees from the ADEPT2 project (Cumbie et al. 2011; Lu et al. 2016). Exons were captured using custom-designed NimbleGen oligonucleotide hybridization probes, and DNA fragments were sequenced using the Illumina HiSeq 2500 platform with a 2x125 bp pair-end sequencing strategy. The designed oligonucleotide covered 90.2% of annotated exons based on the loblolly pine reference genome (Lu et al. 2016). The resequencing reads data, in BAM format, are 1.2T in total. Given the size of this dataset and the fact that all reads were generated in the same sequencing platform, only a fraction of the data were used to determine the substitution error distribution and build the associated error model. Specifically, I used the ten longest scaffolds of the genome assembly (tscaffold2120, tscaffold813, tscaffold4352, tscaffold4850, tscaffold59, tscaffold2259, tscaffold616, tscaffold4938, tscaffold2404, tscaffold2221). The short reads were taken from the sample X001B. To generate the substitution error, GemSIM matches all selected short reads to the associated reference genome and then calculates the error rate for each position along the length of reads. The computational time in this process increases with the size of the reference genome, or number of analyzed scaffolds. A goal of this part of my thesis is to determine the minimum number of reads needed to yield a reliable error model. If the error models approach a distribution with the growth of sample size, and if the difference among the distributions of error rate along the length is negligible when the sample size is greater than a particular value, then adding more samples is meaningless. To this end, I generated a series of sets of reads of increasing size. The sample size ranged from 10,000 to 120,000 with intervals of 10,000 reads. Then, I parsed these sets of reads to GemSIM to build corresponding error models. I also modified the sequence of the 8 scaffolds from the X001B sample to include the variants (SNPs) found in the reference genome to generate an improved error model.

2.3. Results

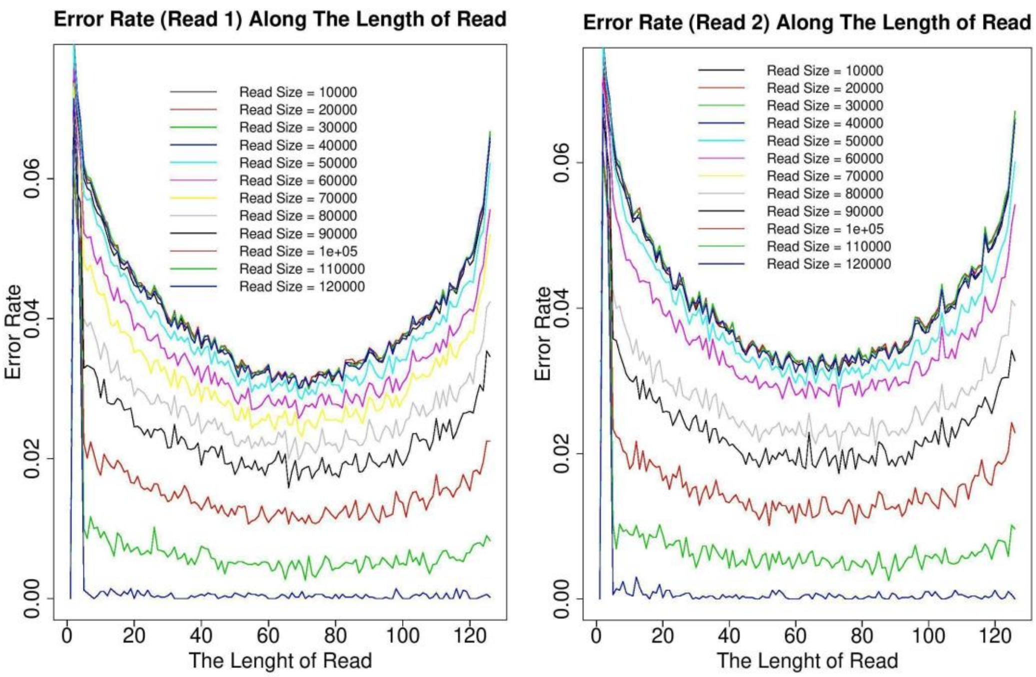
The overall error rate, i.e., the substitution error rate regardless of the position, ranged from ~0.17% to ~4%, as shown in **Fig. 2.1**. The overall error rate increases with the dataset size until 90,000. This may suggest 90,000 reads is a good choice to yield a useful error model.



(A) **(B)**

Figure 2.1 Overall error rate with different sample sizes. (A) Overall error rate for the forward reads. (B) Overall error rate for the reverse reads.

Since the shorts reads are pair-ended, we have both forward and reverse reads for each DNA fragment. The results suggest that the differences in overall error rates between the forward and reverse reads are very small (less than 0.06%). This implies that the forward and reverse reads have very similar accuracy for each sample set. The error rate is high at both ends of the reads and relatively low toward the middle, as shown in **Fig. 2.2**. E.g., the error rate in the middle can be as low as 3%, or less than one half of the error rate at both ends, when using 10,000 reads. The difference between the error rates at the ends and the middle part decreases with the sample size increasing (**Fig. 2.2**).



(A) **(B)**

Figure 2.2 Error rate along the length of reads. (A) The error rate along the length of forward reads. (B) The error rate along the length of reverse reads.

As a whole, these error rates are much higher than the reported overall error rate of ~0.1%. This is likely due to the fact that the reads used in these analyses contain SNPs compared to the reference genome (see **Background**). After removing these sites and re-analyzing the data with GemSIM and limiting the analysis only to reads that map concordantly to the assembly, I obtained an improved error model (**Table 2.1**).

Table 2.1 Overall error rate with and without SNPs. "1" represents the read 1 (forward read). "2" represents the read 2 (reverse read). "e SNPs" means taking into account SNPs (exclude SNPs). "i SNPs" means not taking into account SNPs (include SNPs). "# reads" shows the number of reads of each scaffold.

Scaffold	1_e_SNPs	1_i_SNPs	2_e_SNPs	2 _i_SNPs	# reads
tscaffold2120	0.0116	0.0129	0.0119	0.0131	26394
tscaffold2221	0.0310	0.0327	0.0306	0.0322	46669
tscaffold2259	0.0066	0.0070	0.0068	0.0071	18497
tscaffold2404	0.0133	0.0153	0.0134	0.0154	29374
tscaffold4850	0.0075	0.0078	0.0075	0.0079	20180
tscaffold4938	0.0142	0.0152	0.0144	0.0153	29161
tscaffold59	0.0106	0.0108	0.0104	0.0106	22345
tscaffold813	0.0130	0.0136	0.0133	0.0138	26731

The SAM files of two scaffolds (tscaffold4352, tscaffold616) were not analyzed. As shown in **Table 2.1,** the overall error rates range from 0.0066 in tscaffold2259 to 0.0310 in tscaffold2221 (0.66-3.1%) and are less than the corresponding overall error rates if including SNPs. The error rate was consistent across forward and reverse reads for each scaffold.

2.4. Discussion

Determining the error model for the WES data was a necessary step in order for SECNVs to generate SNPs in simulated CNVs. However, I did not use this specific feature of SECNVs to simulate CNVs. Thus, the accuracy of the error model did not affect the subsequent analyses in this project. Nevertheless, the error model analysis provided some important results. First, I found that the error estimates plateau when

90,000 reads are used to generate the model, a result that will help developing error models in this species. Second, the error rates were much higher than expected in all scaffolds, and on average more than 10 times higher than the expected 0.1%. Third, the error distribution along the read length mirrored what found in other studies, for example by Ma et al. (2019). Interestingly, there was up to a nearly 5-fold difference in the error rate between scaffolds. However, this is likely due to the small number of reads per scaffold, and it should not been considered a reliable assessment of the actual variation in error rate between scaffolds. Further analyses are warrant to determine the actual variation in error rate due to SNPs and between scaffolds.

3. USING SIMULATIONS TO OPTIMIZE THE PARAMETER OF A WIDELY USED CNV DETECTOR, EXOMEDEPTH, TO IMPROVE THE ACCURACY OF CNV DETECTION IN LOBLOLLY PINE

3.1. Background

The majority of CNV detectors were originally designed for analyzing the human genome for the purpose of identifying CNVs associated with disease. Differences in exon length, intron length, GC content and other factors that change between genomes both locally and genome-wide can affect the accuracy of these programs. Simulations allow us to test how these factors affect CNV detection. Some studies of evaluating CNV detectors' performance by using simulated CNVs on WES data have been conducted. However, most of these researches are performed on the human genome, especially for studying cancer. For instance, Zare et al (2017) used simulated data to evaluate the performance of ADTEx, CONTRA, cn.MOPS, ExomeCNV, VarScan 2 in terms of sensitivity and specificity. There are few studies comparing CNV detection tools' performance on nonmodel organisms.

To simulate CNVs from WES data, I used the recently developed SECNVs program (Xing et al. 2020). SECNVs is highly customizable, allowing users to modify the number, length distribution, position distribution, and many other features of simulated CNVs. Additionally, SECNVs generates sequencing reads corresponding to the simulated CNVs, excluding deletions. Short reads, in the format of fastq files, and associated sequence alignment, in the format of BAM files, are both generated by

SECNVs. CNVs simulated by SECNVs have been analyzed using three CNV detection programs - ExomeDepth, CANOES and CODEX (Xing et al. 2020). In Xing's research, ExomeDepth shows a better performance on detecting simulated CNVs on both human and mouse genomes. Therefore, I expected that ExomeDepth would perform better than the other two tools on detecting CNVs in the loblolly genome.

Also, ExomeDepth was chosen to identify CNVs in loblolly because it applies an effective method to control for technical variation. Unlike most other CNV callers that compare the read depth of the test sample with the reference genome directly, ExomeDepth constructs a reference set instead of the reference genome to base the CNV inference on. More specifically, ExomeDepth first calculates the correlation between the test sample and all the rest samples to find which one or ones are highly correlated with the test sample. Then ExomeDepth constructs a reference set by using that sample or combination of those samples. All subsequent analyses are based on this reference set rather than the reference genome.

ExomeDepth applies the hidden Markov model (HMM) to detect CNVs. Hidden Markov model is widely used to build a probabilistic model of linear sequence labeling problems emerging in biology. Roughly, the process initiates from the first exon by labeling it as normal, duplication, or deletion according to its read depth and expected read depth, as well as the label of the exon that just preceded it (except the first exon which does not have an exon before it). The same procedure is then reiterated exon-by-exon until the last one in the dataset. A critical parameter that must be considered is the transition probability for CNVs detectors applying HMM, such as ExomeDepth. The transition probability is the probability of transition from one state to another one. For example, if the probability of transitioning from normal to duplication is p, when one exon is labeled normal then the probability of the next exon being duplication is p. There is no ideal transition probability p that is suitable under any conditions. My goal, thus, was to explore ExomeDepth's behavior and evaluate ExomeDepth's performance under different conditions in order to determine an appropriate transition probability, or multiple transition probabilities for deletions and duplications, in loblolly pine genome.

In this study, I have investigated several aspects regarding ExomeDepth's performance under different conditions. First, I tested if ExomeDepth's performs differently when more samples are used to construct a reference set. For example, if there are 100 samples in total except the test sample, is it better to use all of these 100 samples to construct a reference set than to use only part of them? More samples imply more time and computational power that would need to be used in the analysis. It is important to consider the time-accuracy tradeoff associated with using a large number of samples to build a reference set.

Second, I tested if ExomeDepth's performance depends on the number of simulated CNVs given a fixed amount of DNA. This was important to determine given that the frequency of CNVs may change between species and regions of the genome in the same species and the transition probability of ExomeDepth reflects our prior belief of the frequency of CNVs. Currently there is no information concerning the genome-wide frequency of CNVs in loblolly. Thus, understanding ExomeDepth's behavior under both high and low frequency of CNVs would help choosing a better transition probability.

Third, I investigated if read depth, that is the number of reads per bp, affects ExomeDepth's performances. One expectation is that a higher coverage would increase the false discovery rate in deletions, because of spurious mapping of reads in regions that are deleted in the analyzed sample compared to the reference genome. Another possible outcome of higher read depth is an increase is sensitivity of duplications, because more reads are expected to generate a stronger duplication signal. Additionally, this analysis would inform on the sensitivity of CNV detection given the actual read depth of the WES data from Lu et al. (2016).

3.2. Methods

Simulations are often run on part of a dataset, particularly large ones such as the entire loblolly genome. In this section of my thesis, I used the 10 or the 100 longest scaffolds of loblolly. To increase the accuracy of this analysis, I performed 20 to 30 repetitions for each experiment. Due to the limited computational power and time, the repetitions were performed using the following strategy. First, a large number of simulations were independently generated from the same dataset using identical parameters. One of these simulations was randomly chosen as the test sample and part or all of the remaining simulations were used to construct a reference set. I defined each of these experiments as a "repetition". Then, I reiterated this process by changing the tested simulations and using again part or all of the remaining simulations to build a reference set. Therefore, if I started with a 100 simulations, I could obtain up to 100 repetitions if I used all remaining simulations to construct a reference set.

