ABSTRACT

SUPPLE, MEGAN A. Probabilistic Allele Calling to Improve Population Size Estimates from Non-Invasive Genetic Mark–Recapture Analysis. (Under the direction of Dr. Kenneth H. Pollock).

Accurate estimates of population sizes are often necessary to help researchers better understand how wildlife populations are changing over time. Researchers often use traditional mark–recapture methods to estimate wildlife population sizes. A variety of models, with varying assumptions, are available to analyze traditional mark–recapture data. The utility of traditional mark–recapture methods is limited when sampling rare or elusive species. Capture probabilities may not be high enough due to the difficulties and cost of capturing the animals. In addition, physical capture can be stressful, even deadly, to the animals.

The limitations of traditional mark–recapture methods can sometimes be addressed by utilizing non-invasive genetic mark–recapture methods. Using the non-invasive genetic method, individuals are not physically captured and tagged. Instead, non-invasive genetic samples, such as hair or scat, are collected and genotyped at multiple microsatellite markers. An individual's genotype serves as a DNA tag, uniquely identifying that individual. DNA is extracted from each sample and the extracted DNA is PCR amplified multiple times at several microsatellite loci. The results of each PCR amplification are visualized using capillary electrophoresis, resulting in an electropherogram. Alleles are called by interpreting the peak heights and/or peak areas on the electropherogram.

While non-invasive genetic methods solve some of the problems of traditional mark–recapture, they also introduce some new problems. One major problem introduced by non-invasive genetic methods is the misidentification of individuals. The DNA from non-invasive samples is often low in quality and/or low in quantity, which increases the probability of genotyping errors. In addition, poor marker selection can result in individuals sharing a genotype. Traditional mark–recapture methods are not robust to violations of the assumption that individuals are correctly identified. Genotyping errors cause overestimation of population size; markers that lack the power to distinguish between individuals cause underestimation of population size.

To achieve better population size estimates, I propose a new probabilistic allele calling method. In the traditional method, definitive allele calls are made independently for each PCR replicate of a sample. Then, the definitive allele calls are examined to determine the sample's genotype. The new method assigns probabilities to allele calls, rather than determining a definitive allele call. Probabilities are assigned to possible allele calls based on electropherogram peak heights. For cases of possible allelic drop out, a portion of the probability distribution for the PCR replicate is assigned to a heterozygous allele call with one undesignated allele. For each sample, the allele call probabilities at each locus, including allele calls with undesignated alleles, are averaged from the PCR replicates. Then, possible allele calls with undesignated alleles are assigned based on the allele frequencies in the averaged probabilities. The genotype with the highest probability is assigned as the sample's genotype. Using the probabilistic method, uncertainty remains in the allele calls until all the PCR replicates of a sample are examined. This allows more information from the electropherograms to be utilized when determining genotypes.

To examine the proposed probabilistic allele calling method, I compared it to a traditional method by running computer simulations that examine a variety of scenarios. For each simulation scenario, a population was generated and sampled using non-invasive genetic mark–recapture methods. Each sample, which contained DNA of low quality and quantity, was genotyped at multiple microsatellite loci, with multiple PCR replicates for each locus. Genotypes were determined for samples using a traditional allele calling method and the new probabilistic allele calling method. The resulting genotypes were matched and the data was analyzed using four traditional closed mark–recapture models.

The probabilistic method performed better than the traditional method in almost all cases. When more than two PCR replicates were examined, the estimates from the probabilistic method were less biased and more precise than estimates from the traditional method. Using the probabilistic method, good estimates can be achieved using fewer PCR replicates. This new method of analyzing non-invasive genetic mark–recapture data has the potential to allow wildlife population sizes to be accurately estimated using non-invasive methods in less time and at lower cost than current methods.

Probabilistic Allele Calling to Improve Population Size Estimates from Non-Invasive Genetic Mark–Recapture Analysis

by
Megan A. Supple

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Biomathematics

Raleigh, North Carolina

2009

APPROVED BY:

Dr. Kevin Gross

Dr. W. Owen McMillan

Dr. Kenneth H. Pollock
Chair of Advisory Committee

DEDICATION

To Hedgy.

BIOGRAPHY

From the time I learned to ride horses when I was a kid, I spent as much time as possible at the barn. I would ride any horse I could get my hands on—including the unbroke or wild ones. My sister and I rode a pony named Sparkle (also known as The Little Witch) who was being given away because she had the intelligence and athleticism to dump any rider. Our parents, not being knowledgeable about horses, thought keeping Sparkle was a good idea. Sparkle excelled at jumping, bucking, and keeping her rider humble...the kind of pony every kid should learn to ride on.

At 18 years old, it was on to the University of Michigan. Being born and raised in sunny southern California, I thought it would be fun to go somewhere with snow. I imagined sledding, snowball fights, and building snowmen. Instead it was just bitter cold the entire school year. Every summer I took the opportunity to seek adventure elsewhere, including the Sierra Nevada Mountains and Yellowstone National Park. After four years, I graduated from Michigan with a degree in Aerospace Engineering. Unlike most rocket scientists, I did not to put my degree to work. Instead I thru-hiked the Appalachian Trail, spending six months walking from Georgia to Maine.

After a stint living in a tent in Alaska, I moved to Washington State to pursue a career as a stable hand. While working as a stable hand, I also worked as an EMT on an ambulance, retrained problem horses, and worked at a fecal DNA lab. While in Washington, I took up foxhunting. I am proud to say that, while foxhunting, I have fallen off horses on three different continents.

Eventually I landed in Raleigh, North Carolina to pursue higher education and more horses. I spend my free time volunteering for a local equine rescue organization. I am the proud foster mom to a string of naughty ponies who need to learn to behave themselves before anyone will adopt them. I still backpack every chance I get, even if it is -20^0F or over 100^0F. Breaks from school find me anywhere from the swamps to the mountains.