Read depth is a key feature of sequencing data and plays a crucial role in detecting CNVs. However, we can only determine the total number of reads generated from the target regions when using SECNVs. Also, the actual number of reads that can be aligned with the reference genome may be significantly less than the number of reads generated from SECNVs. Therefore, it is necessary to calculate the actual coverage for further analysis. The coverage is calculated as follows: ExomeDepth would calculate the read depth for each target region. First I calculated the total number of nucleotides mapped to each target region by multiplying the read depth of each target region by the length of the target regions. Then I added these numbers up over all target regions. Finally, I divided this sum by the total length of the target regions. The result is the mean coverage for one simulation.

The following experiments were set up to determine ExomeDepth's performance under the conditions explained in the Background section:

1) **Number of samples used to build a reference set.** Testing how the number of samples used to build a reference set affects ExomeDepth's performance. I compared ExomeDepth's performance using 30 or 373 simulations to construct a reference set. For this experiment, I generated 100 CNVs on 10 longest scaffolds and 100,000 paired-end reads for each simulation.
2) **Number of CNVs.** In this test, I compared ExomeDepth's performance using datasets of 100 CNVs and 1,000 CNVs simulated on the 100

longest scaffolds and using 1,000,000 paired-end reads per simulation. Thirty-one simulations were generated in both CNV sample sizes.

3) **Coverage.** In this experiment, I compared ExomeDepth's performance using 50,000, 200,000 and 1,000,000 paired-end reads per simulation, obtained from 100 CNVs simulated on the 10 longest scaffolds. Thirty-one simulations were generated for each case.

All the experiments above shared the following SECNVs settings: the distance between each pair scaffolds of at least 1,000 bp; the CNVs must overlap with target regions for at least 50 bp; the quality score offset for short reads simulation was equal 33; no gaps ("Ns") on the loblolly genome was replaced; no SNPs were simulated; the range of CNVs' length was 50-10,000 bp; the mean fragment length to be generated was 300 bp; 125 bp paired-ended reads were generated; the error model was the one built from tscaffold2259, which has the lowest overall error rate. The two main criteria I considered to determine ExomeDepth's performance were false discovery rate (FDR) and sensitivity. False discovery rate is a key criterion in statistical hypothesis tests. It is the expected proportion of detected CNVs that are false. For instance, if a statistical procedure has FDR of 0.05 and 100 individuals are detected positive, then there would be less than 0.05×100=5 individuals, on average, are false positives. Sensitivity is the proportion of actual positives that are correctly identified. For instance, if the number of actual positives is 100 and 60 of them are correctly identified, then the sensitivity is 60/100 = 0.6. In practice, we need to balance between false discovery rate and sensitivity, i.e., FDR-sensitivity tradeoff. If we require the FDR to be very low, the sensitivity tends to be small as well, and vice versa. In scientific studies, the false discovery rate is often required to be less than a designated small value, usually 0.01 or 0.05. One of the goals of this study was to determine the transition probability at which ExomeDepth reaches the highest sensitivity for CNV deletions and duplications at FDR<0.05.

Here, I approximated the true false discovery rate and sensitivity by using the empirical FDR and sensitivity respectively. More specifically, let's define Q to be the proportion of the false discoveries among all discoveries. That is, $Q = V/R$, where V is the false positives and R is the total number of detected CNVs in our case. FDR then can be defined as $FDR = E[Q]$, where E is the expectation. We use $\widehat{FDR} = 1/n(\sum_{i=1}^{n} Q_i)$ to estimate $FDR,$ where n is the number of repetitions, the sum is over i, i takes value from 1 to n, and $Q_i$is the Q for the ith repetition. Similarly, let's define S to be the proportion of truly detected positives among actual positives. In symbols, $S = T/P$, where T is the number of correctly detected positives, p is the number of actual positives. We use $\hat{S} = 1/n(\sum_{i=1}^{n} S_i)$ to estimate S, where n is the number of repetitions, the sum is over i, i takes value from 1 ton, S_i is the S for the ith repetition. Empirical false discovery rate for deletion/duplication and empirical sensitivity for deletion/duplications are calculated in the same manner. In the following analysis, when we say false discovery rate or sensitivity, we mean our empirical false discovery rate and empirical sensitivity defined above.

My preliminary study shows that the appropriate transition probability for ExomeDepth may be around 0.3. Therefore, I investigated ExomeDepth's performance when the transition probability ranges from 0.2 to 0.4 by 0.01 increments. Since it is highly unlikely that a truly detected CNVs will share the exact same start and end coordinates with an actual CNV, I defined a detected CNV as a true positive (truly classified as a CNV) when this CNV had any overlap with an actual CNV.

3.3. Results

The analysis of simulated CNVs with ExomeDepth showed that, overall, both false discovery rate and sensitivity positively correlate with the transition probability. This is in line with expectations, because as the transition probability increases, ExomeDepth tends to label more target regions as CNVs. As a result, it is expected that more actual CNVs are identified correctly. Also, since the number of actual CNVs remained constant, it was more likely for a region to be misclassified as a CNV.

3.3.1. Comparison of different reference size

The results of the test comparing different numbers of samples used to build a reference set are shown in **Figs. 3.1** and **3.2**. Overall, using a larger number of samples does not lead to increased sensitivity, while it is associated with a slightly higher FDR. As expected, sensitivity is much higher for deletions than duplications. In general, ExomeDepth detects deletions rather accurately regardless of the reference size, with a sensitivity well above 0.7 at the highest transition probability (**Fig. 3.1**), and an FDR always well below 0.05 (**Fig. 3.2**). Precision, which is defined as 1 - FDR, is shown along with sensitivity. The sensitivity converges to around 0.75 when the transition probability is greater than 0.33. This implies that there would be no benefit in sensitivity from an increase in the transition probability beyond this value.

Conversely, the sensitivity curve for duplications showed a steep increase after transition probabilities around 0.23, but the highest sensitivity at FDR<0.5 was reached at transition probabilities of 0.28 and 0.29 for reference sets of 373 and 30 samples, respectively, corresponding to the detection of only about 1/10 of duplications. When transition probability is less than 0.28 or 0.29, no duplications are detected for one or more repetitions in either case. In this study, if no duplication is detected even in one repetition for a specific transition probability, we would not calculate the false discovery rate for that transition probability since the sample size changes. This principle is also applied to calculate false discovery rate for duplication, sensitivity, etc.

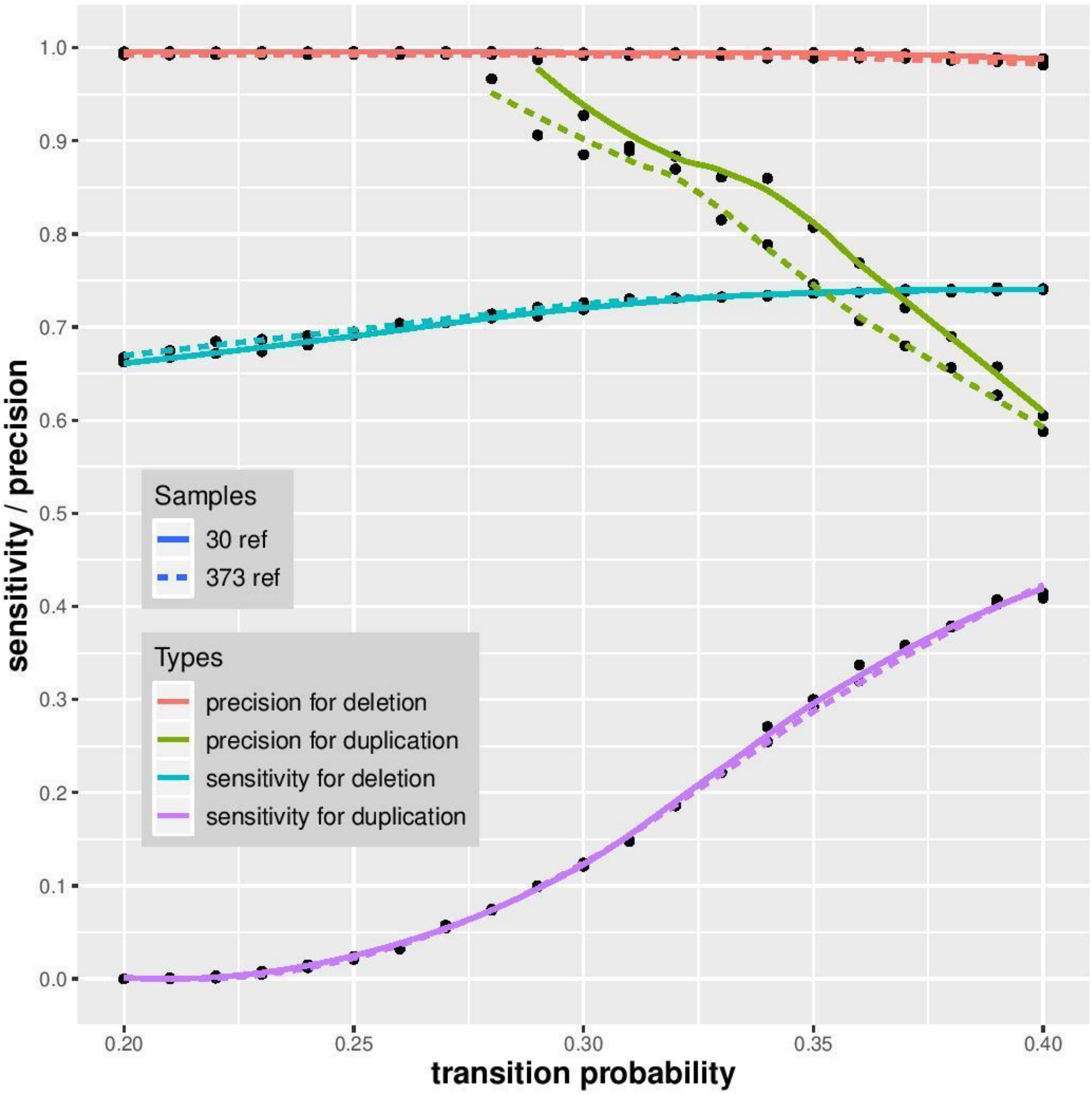


Figure 3.1 ExomeDepth's sensitivity and precision of using different numbers of simulated samples to construct a reference set.

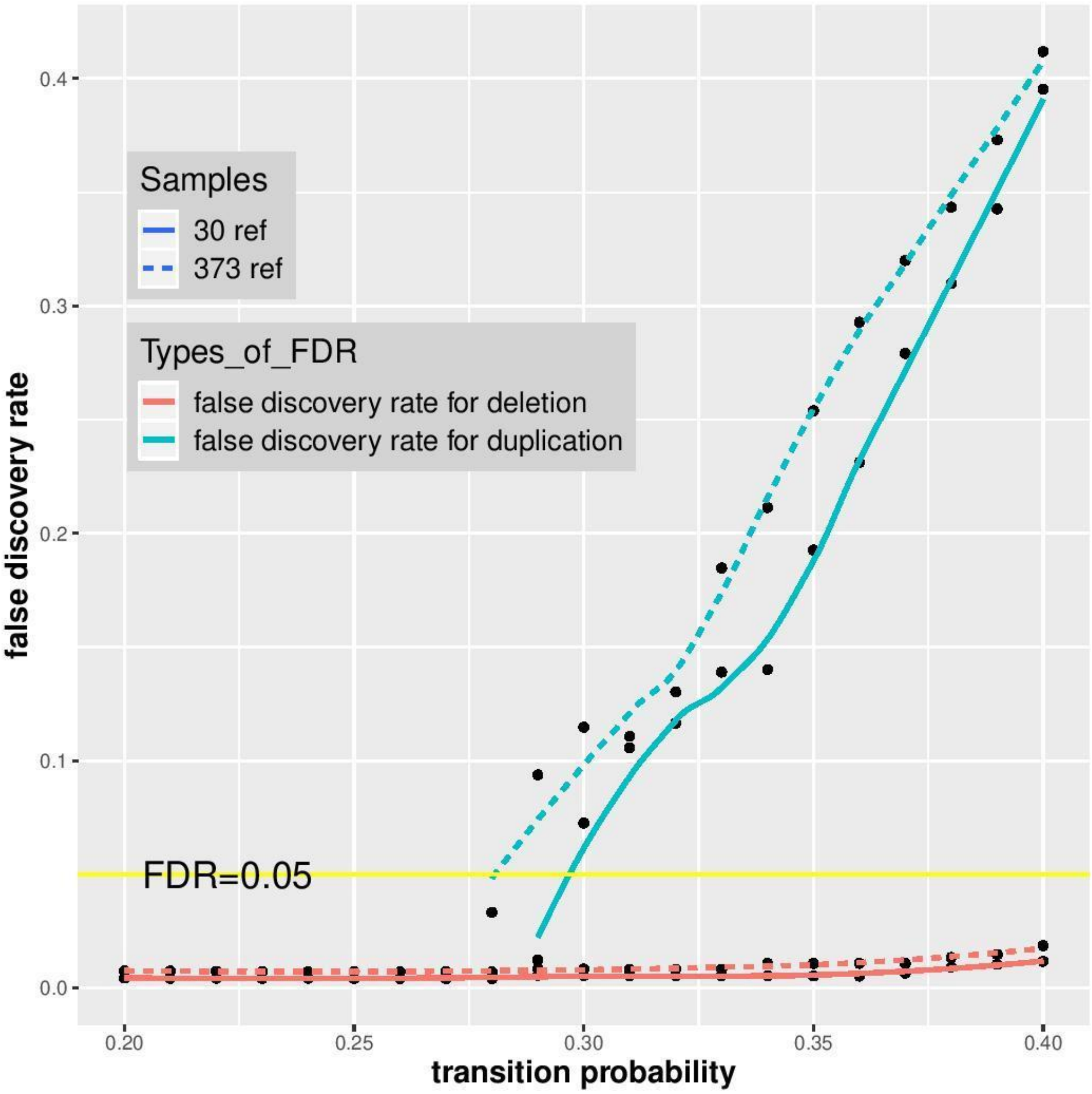


Figure 3.2 ExomeDepth's false discovery rate of using different numbers of simulated samples to construct a reference set.

3.3.2. Comparison of different CNVs size

The ExomeDepth's performance on 100 and 1000 simulated CNVs were significantly different. Notably, the sensitivity was much higher in simulations of 100 CNVs for both deletions and duplication (**Fig. 3.3**). The deletion and duplication sensitivity curves followed closely the trends shown in the analysis of different reference sets, the main difference being a stable sensitivity for deletions across transition probabilities **(Fig. 3.3**). Interestingly, the false discovery rate was higher in deletions for the smaller number CNVs, while it was higher in duplications for the 1,000 CNVs simulations (**Fig. 3.4**). In the 100 CNVs simulations, deletion FDR was above 0.05 at any transition probabilities, whereas FDR reached 0.05 at a transition probability of 0.36 for duplications in the 100 CNV simulations set, which corresponds to sensitivity slightly above 0.25.

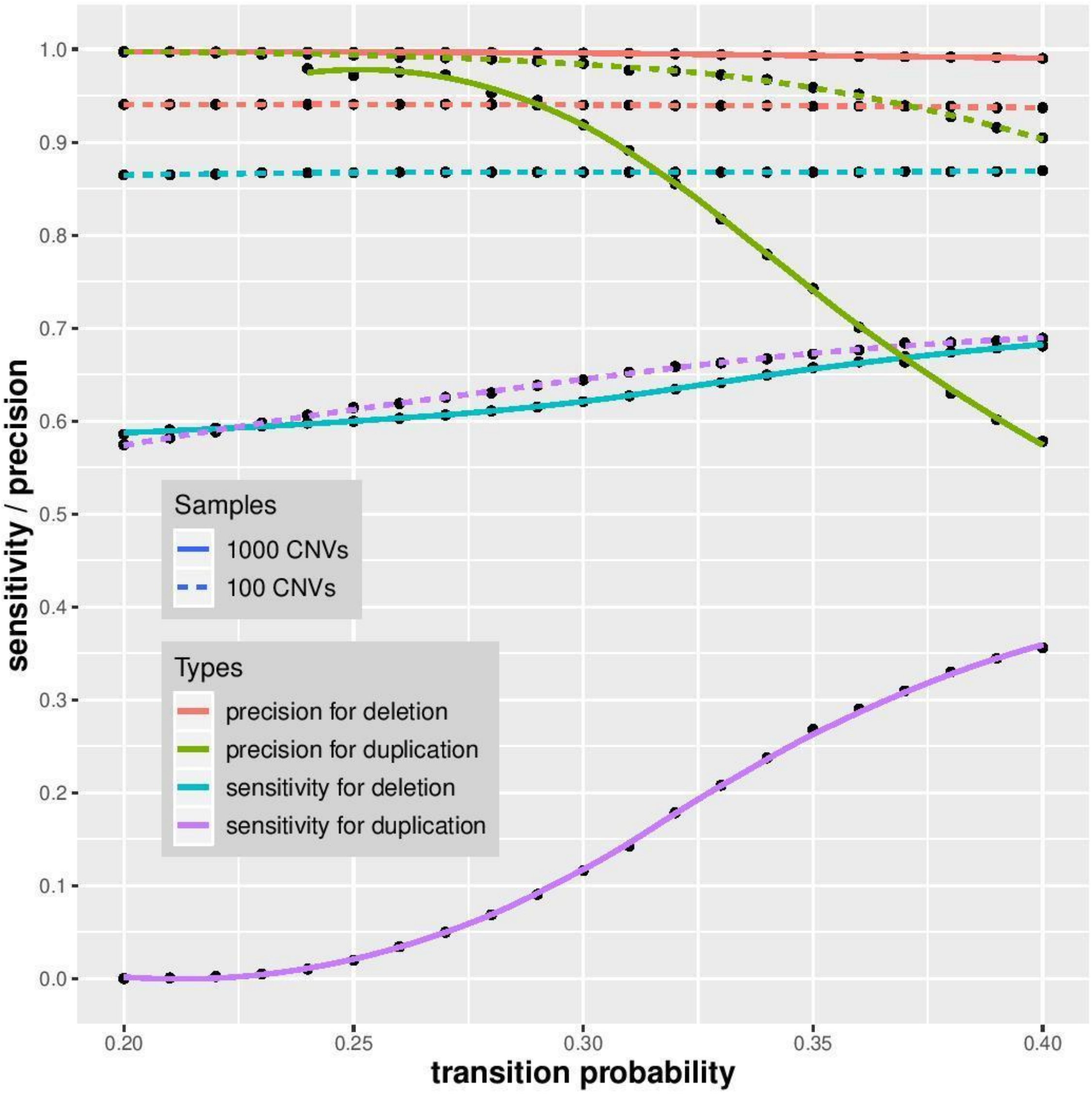


Figure 3.3 ExomeDepth's sensitivity of using different numbers of simulated CNVs on the 100 longest scaffolds.

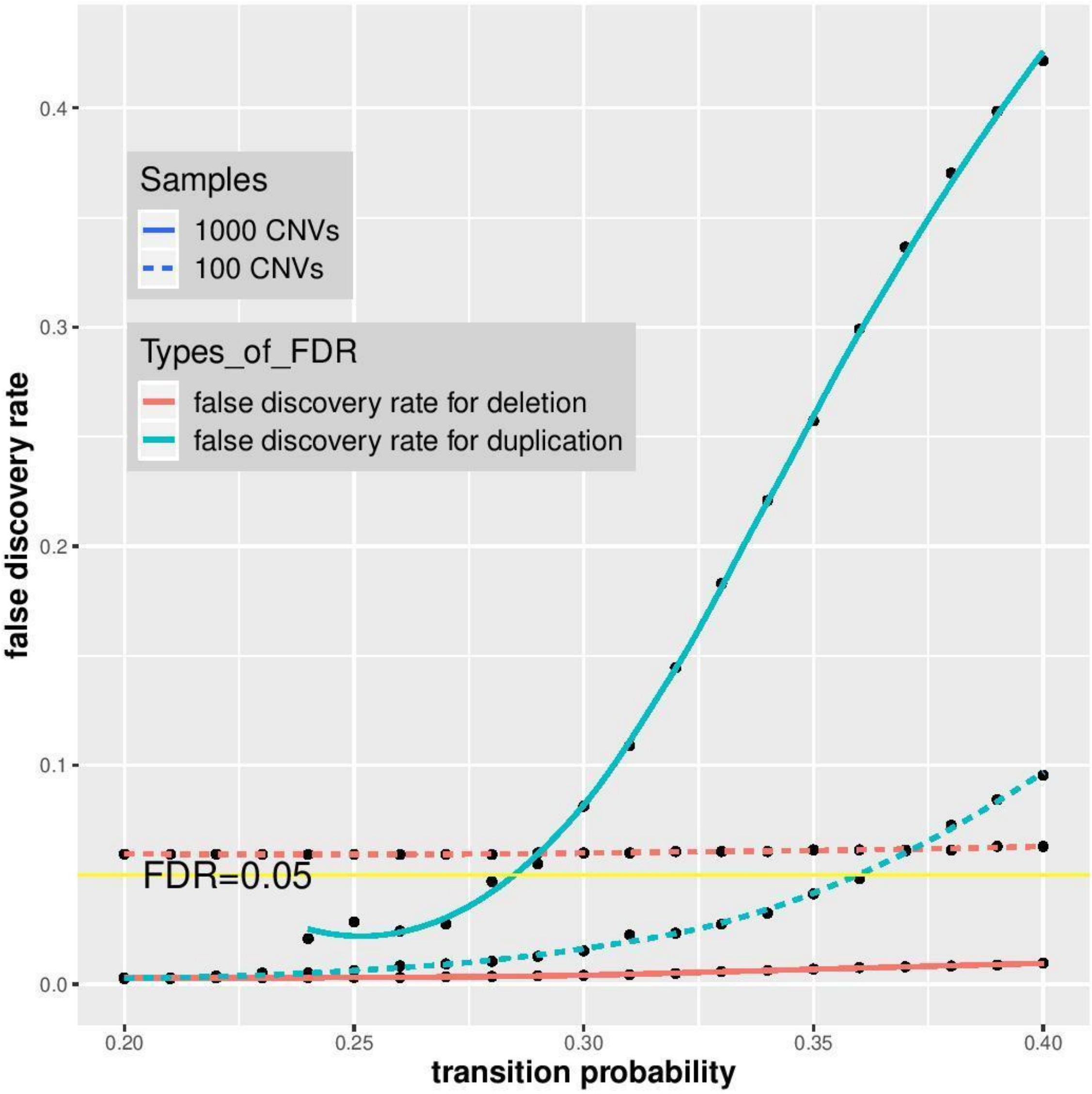


Figure 3.4 ExomeDepth's false discovery rate of using different numbers of simulated CNVs on the 100 longest scaffolds.

3.3.3. Comparison of different read depths

In this test, I simulated 100 CNVs on 10 longest scaffolds and generated 50,000, 200,000 and 1,000,000 reads respectively. The coverage provided by these simulated reads was approximately 61, 148, and 259, respectively. The overall trends of sensitivity curves largely mirrored those of previous experiments (**Fig. 3.5**). Different read depths provided a very similar level of sensitivities across transition probabilities, although the highest sensitivity was reached with the highest read depth for deletions and the middle read depth for duplications. The FDR remained well below 0.05 for deletions under all read depths, whereas it increased substantially with read depth for duplications for increasing transition probabilities (**Fig. 3.6**). The highest transition probability for duplications leading to a FDR < 0.05 was 0.30 for the lowest read depth, which corresponded to a sensitivity of 5-10%.

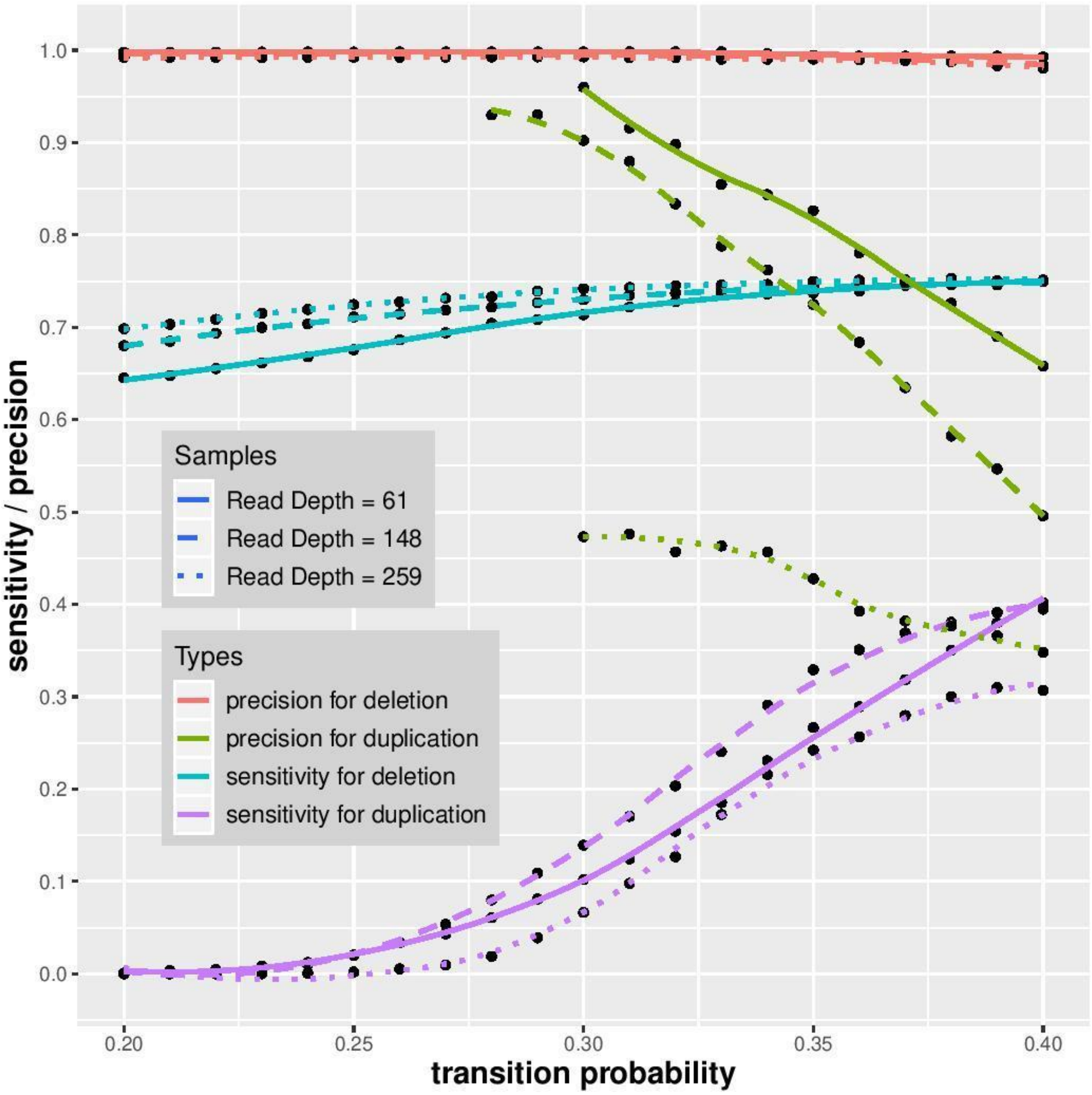


Figure 3.5 ExomeDepth's sensitivity under varying sequencing coverage.

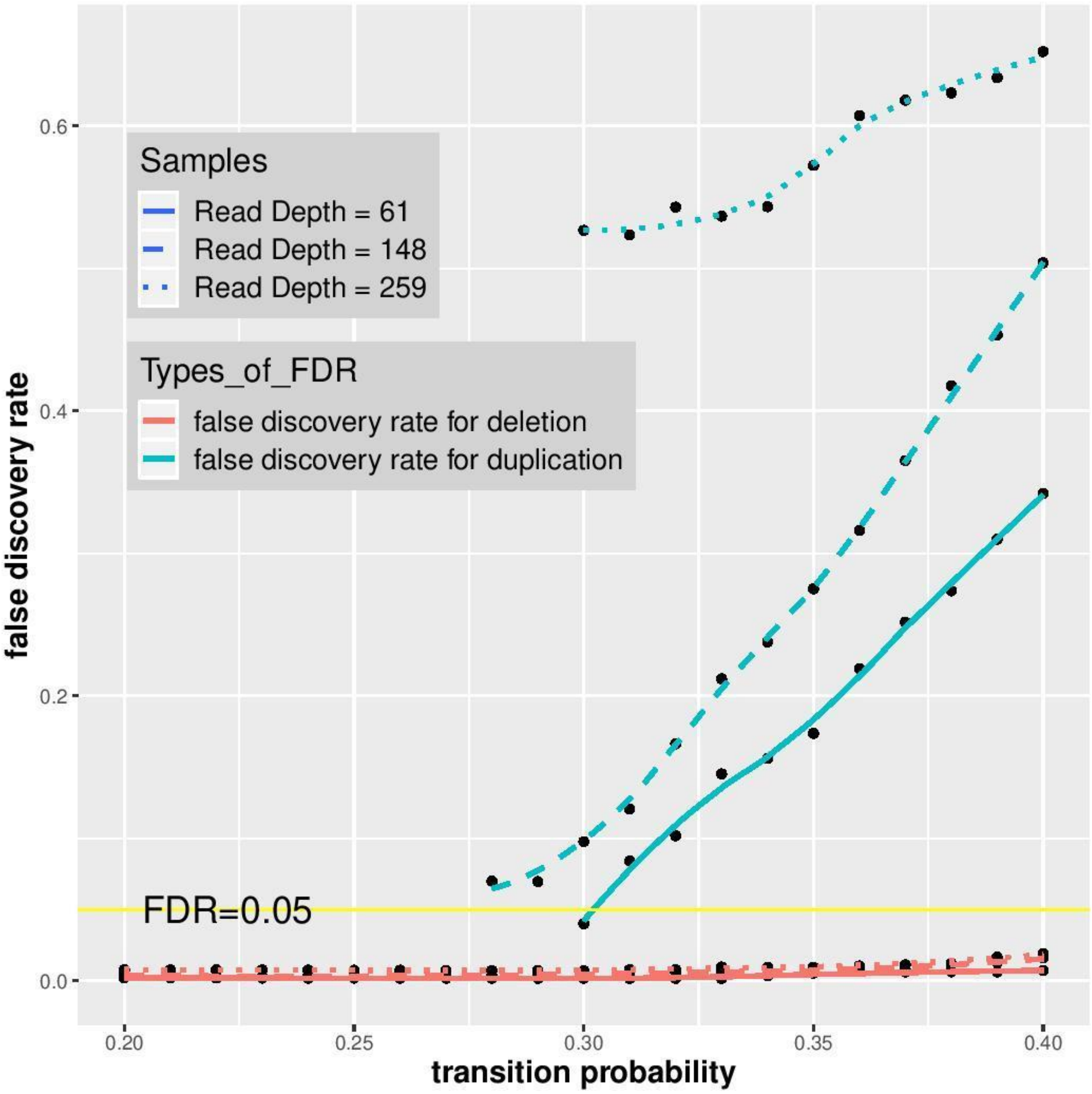


Figure 3.6 ExomeDepth's false discovery rate under varying sequencing coverage.

3.4. Discussion

Overall, I found that ExomeDepth shows high sensitivity and low FDR for deletions, whereas both sensitivity and precision were low for duplications. Thus, the detection of CNVs due to duplications has low reliability. Transition probability, for ExomeDepth, is the probability of transitioning from normal to either duplication or deletion. The range of 0.2-0.4 I used in my thesis seems very high, but this transition probability is only the prior probability, which reflects our prior belief, rather than the actual probability.

Because both the sensitivity and FDR curves show a sharp increase at similar transition probabilities, the benefit of higher detection rates for elevated probabilities is greatly diminished by the parallel increase in false positives. Increasing the number of samples to construct a reference set does not have advantage over using fewer samples, at least for the simulated datasets. In particular, for a given transition probability, FDR for 30 simulations is lower than for 373 simulations in both deletions and duplications. For sensitivity, using 30 and 373 simulations lead to very similar results so there is no advantage in using 373 simulations. The greatest transition probability that controls the overall false discovery rate less than 0.05 is around 0.32 for 373 simulations and around 0.34 for 30 simulations. Given the fact that ExomeDepth has a very different capability to detect deletions and duplications, using two distinct transition probabilities to identify deletion and duplication separately is recommended. Specifically, transition probabilities for detecting deletions should be around 0.2, because the curves of false discovery rate and sensitivity for deletion reach a plateau when transition probability lies in the interval of 0.2 - 0.4. In the simulations of 100 CNVs versus 1,000 CNVs, sensitivity is much higher in the smaller dataset. Notably, sensitivity above 0.5 is found in the smaller dataset for deletions as well as duplications. This indicates that high levels of recall can be obtained under some conditions. In the 100 CNVs dataset, FDR remains lower than 0.05 for duplication when the transition probability is less than around 0.36. FDR appears to be higher for deletions in the smaller dataset, but remains constantly low at approximately 0.06. Interestingly, FDR remained consistently below 0.05 for deletions in the 1,000 CNVs dataset, while it increases sharply above this value for duplications when the transition probability exceeds 0.28. These results illustrate that both false discovery rate and sensitivity for duplication grows quickly as the transition probability increases from 0.2 - 0.4. ExomeDepth's results differ markedly between the test of the two reference sets using 100 CNVs, and the 100 CNVs dataset of the test concerning the number of CNVs. Two differences in the settings between these tests explain these differences. The first one is that in the reference set test, only 10 scaffolds were used to produce the 100 CNVs, as opposed to 100 scaffolds used in the second test. Thus, the CNVs occurred at a lower frequency in the second test. Second, I required SECNVs to generate 1,000,000 reads in the second test but only 100,000 in the first one. As a result, a direct comparison is not possible between the 100 CNVs of both tests. Therefore, ExomeDepth's capability of detecting CNVs may change significantly when the frequency of CNVs are different. This poses a potential issue to the detection of CNVs in loblolly, where the frequency of CNVs is unknown. It also exposes the potential problem of a significant variation in ExomeDepth's performance between genomic regions of high and low frequencies of CNVs.

Given the results of my research, the main challenge to identify CNVs on all loblolly WES samples is the computing time required to perform ExomeDepth analyses. It takes 10-20 hours to run ExomeDepth on 100 scaffolds and 60-80 hours on 431 scaffolds using 30 samples to construct a reference set. However, it takes less than 1 minute to run on 10 scaffolds using 30 samples to construct a reference set. Therefore, the run time appears to increase exponentially with the number of scaffolds. It may not be feasible to run on all 31,044 scaffolds and 374 samples unless some optimization approaches are developed. A possible solution will be to focus on CNVs equal to or larger than 1kb, similarly to what has been done in humans and other species.

4. DETECTING CNVS IN LOBLOLLY USING EXOMEDEPTH WITH OPTIMIZED PARAMETER

4.1. Background

Gene duplications and deletions occur at high frequency in loblolly and other conifers (Casola, Koralewski 2018). A few studies have shown the adaptive role of CNVs in conifers (Hall et al. 2011). CNVs have been detected based on WES data in loblolly in a single study, where only presence/absence variants were identified (Neves et al. 2014). Array based analyses of CNVs have been carried out in at least another conifer (Prunier et al. 2017). However, there is no information on how reliable CNV detection programs are on loblolly WES data. Additionally, no other WES dataset is as comprehensive both in terms of sampled trees and sequencing coverage as the one obtained by Lu et al. (2016). The goal of this part of my thesis is to identify CNVs in this dataset using ExomeDepth with a loblolly-optimized parameter.

4.2. Methods

In Section 3, I have found that using more samples (373) to construct a reference genome does not provide an advantage over using fewer samples (30). Therefore, it is not necessary to use all samples to construct a reference set, which greatly accelerates the processing of the samples. However, this could generate biases due to the composition of specific samples and to the population structure of the samples analyzed. In order to decrease the possible influence of these biases, and to increase the accuracy of CNVs detection, I have applied the following resampling workflow. First, 30 samples are randomly selected to construct a reference set, and this step is carried out multiple times from the available 373 samples, with one sample maintained as the sample to test for all these *resampling runs*. Second, CNVs are detected in the tested sample for each repetition. The results of each run are then compared to identify overlap between the detected CNVs. For example, let's assume that in repetition 1, the region with coordinates 100-200 on scaffold A is labeled as deletion; in repetition 2, the region with coordinate 120-200 on scaffold A is labeled as deletion; in repetition 3, the region with coordinate 150-250 on scaffold A is labeled as deletion. In this approach, the final result would consist of a deletion in the region with coordinate 150-200 on scaffold A. I called these overlapping regions between runs *intersections*.

To identify intersections, I used an R package (IRanges) that provides a function to obtain intersected regions of sequencing data. To verify this method, I used the same dataset used to compare ExomeDepth's performance under different reference sizes (test 1) in Section 3. Each set consisted of 100 simulated CNVs on the 10 longest scaffolds of the loblolly reference genome. The transition probability was set from 0.2 to 0.50 with an increase of 0.01. In each resampling run, a simulation was randomly chosen as the tested sample, and 30 other simulations were randomly selected to construct a reference set. This process was repeated 30 times, generating the final results given the intersection of all 30 individual runs.

For the planned analysis of 30 repetition for 30 runs and 31 transition probabilities (from 0.2 to 0.5 with increase of 0.01), the computing memory for intersecting exceeded the available memory of ~80Gb on the Ada cluster at the TAMU High Performance Computing Research facility. As a result, I used 16 repetitions that have intersected regions up to transition probability = 0.40. The false discovery rate and sensitivity were calculated for each run and, in turn, the average false discovery rate and sensitivity for the total 16 repetitions.

Finally, ExomeDepth was used to detect CNVs existing on 10 and 100 longest scaffolds of X001B. The transition probability was set at 0.39 because the false discovery rate for duplication is less than 0.05 when transition probability is less than or equal to 0.39. A total of 30 randomly selected samples were used to construct a reference set. This process was repeated 30 times and the final result was obtained by the intersection of these 30 independent runs.

4.3. Results

4.3.1. ExomeDepth's performance using *resampling runs* and *intersections*

The resampling approach with intersection showed a similar sensitivity for deletions and slightly lower sensitivity for duplications (**Fig. 4.1**). However, precision was significantly higher for duplications when using intersections, and FDR reached 0.05 only at a transition probability of 0.4 (**Fig. 4.2**). At an FDR=0.05 (precision=0.95), the sensitivity for duplications was 0.1 in simulations with no resampling and intersections compared to a sensitivity of ~0.38 when resampling/intersections were applied (**Fig. 4.2**).

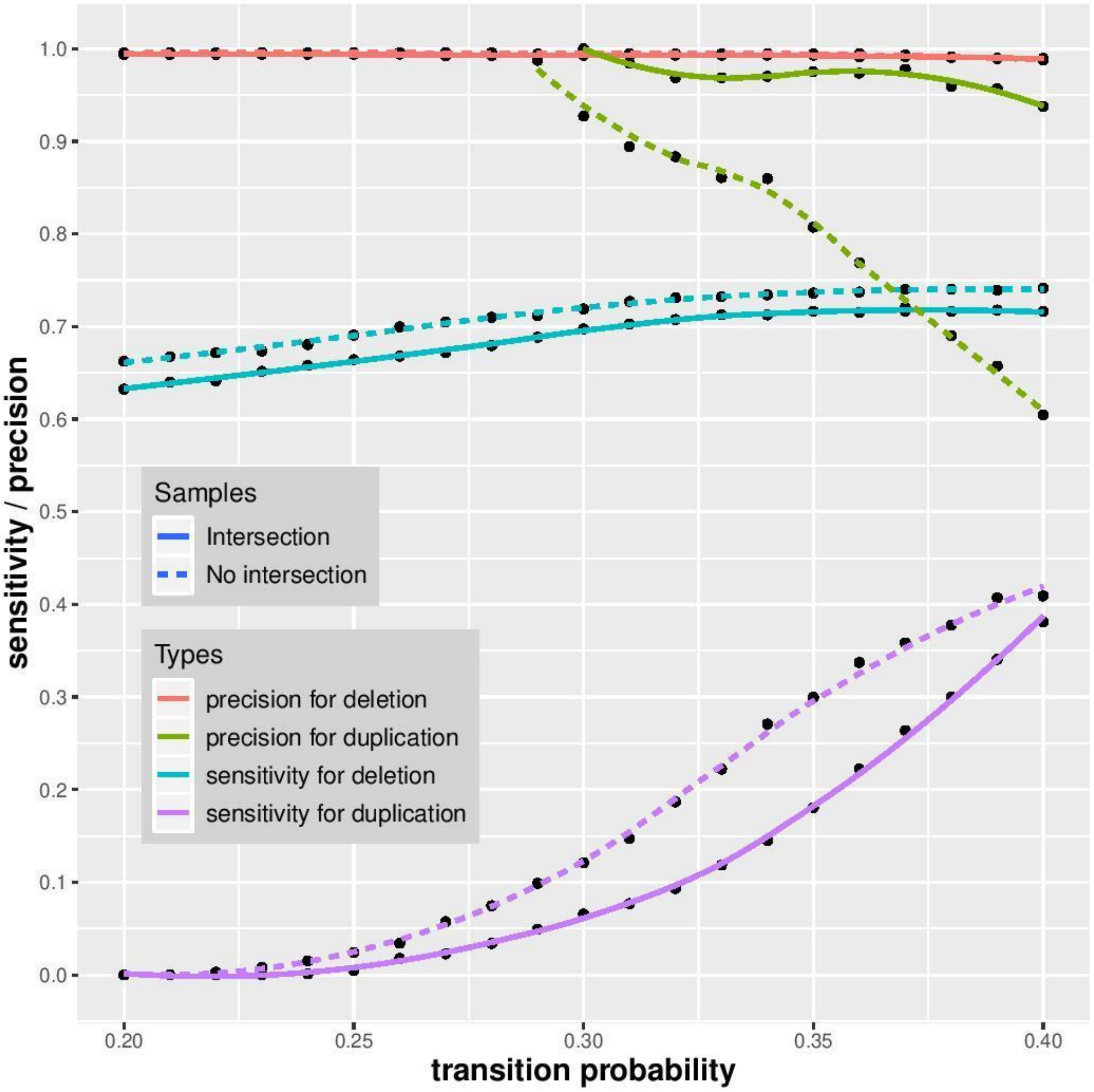


Figure 4.1 ExomeDepth's sensitivity and precision with and without resampling/intersections for 100 CNVs from 10 scaffolds and using 30 samples to build a reference set.

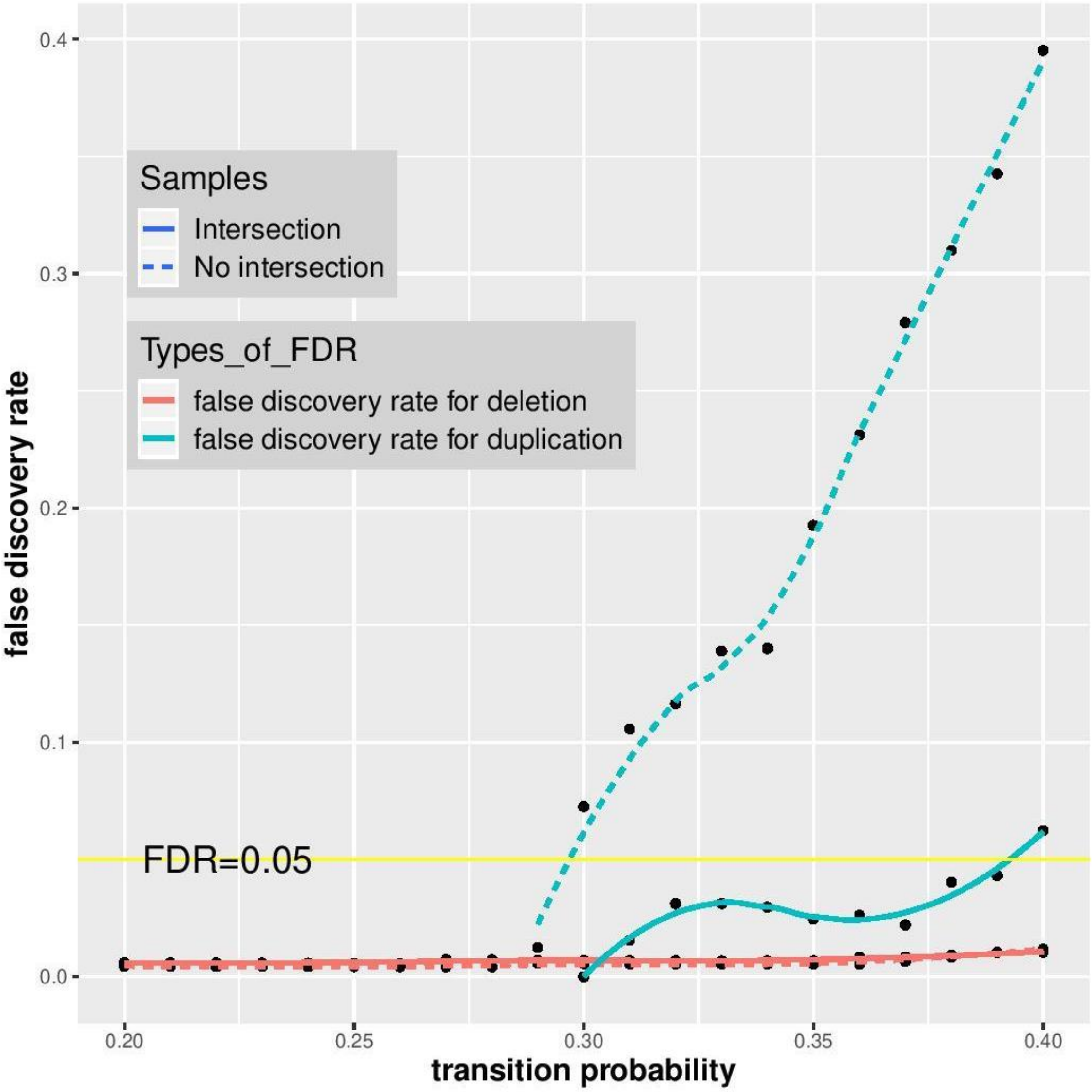


Figure 4.2 ExomeDepth's false discovery rate with and without resampling/intersections for 100 CNVs from 10 scaffolds and using 30 samples to build a reference set.

To determine the limitations of this approach at higher transition probabilities, I performed further analysis increasing the transition probability at intervals of 0.01 up to 0.47. Precision decreased sharply below 0.95 for duplications at transition probabilities of 0.4 and higher (**Figs. 4.3-4.4**). No significant change was found in sensitivity and precision for deletions above transition probability=0.4.

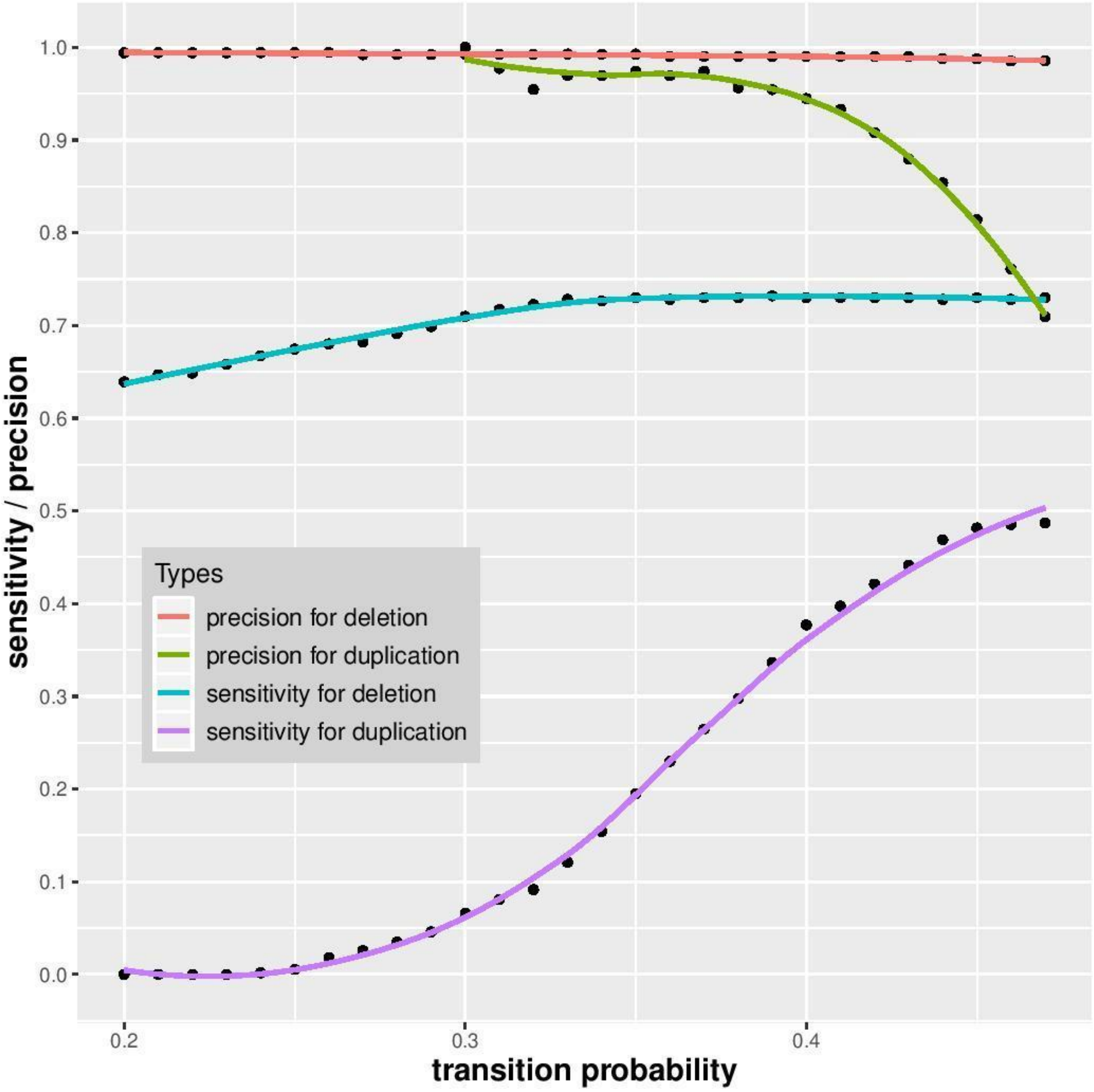


Figure 4.3 ExomeDepth's sensitivity and precision with resampling/intersections at transition probabilities up to 0.47.

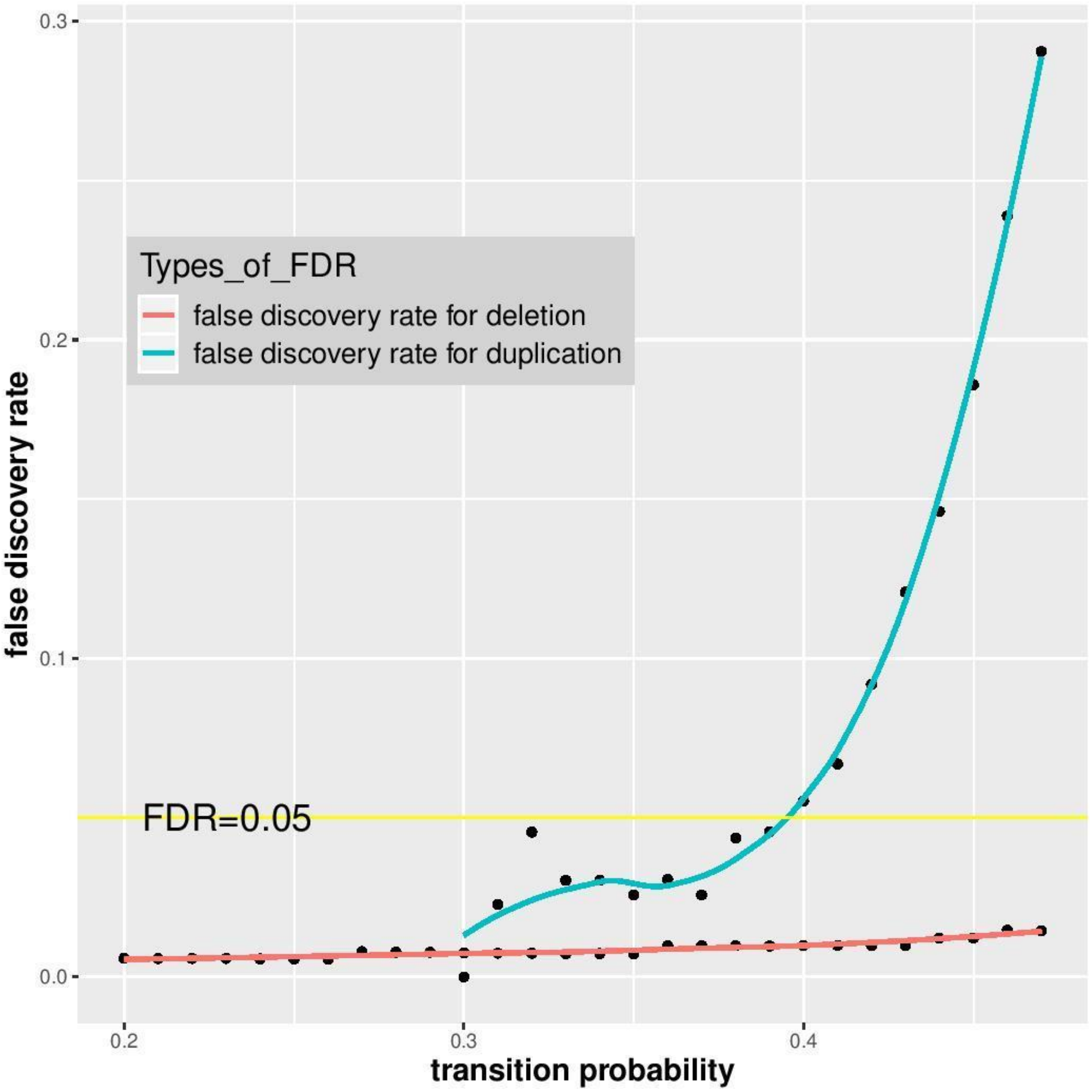


Figure 4.4 ExomeDepth's false discovery rate with resampling/intersections at transition probabilities up to 0.47.

4.3.2. Detecting CNVs existing on 10 and 100 longest scaffolds of X001B

The loblolly exome is distributed across 31,044 scaffolds making up 37,620,106 bp of the genome and containing 1-145 target regions (**Fig. 4.5**). The majority of scaffolds contain less than 20 target regions. I selected the 10 longest scaffolds for the analysis of loblolly WES data. These scaffolds contain 965 exons covering 130,984 bp and are thus informative of the ability of ExomeDepth to capture CNVs.

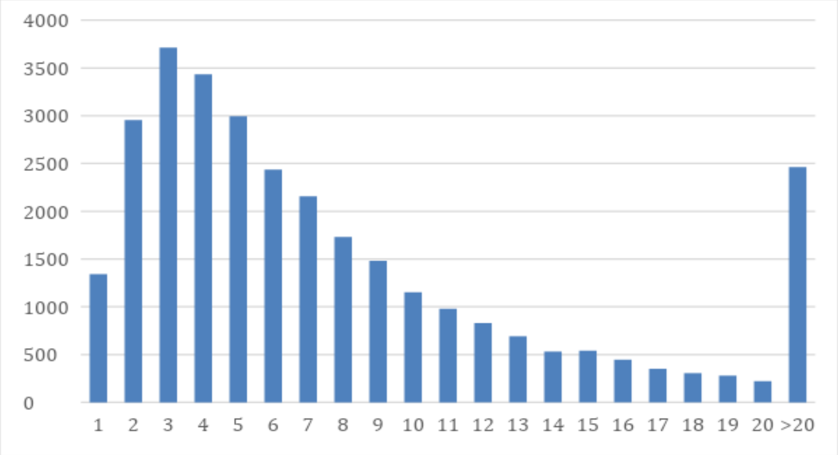


Figure 4.5 Distribution of the number of target regions per scaffold.

A total of 31 CNVs were detected in the 10 longest scaffolds for the X001B sample, divided into 22 deletions and 9 duplications (**Table 4.1**). The average length of deletions and duplications was 1,051 bp and 299 bp, respectively. The longest deletion was ~12,000 bp and the longest duplication was lower than 1,000 bp. Only one scaffold had no CNVs, and 7/9 remaining scaffolds showed both deletions and duplications, with up to 4 deletions and 2 duplications per scaffold. On average, I found 2.4 deletions and 1.3 duplications per scaffold.

Table 4.1 CNVs detected on 10 longest scaffolds of X001B.

CNV type	Scaffold	Start	End	Length
deletion	tscaffold2120	3166733	3167956	1224
deletion	tscaffold2120	4732490	4732665	176
deletion	tscaffold813	2155891	2159997	4107
deletion	tscaffold4352	676735	676959	225

Table 4.1 Continued

CNV type	Scaffold	Start	End	Length
deletion	tscaffold4352	1116387	1116501	115
deletion	tscaffold4352	1118177	1118280	104
deletion	tscaffold4352	1148033	1148369	337
deletion	tscaffold4850	3540530	3552616	12087
deletion	tscaffold4850	5074070	5074159	90
deletion	tscaffold59	649105	649181	77
deletion	tscaffold59	1473353	1473429	77
deletion	tscaffold59	3214794	3214865	72
deletion	tscaffold59	3215254	3215329	76
deletion	tscaffold616	3229817	3231089	1273
deletion	tscaffold616	5355406	5355481	76
deletion	tscaffold4938	4942159	4942481	323
deletion	tscaffold4938	5062768	5064460	1693
deletion	tscaffold2404	1785793	1785875	83
deletion	tscaffold2404	3697073	3697156	84
deletion	tscaffold2221	5429230	5429298	69
deletion	tscaffold2221	5829154	5829263	110
deletion	tscaffold2221	7694368	7695019	652
duplication	tscaffold2120	2770859	2771598	740
duplication	tscaffold813	2154186	2154291	106
duplication	tscaffold4352	1178939	1179186	248
duplication	tscaffold4352	4550896	4551001	106
duplication	tscaffold59	4784470	4784658	189
duplication	tscaffold616	205320	205488	169
duplication	tscaffold616	519992	520068	77
duplication	tscaffold4938	1987302	1987411	110
duplication	tscaffold2404	3641339	3642283	945

I also analyzed in the 100 longest scaffolds for the X001B sample and detected 147 deletions and 92 duplications (**appendix, Table B.1**). The average length of deletions and duplications was 20,288 bp and 12,129 bp, respectively. Median values were significantly lower, with median CNV lengths of 225 bp and 394 bp for deletions and duplications, respectively. Indeed, deletions showed a sharp decrease in frequency with increasing CNV length, whereas duplications had comparable frequencies at 100, 200 and 400 bp (**Fig. 4.6**). A few very long deletions and duplications were also observed, including a notable deletion slightly longer than 944 kb.

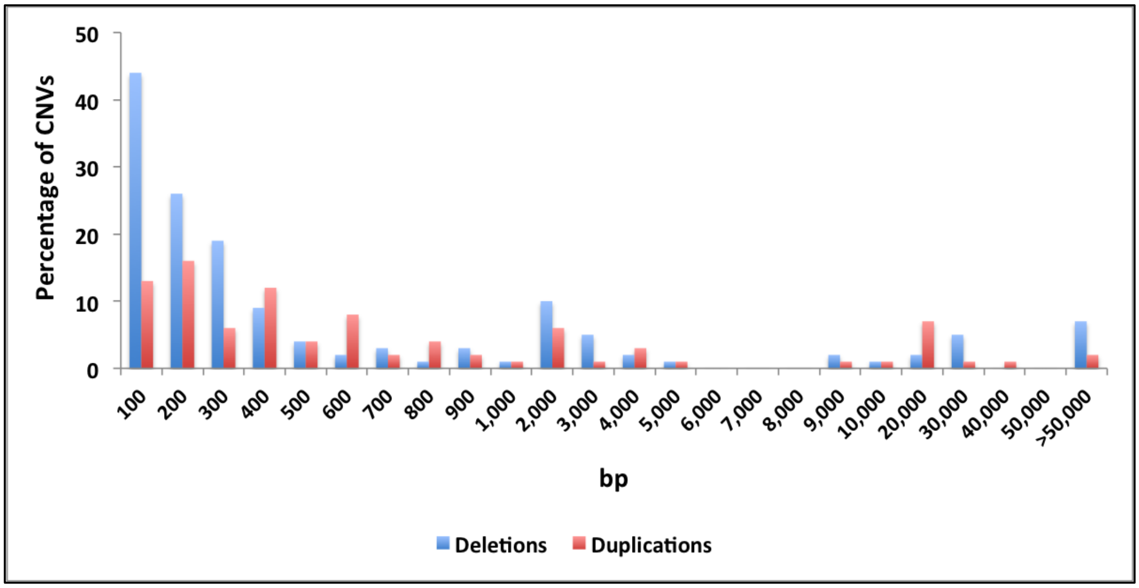


Figure 4.6 Length of detected CNVs on 100 longest scaffolds.

Interestingly, CNVs were detected in only 59/100 scaffolds, with all 59 scaffolds containing deletions and 38 scaffolds containing duplications. On average, there were 2.5 deletions and 2.1 duplications per scaffold, with a maximum of 12 deletions and 5 duplications in a single scaffold.

All the CNVs detected in the 10 scaffolds were present in the 100 scaffolds, with 20/22 deletions and 5/9 duplications matching perfectly, and the remaining CNVs including slightly longer DNA segments in the 100 scaffolds analysis. However, the same 9 scaffolds with CNVs included 7 more deletions and 7 more duplications in the 100 scaffolds results compared to the 10 scaffolds analysis.

4.4. Discussion

The simulation analyses in Section 3 indicated that detecting duplicative CNVs is particularly challenging when using simulated WES data, even with a state-of-the-art tool such as ExomeDepth. This issue is likely to affect the analysis of real exome datasets as well. To circumvent this problem, I have developed a resampling approach to compare multiple runs of ExomeDepth and retain only overlapping CNVs, or intersections. I found that using intersections leads to a significant improvement in the false discovery rate in detecting duplications. As expected, sensitivity was slightly lower with intersections, because fewer and shorter CNVs are detected when they are required to overlap across all resampling runs. However, this drawback is acceptable because the improvement in precision is significant.

The primary limitation of this method is that it may require a high computation memory and generate a large number of files in the entire process. An experiment with 30 repetitions, each repetition with 30 individual runs, and the transition probabilities is set to 0.2-0.5 by 0.01 increments, there will be $30 \times 30 \times 31 = 27{,}900$ files generated in total. Also, computing memory demands increase with the transition probability because more regions are identified as CNVs. However, this high demand should be ameliorated when applying ExomeDepth to real data because repetitions are not needed to compute the average sensitivity and false discovery rate, and the range of transition probabilities can be narrowed down. Given the simulation results, a narrow set of transitions probabilities can be applied in order to analyze real data. Future studies could determine a minimum number of resampling runs that is required to generate reliable intersections.

The analysis of 10 and 100 scaffolds from the same loblolly sample revealed some additional features regarding the performance of ExomeDepth with real datasets. The overlap of detected CNVs between the two analyses seems to suggest that ExomeDepth results are not affected by the number of scaffolds. However, there were a significant additional number of CNVs detected in 9 longest scaffolds when the large scaffold dataset was used. Increasing the number of scaffolds either improve the ability of ExomeDepth to detect CNVs or it leads to a higher false discover rate. Given that FDR for deletions is consistently low across my simulation analyses, it is possible that a larger number of scaffolds increase the sensitivity of ExomeDepth in real datasets. Future studies should determine if the number of detected CNVs keep on increasing by adding more scaffolds or if it tends to plateau.

The cumulative length of all deletions identified in the 100 scaffolds corresponds to ~0.85% of the total DNA found in these scaffolds. The length of all duplications combined is ~0.32% of the 100 scaffolds DNA. This is much higher that what found in human, where an estimate of the proportion of DNA affected by CNVs per individual varies between 0.02-0.06% (Wineinger et al. 2011, Itsara et al. 2009). In those studies, only CNVs longer than 1kb were included. However, after I removed CNVs<1kb from the results of the 100 scaffolds, I found a similarly high proportion of DNA occupied by deletions and duplications. One possible explanation is that some of the longest CNVs are artifacts and represent instead two or more shorter CNVs. The three longest deletions and duplications detected in the 100 scaffold dataset represent ~80% of the DNA within CNVs. CNV breakpoints are notoriously difficult to be correctly identified using WES dataset. Individual CNVs occurring on two separate genes on the same scaffolds or chromosome and sharing similar read depth might be erroneously reported as a single, much longer CNV by ExomeDepth. This possible source of artifacts could be verified by checking if the read depth is uniform across all exons within the boundaries of each CNV. Further, given the size of the loblolly pine, it is possible that large deletions and duplications are more tolerated, especially if they don't overlap many genes. I have not specifically searched for genes within loblolly CNVs in these results for this project, but this should be determined when more extensive CNV data in loblolly will be available. This will depend on developing improved computational approaches to enable the analysis of the complete loblolly WES dataset.

Identifying CNVs in loblolly will be critical to determine the impact of these genetic variants in local adaptation. Phenotypic and climatic data are available for the 374 trees sampled in the WES study (Lu et al. 2016, 2017, 2019). Association analyses between CNVs and these datasets will be important to reveal the possible role of CNVs in local adaptation, similarly to what has been done using SNPs (Lu et al. 2016, 2019).

5. CONCLUSIONS

Detecting CNVs using whole-exome sequencing is a computational and statistical challenge given the currently available detection tools. In this project, I have shown that using simulations provides fundamental information on the pitfalls of commonly used CNV detection programs such as ExomeDepth, and can assist in developing better practices to increase the sensitivity of this investigation while reducing the rate of false positives. This is especially important for duplicative CNVs, which are inherently more challenging to detect even at high sequencing coverage.

With the goal of setting specific guidelines to identify CNVs from WES data, I have first attempted to obtain a specific error model to be implemented in the CNV simulator of choice, SECNVs. While the error model I generated is not accurate, I have delineated a process that can be implemented by other researchers to more accurately estimate error rates in their data before generating simulated CNVs. In this project, such error models are not required because I did not need to include SNPs in the CNV simulations.

Second, I have simulated CNVs under a variety of conditions to test three main factors that can affect the accuracy of the CNV detector ExomeDepth: the number of samples used to build a reference set; the number of CNVs; and the sequencing coverage. I have also explored how ExomeDepth’s transition probability influences sensitivity and precision under these conditions. I have found that ExomeDepth’s accuracy in detecting deletions is generally high and not affected by these variables.

However, ExomeDepth showed a relatively low sensitivity and a high false discovery rate for detecting duplications. Transition probability is the key parameter that determines the performance of ExomeDepth. In general, higher transition probabilities lead to more detected CNVs. As a result, both sensitivity and false discovery rate would increase. When transition probability is higher than some value, the sensitivity tends to plateau, while FDR sharply increases. I have also determined that using a large number of samples (373) to construct a reference set may not improve the performance of ExomeDepth compared to using a relatively small number of samples (30). Additionally, ExomeDepth's performance is better with fewer simulated CNVs, and is not greatly improved by a higher number of reads.

I have found that intersecting the results of reiteratively resampled runs can greatly improve the accuracy of duplications detection with ExomeDepth. This approach led to a nearly 4-fold increase in sensitivity at FDR<0.05. When applied to a portion of the most extensive loblolly pine WES dataset, this strategy has led to the identification of a relatively high number of CNVs. I also found that increasing the number of scaffolds leads to higher sensitivity, although more analyses will be needed to rule out a possible increase in false discovery rates. Improved computational methods must be developed in order to make this approach computationally feasible for the complete loblolly WES data, or for similar datasets in other species.

REFERENCES

Casola, Claudio, and Tomasz E. Koralewski. Pinaceae Show Elevated Rates of Gene Turnover That Are Robust to Incomplete Gene Annotation. Plant J 2018 95(5):862–876., doi:10.1111/tpj.13994.

Cumbie WP, Eckert A, Wegrzyn J, Whetten R, Neale D, Goldfarb B. Association genetics of carbon isotope discrimination, height and foliar nitrogen in a natural population of *Pinus taeda* L. Heredity 2011 107(2):105-14.

Hall DE, Robert JA, Keeling CI, Domanski D, Quesada AL, Jancsik S, Kuzyk MA, Hamberger B, Borchers CH, Bohlmann J. An integrated genomic, proteomic and biochemical analysis of (+)-3-carene biosynthesis in Sitka spruce (*Picea sitchensis*) genotypes that are resistant or susceptible to white pine weevil. Plant J 2011 65(6):936-48.

Hull RM, Cruz C, Jack CV, Houseley J. Environmental change drives accelerated adaptation through stimulated copy number variation. PLoS Biol 2017 15(6):e2001333.

Itsara, Andy, et al. “Population Analysis of Large Copy Number Variants and Hotspots of Human Genetic Disease.” The American Journal of Human Genetics, vol. 84, no. 4, 2009, pp. 550–551., doi:10.1016/j.ajhg.2009.03.008.

Lin BY, Zieve JJ, Dougherty WM, Fuentes-Soriano S, Wu LS, Gilbert D, Marçais G, Roberts M, Holt C, Yandell M, Davis JM, Smith KE, Dean JF, Lorenz WW, Whetten RW, Sederoff R, Wheeler N, McGuire PE, Main D, Loopstra CA, Mockaitis K, deJong PJ, Locke DP, Sharp AJ, McCarroll SA, McGrath SD, Newman TL, Cheng Z, Schwartz S, Albertson DG, Pinkel D, Altshuler DM, Eichler EE. Linkage disequilibrium and heritability of copy-number polymorphisms within duplicated regions of the human genome. Am J Hum Genet 2006 79(2):275-90.

Lu M, Krutovsky KV, Nelson CD, Koralewski TE, Byram TD, Loopstra CA. Exome genotyping, linkage disequilibrium and population structure in loblolly pine (*Pinus taeda* L.). BMC Genomics 2016 13; 17(1):730.

Lu M, Krutovsky KV, Nelson CD, Koralewski TE, Byram TD, Loopstra CA. Association Genetics of Growth and Adaptive Traits in Loblolly Pine (*Pinus Taeda* L.) Using Whole-Exome-Discovered Polymorphisms. Tree Genet Genom 2017 13(57).

Lu M, Loopstra CA, Krutovsky KV. Detecting the Genetic Basis of Local Adaptation in Loblolly Pine (*Pinus Taeda* L.) Using Whole Exome-Wide Genotyping and an Integrative Landscape Genomics Analysis Approach. Ecology and Evolution, vol. 9, no.12, 2019, pp.6798–6809, doi:10.1002/ece3.5225.

Lupski, JR. Genomic Rearrangements and Sporadic Disease. Nat Genet 2007 39(7 Suppl):S43-7.

Ma X, Shao Y, Tian L, Flasch DA, Mulder HL, Edmonson MN, Liu Y, Chen X, Newman S, Nakitandwe J, Li Y, Li B, Shen S, Wang Z, Shurtleff S, Robison LL, Levy S, Easton J, Zhang J. Analysis of error profiles in deep next-generation sequencing data. Genome Biol 2019 14; 20(1):50.

Makino, Takashi, et al. “Genome-Wide Deserts for Copy Number Variation in Vertebrates.” Nature Communications, vol. 4, no. 1, 2013, doi:10.1038/ncomms3283.

McElroy KE, Luciani F, Thomas T. GemSIM: general, error-model based simulator of next-generation sequencing data. BMC Genomics 2012 15;13:74.

Neale DB, Wegrzyn JL, Stevens KA, Zimin AV, Puiu D, Crepeau MW, Cardeno C, Koriabine M, Holtz-Morris AE, Liechty JD, Martínez-arcía PJ, Vasquez-Gross HA, Neves LG, Davis JM, Barbazuk WB, Kirst M. A high-density gene map of loblolly pine (*Pinus taeda* L.) based on exome sequence capture genotyping. G3 2014 Jan 10;4(1):29-37.

Plagnol V, Curtis J, Epstein M, Mok KY, Stebbings E, Grigoriadou S, Wood NW, Hambleton S, Burns SO, Thrasher AJ, Kumararatne D, Doffinger R, Nejentsev S. A robust model for read count data in exome sequencing experiments and implications for copy number variant calling. Bioinformatics 2012 1;28(21):2747-54.

Prunier J, Caron S, MacKay J. CNVs into the wild: screening the genomes of conifer trees (*Picea spp.*) reveals fewer gene copy number variations in hybrids and links to adaptation. BMC Genomics 2017 18;18(1):97.

Smith, BW, Miles PD, Perry, CH, Pugh, SA. Forest Resources of the United States, 2007: a Technical Document Supporting the Forest Service 2010 RPA Assessment. Washington Office, Forest Service, U.S. Dept. of Agriculture, 2009.

Tan R, Wang Y, Kleinstein SE, Liu Y, Zhu X, Guo H, Jiang Q, Allen AS, Zhu M. An evaluation of copy number variation detection tools from whole-exome sequencing data. Hum Mutat 2014 35(7):899-907.

Turner, GP, Koerper GJ, Harmon ME, Lee JJ. A Carbon Budget for Forests of the Conterminous United States. Ecol Appl 1995 5;2421–436.

Wineinger, Nathan E, et al. “Characterization of Autosomal Copy-Number Variation in African Americans: the HyperGEN Study.” European Journal of Human Genetics, vol. 19, no. 12, 2011, pp. 1271–1275., doi:10.1038/ejhg.2011.115.

Xing Y, Dabney AR, Li X, Wang G, Gill CA, Casola C. SECNVs: A Simulator of Copy Number Variants and Whole-Exome Sequences From Reference Genomes. Front Genet 2020 21;11:82.

Yorke JA, Salzberg SL, Langley CH. Decoding the massive genome of loblolly pine using haploid DNA and novel assembly strategies. Genome Biol 2014;15(3):R59.

Zare F, Dow M, Monteleone N, Hosny A, Nabavi S. An Evaluation of Copy Number Variation Detection Tools for Cancer Using Whole Exome Sequencing Data. BMC Bioinformatics 2017 31;18(1):286.

Zarrei M, MacDonald JR, Merico D, Scherer SW. A Copy Number Variation Map of the Human Genome. Nat Rev Genet, 16(3):172-83.

Zimin AV, Stevens KA, Crepeau MW, Puiu D, Wegrzyn JL, Yorke JA, Langley CH, Neale DB, Salzberg SL. An Improved Assembly of the Loblolly Pine Mega-Genome Using Long-Read Single-Molecule Sequencing. GigaScience 2017 Jan 1;6(1):1-4.

APPENDIX A

BIOINFORMATIC SCRIPTS

1. To generate an error model

1.1. Without excluding SNPs

```
python SECNVs/GemSIM/GemErrxy.py -r 125 -f pita_genome_seqs_10longest.fa -s
reads_10longest.sam -n error_model_10longest -p
```

1.2. Excluding SNPs

```
python SECNVs/GemSIM/GemErrxy.py -r 125 -f pita_genome_seqs_10longest.fa -s
reads_10longest.sam -n  error_model_10longest -e SNPs -p
```

2. To generate statistical reports

```
python SECNVs/GemSIM/GemStats.py -m error_model_p.gzip -n error_model -p
```

3. To simulate CNVs and generate reads

```
python SECNVs/SECNVs.py -G pita_genome_seqs_10longest.fa -T
10longest_scaffolds.bed -e_chr 10 -o_chr 1  -o test_loblolly -rn loblolly -f 1000 -ol 50 -q
33 -eN none  -n 31 -s_r 0 -i_r 0 -min_len 50 -max_len 10000 -picard
/general/software/x86_64/easybuild/software/picard/2.18.27-Java-1.8  -GATK
/general/software/x86_64/easybuild/software/GATK/3.8-1-0-Java-1.8.0 \
 -ssr -nr 50000 -sc -pr -fs 300 -sb -l 125 -tf 100 -clr 600 -s 1  -m
error_model_tscaffold2259_p.gzip
```

4. Using ExomeDepth to detect CNVs

```
library(ExomeDepth)

bai.files <- read.table(file = "bai_file_name")
bai.files <- as.character(bai.files[, 1])
bam.files <- read.table(file = "bam_file_name")
bam.files <- as.character(bam.files[, 1])

counts <- getBamCounts(bed.file = "10longest_scaffolds.bed", bam.files = bam.files,
index.files = bai.files, include.chr = FALSE, referenceFasta =
"pita_genome_seqs_10longest.fa")

target <- paste(counts$chromosome, counts$start, counts$end, sep = ":")
colomn_names <- colnames(counts)
counts["target"] <- target
colomn_names <- c("target", colomn_names)
counts <- counts[, colomn_names]

mm <- 30
```

```
for (nn in 1:30){


my.test <- counts[, nn+5]

sample_name <- colnames(counts)[nn+5]

        my.ref.samples <- read.table("bam_file_name", stringsAsFactors= F)[-nn,][1:mm]

        my.reference.set <- as.matrix(counts[, my.ref.samples])

        my.choice <- select.reference.set (test.counts = my.test, reference.counts =

my.reference.set, bin.length = (counts$start - counts$end)/1000, n.bins.reduced = 10000)

        print(my.choice[[1]])


        my.matrix <- as.matrix(counts[, my.choice$reference.choice, drop = FALSE])

        my.reference.selected <- apply(X = my.matrix, MAR = 1, FUN = sum)


        all.exons <- new('ExomeDepth', test = my.test, reference = my.reference.selected,

formula = 'cbind(test, reference) ~ 1')


for (i in seq(0.2,0.4,0.01)) {

        all.exons <- CallCNVs(x = all.exons, transition.probability = i, chromosome =

counts$chromosome, start = counts$start, end = counts$end, name = counts$target)


        head(all.exons@CNV.calls)

        output.file <- 'exome_calls.csv'
        write.csv(file = paste0("output_", sample_name,"_",  mm,
"_samples_transition_probability_", i,".csv"), x = all.exons@CNV.calls, row.names =
FALSE)
}
}
```

APPENDIX B

DETECTED CNVS ON 100 LONGEST SCAFFOLDS OF LOBLOLLY

Table B.1 CNVs detected on 100 longest scaffolds of X001B

type	chromosome	start	end	width
deletion	tscaffold2221	5429230	5429298	69
deletion	tscaffold2221	5829154	5829263	110
deletion	tscaffold2221	7556705	7556860	156
deletion	tscaffold2221	7694368	7696565	2198
deletion	tscaffold2404	1785793	1785875	83
deletion	tscaffold2404	2455156	2455254	99
deletion	tscaffold2404	3697073	3699161	2089
deletion	tscaffold2404	4576152	4576223	72
deletion	tscaffold4938	917075	917707	633
deletion	tscaffold4938	1957319	1957402	84
deletion	tscaffold4938	4823784	4824746	963
deletion	tscaffold4938	4942159	4942481	323
deletion	tscaffold4938	5062768	5064460	1693
deletion	tscaffold616	3229817	3231089	1273
deletion	tscaffold616	5355406	5355481	76
deletion	tscaffold59	649020	649181	162
deletion	tscaffold59	1473353	1473429	77
deletion	tscaffold59	3214794	3214865	72
deletion	tscaffold59	3215254	3215329	76
deletion	tscaffold4850	3540530	3552616	12087
deletion	tscaffold4850	5074070	5074159	90
deletion	tscaffold4352	676735	676959	225
deletion	tscaffold4352	1116387	1116501	115
deletion	tscaffold4352	1118177	1118280	104
deletion	tscaffold4352	1148033	1148369	337
deletion	tscaffold813	2155891	2159997	4107
deletion	tscaffold2120	3166733	3167956	1224
deletion	tscaffold2120	4726870	4726938	69
deletion	tscaffold2120	4732490	4732665	176
deletion	tscaffold2974	267863	268116	254
deletion	tscaffold2974	709482	709555	74
deletion	tscaffold301	2585402	2586198	797
deletion	tscaffold2422	3404949	3406488	1540
deletion	tscaffold5916	4335958	4336120	163

Table B.1 Continued

Type	chromosome	start	end	width
deletion	tscaffold5916	4336623	4340081	3459
deletion	tscaffold4682	3144410	3144569	160
deletion	tscaffold3024	3411400	3567357	155958
deletion	tscaffold6748	1010809	1019499	8691
deletion	tscaffold6748	2495825	2495974	150
deletion	tscaffold6748	3851807	3851880	74
deletion	tscaffold4439	401488	401811	324
deletion	tscaffold4439	2114900	2114975	76
deletion	tscaffold3224	4012651	4012724	74
deletion	tscaffold691	2818785	2819185	401
deletion	tscaffold239	33738	34010	273
deletion	tscaffold239	3084227	3084321	95
deletion	tscaffold5599	551069	581041	29973
deletion	tscaffold5599	960882	961141	260
deletion	tscaffold5599	3267488	3295138	27651
deletion	tscaffold2197	3896046	3896146	101
deletion	tscaffold5122	345861	346301	441
deletion	tscaffold5122	351641	352224	584
deletion	tscaffold5122	1053586	1054139	554
deletion	tscaffold5122	1932865	1932935	71
deletion	tscaffold5122	2225956	2226029	74
deletion	tscaffold5122	2231586	2231658	73
deletion	tscaffold5122	3949019	3950094	1076
deletion	tscaffold788	181381	181555	175
deletion	tscaffold788	182311	242454	60144
deletion	tscaffold788	2764070	2856142	92073
deletion	tscaffold788	3926925	3927210	286
deletion	tscaffold7725	1939643	1939942	300
deletion	tscaffold7725	2574871	3519039	944169
deletion	tscaffold1619	341000	341076	77
deletion	tscaffold1619	1060654	1060802	149
deletion	tscaffold1619	1583104	2136338	553235
deletion	tscaffold1619	2176657	2176731	75
deletion	tscaffold1619	2299176	2299336	161
deletion	tscaffold1619	3622351	3622555	205
deletion	tscaffold7478	169124	169268	145
deletion	tscaffold7478	3484776	3501995	17220
deletion	tscaffold6439	1973479	1973634	156
deletion	tscaffold6439	2260592	2261281	690

Table B.1 Continued

type	chromosome	start	end	width
deletion	tscaffold6439	3363017	3372260	9244
deletion	tscaffold5694	1338460	1338532	73
deletion	tscaffold8864	59045	59269	225
deletion	tscaffold8864	920760	920834	75
deletion	tscaffold2247	663538	664963	1426
deletion	tscaffold2247	964112	964958	847
deletion	tscaffold2296	991459	991532	74
deletion	tscaffold2296	2008760	2008830	71
deletion	tscaffold2296	2922031	2922109	79
deletion	tscaffold1961	2543349	2543773	425
deletion	tscaffold2178	1556446	1556763	318
deletion	tscaffold3776	1480306	1482124	1819
deletion	tscaffold3776	1667804	1670653	2850
deletion	tscaffold3776	1715620	1715694	75
deletion	tscaffold695	3178681	3179510	830
deletion	tscaffold695	3258762	3259408	647
deletion	tscaffold4244	1112875	1113759	885
deletion	tscaffold2590	210615	212075	1461
deletion	tscaffold2590	290971	291226	256
deletion	tscaffold2590	2899091	2901865	2775
deletion	tscaffold4188	1416874	1417060	187
deletion	tscaffold4188	1435074	1438567	3494
deletion	tscaffold4188	1745494	1745595	102
deletion	tscaffold4188	3191757	3191988	232
deletion	tscaffold1547	750335	775546	25212
deletion	tscaffold1547	789293	789443	151
deletion	tscaffold1547	1010745	1033404	22660
deletion	tscaffold1547	1034041	1034119	79
deletion	tscaffold1547	1290892	1290966	75
deletion	tscaffold1547	1301927	1302141	215
deletion	tscaffold1547	1469926	1470076	151
deletion	tscaffold1547	2308031	2308290	260
deletion	tscaffold1547	2810836	2811194	359
deletion	tscaffold1547	2822580	2822878	299
deletion	tscaffold1547	2847693	2847758	66
deletion	tscaffold1547	2869626	2869877	252
deletion	tscaffold355	52165	897128	844964
deletion	tscaffold355	897390	977549	80160
deletion	tscaffold355	1879965	1880259	295

Table B.1 Continued

type	chromosome	start	end	width
deletion	tscaffold355	2436738	2445257	8520
deletion	tscaffold667	851815	852181	367
deletion	tscaffold667	1212129	1212434	306
deletion	tscaffold2554	567208	567279	72
deletion	tscaffold2554	635753	635908	156
deletion	tscaffold2554	1666099	1666169	71
deletion	tscaffold2554	2530192	2530273	82
deletion	tscaffold6451	1577050	1577125	76
deletion	tscaffold6451	2919316	2919408	93
deletion	tscaffold4893	306269	308710	2442
deletion	tscaffold2942	2147449	2147585	137
deletion	tscaffold1117	966359	966603	245
deletion	tscaffold245	946572	946647	76
deletion	tscaffold7299	498906	499009	104
deletion	tscaffold7299	1336550	1336779	230
deletion	tscaffold7299	1337200	1337416	217
deletion	tscaffold7299	2554609	2554713	105
deletion	tscaffold197	931880	931977	98
deletion	tscaffold197	1523216	1523626	411
deletion	tscaffold220	1660872	1660949	78
deletion	scaffold482563	818454	818530	77
deletion	tscaffold3236	441358	442752	1395
deletion	tscaffold3236	1706194	1706268	75
deletion	tscaffold3236	1892671	1892825	155
deletion	tscaffold3236	2528267	2528341	75
deletion	tscaffold3236	2528877	2529183	307
deletion	tscaffold1594	1626736	1626987	252
deletion	tscaffold1594	1883803	1883886	84
deletion	tscaffold934	503342	503643	302
deletion	tscaffold1591	1126728	1126808	81
deletion	tscaffold1591	1165436	1165512	77
deletion	tscaffold1591	1274270	1274376	107
deletion	tscaffold8876	1023388	1051753	28366
deletion	tscaffold164	893468	895274	1807
deletion	tscaffold5336	913729	913846	118
duplication	tscaffold2404	3641339	3642283	945
duplication	tscaffold2404	3763499	3765253	1755
duplication	tscaffold4938	362887	363278	392
duplication	tscaffold4938	1987302	1987411	110

Table B.1 Continued

type	chromosome	start	end	width
duplication	tscaffold616	205320	205488	169
duplication	tscaffold616	519992	520068	77
duplication	tscaffold59	1299917	1299995	79
duplication	tscaffold59	1430354	1430749	396
duplication	tscaffold59	4784470	4784734	265
duplication	tscaffold4850	1666935	1670526	3592
duplication	tscaffold4850	2781684	2782194	511
duplication	tscaffold4352	1178669	1179186	518
duplication	tscaffold4352	4550506	4551302	797
duplication	tscaffold813	2154186	2154291	106
duplication	tscaffold813	4027436	4027512	77
duplication	tscaffold2120	2770859	2772541	1683
duplication	tscaffold2120	3323081	3323156	76
duplication	tscaffold2120	3869546	3869664	119
duplication	tscaffold714	675433	679021	3589
duplication	tscaffold714	1183751	1183912	162
duplication	tscaffold6466	4497515	4497665	151
duplication	tscaffold2974	236688	237126	439
duplication	tscaffold2974	3211487	3211587	101
duplication	tscaffold1706	1497761	1498056	296
duplication	tscaffold5916	407377	691161	283785
duplication	tscaffold5916	810874	824265	13392
duplication	tscaffold5916	1895200	1895268	69
duplication	tscaffold4682	975780	976113	334
duplication	tscaffold3024	1315999	1316541	543
duplication	tscaffold4439	279161	289897	10737
duplication	tscaffold3224	1765668	1765737	70
duplication	tscaffold691	272136	274725	2590
duplication	tscaffold691	1359275	1359474	200
duplication	tscaffold691	1404580	1404678	99
duplication	tscaffold691	2803564	2818670	15107
duplication	tscaffold5599	58223	62696	4474
duplication	tscaffold2197	13336	13866	531
duplication	tscaffold788	2025120	2034915	9796
duplication	tscaffold788	2111368	2112189	822
duplication	tscaffold788	2764070	2780770	16701
duplication	tscaffold788	3000039	3000716	678
duplication	tscaffold788	3730377	3730877	501
duplication	tscaffold7725	311251	313079	1829

Table B.1 Continued

type	chromosome	start	end	width
Duplication	tscaffold7725	552477	552906	430
duplication	tscaffold7725	560011	560086	76
duplication	tscaffold7725	783158	783407	250
duplication	tscaffold7478	1822829	1823330	502
duplication	tscaffold7478	3484776	3517965	33190
duplication	tscaffold7478	3692235	3692346	112
duplication	tscaffold6439	3167538	3182767	15230
duplication	tscaffold4926	2587805	2588256	452
duplication	tscaffold1916	329318	329394	77
duplication	tscaffold1961	2620652	2631458	10807
duplication	tscaffold2178	969848	969980	133
duplication	tscaffold3776	457421	457759	339
duplication	tscaffold2907	71719	71822	104
duplication	tscaffold2907	89611	89823	213
duplication	tscaffold2907	308006	308264	259
duplication	tscaffold2907	627679	627877	199
duplication	tscaffold775	16888	17434	547
duplication	tscaffold695	42505	59635	17131
duplication	tscaffold695	2939808	2947814	8007
duplication	tscaffold1547	1493535	1493855	321
duplication	tscaffold7716	3025228	3025299	72
duplication	tscaffold355	1788490	1789213	724
duplication	tscaffold667	1564064	1585867	21804
duplication	tscaffold2554	810870	811003	134
duplication	tscaffold2554	1102428	1102736	309
duplication	tscaffold2554	2390087	2391558	1472
duplication	tscaffold4893	1012140	1012523	384
duplication	tscaffold1117	421529	421708	180
duplication	tscaffold1117	1421336	1421684	349
duplication	tscaffold1117	1422026	1422895	870
duplication	tscaffold1117	1614556	1615307	752
duplication	tscaffold1117	1615941	1616969	1029
duplication	tscaffold245	918388	920049	1662
duplication	tscaffold245	955625	955695	71
duplication	tscaffold245	1052780	1052979	200
duplication	tscaffold245	2249181	2249564	384
duplication	tscaffold7299	1720511	2331324	610814
duplication	tscaffold7299	2561304	2562066	763
duplication	tscaffold7299	2588945	2589251	307

Table B.1 Continued

type	chromosome	start	end	width
Duplication	tscaffold197	1358538	1358874	337
duplication	scaffold482563	852523	853184	662
duplication	scaffold482563	870972	871210	239
duplication	tscaffold1594	184295	184372	78
duplication	tscaffold2394	2105573	2106024	452
duplication	tscaffold2394	2119144	2119476	333
duplication	tscaffold8876	242116	245819	3704
duplication	tscaffold8876	1579437	1579966	530
duplication	tscaffold164	999608	999683	76
duplication	tscaffold164	1085851	1085960	110