I am one of the founders and a regular contributor to Hillbilly Farms, a world famous website devoted to our battle against perfection in the equestrian world. In addition to contributing fodder for the website, I also write the “Jumping Clinic” column under the pseudonym George Morris. I have recently opened Ponies on Probation, a branch of Hillbilly Farms.

ACKNOWLEDGMENTS

I want to thank my committee members for their invaluable help on this project. I am grateful to my committee chair, Ken Pollock, for his help on the entire project, especially the mark–recapture aspects. I want to thank Owen McMillan for answering my countless genetics questions and Kevin Gross for his assistance with all of my coding issues.

I want to thank my family, who always supports me, even when they don't understand what I do. Thanks to my friends for ensuring that this project did not get done in a timely manner. Thanks to the horses for helping me maintain my sanity.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF NOMENCLATURE

allele A microsatellite allele is defined by the number of tandem repeats of the short DNA sequence.

allele call An estimation of which alleles are present in a sample based on examination of the electropherogram. See *false alleles* and *allelic drop out* for examples of errors in allele calling.

allelic drop out The failure of one allele in a heterozygote to amplify.

capillary electrophoresis The method used to separate DNA fragments of different sizes to allow visualization of alleles.

capture frequencies The capture frequencies are represented by a vector where the i^{th} coordinate is the number of individual animals that were captured exactly i times.

capture history An individual animal's capture history is the record, for each sampling occasion, of whether or not the individual was captured. "0" indicates no capture and "1" indicates a capture.

capture history matrix The collection of individual capture histories for all animals captured during at least one sampling occasion. It indicates for each animal, at each sampling occasion, whether the animal was capture during that occasion. "0" indicates no capture and "1" indicates a capture.

capture probability The probability that an individual animal will be captured during a sampling occasion.

consensus genotype A sample's consensus genotype is the collection of the two alleles determined to be present for each locus. It is determined by examining, for each locus, the allele calls of each PCR replicate. The number of replicates that contain a given allele is compared to the consensus threshold value. If the count is larger than the threshold, the allele is considered part of the genotype at that locus.

electropherogram A graph of the intensity of the electrophoresis signal versus the size of the amplified DNA fragments. One electropherogram is generated for each PCR replicate, at each locus, for each sample. Alleles are called by examining the peak heights and/or peak areas of the electropherogram.

false allele An allele that is called as part of the sample's genotype, but that is not part of the animal's true genotype. False alleles are caused by contamination and stutter.

genotype A locus genotype is the two alleles present at a given locus. An animal's genotype is the collection of locus genotypes for that animal. A sample's genotype is the estimation of the animal's genotype based on a DNA containing sample from the animal.

ghost A non-existent individual that appears in the capture history matrix due to the misidentification of a sample.

locus (loci pl.) A microsatellite locus is a region of DNA containing a microsatellite marker.

mark–recapture A method used to estimate wildlife population sizes by capturing, marking, and releasing individual animals at multiple sampling occasions. Captured individuals are marked with unique tags that allow researchers to determine each individual's capture history. A population size estimate is obtained by analyzing the capture history matrix.

microsatellite markers A region of DNA containing tandem repeats of short DNA sequences. The number of repeats varies between individuals in a population.

non-invasive genetic mark–recapture A mark–recapture method that uses genotypes from non-invasive DNA sources as the tags to uniquely identify individuals.

PCR (polymerase chain reaction) A method to amplify the number of DNA molecules from a selected region of DNA.

PCR replication PCR performed multiple times at a given locus, for a given sample.

sampling occasion The period of time during which animals are captured or samples containing DNA are collected.

shadow An individual whose genotype is identical to the genotype of another individual in the population.

study period The time frame encompassing the multiple sampling occasions, during which animals are captured or DNA samples are collected for the study.

stutter A type of genotyping error that results in a false allele due to a PCR amplification error.

LIST OF NOTATION

$\boldsymbol{a}$	designated allele
$\boldsymbol{b}$	designated allele
f_i	number of animals captured exactly i times
$\boldsymbol{i}$	designated allele
$\boldsymbol{j}$	designated allele
$\boldsymbol{k}$	number of sampling occasions
$\boldsymbol{L}$	total number of loci examined
$\boldsymbol{m}$	total number of PCR replications
MLE	maximum likelihood estimate
$\boldsymbol{N}$	true population size
$\boldsymbol{n}$	is the total number of captures, not including repeat captures of a single individual at a sampling occasion
$\boldsymbol{p}$	probability of capture
relative MSE	relative mean squared error
RMSE	root mean squared error
$\boldsymbol{s}$	number of possible captures per individual animal, per sampling occasion
$\boldsymbol{Z}$	undesignated allele

Chapter 1

Introduction

Accurate estimates of population sizes and other demographic parameters are necessary to help researchers better understand how wildlife populations are changing over time. Understanding how various factors affect demographic parameters allows researchers to design effective conservation plans for species at risk of extinction. The continued monitoring of demographic parameters after a conservation plan has been implemented allows researchers to determine the effectiveness of the plan.

There are a variety of methods that allow researchers to effectively estimate wildlife population sizes when time and money are limited. Most of these methods are based on counts of a portion of the population, rather than a complete census, but each uses a different approach to estimate the unseen portion of the population. These count methods include simple count indicies, distance methods, and mark–recapture methods (Williams et al. 2002). I will focus on mark–recapture methods because they are commonly used and allow researchers to obtain estimates of population sizes using a variety of tagging methods, including non-invasive genetic tags.

1.1 Traditional Mark–Recapture Methods

Researchers often use traditional mark–recapture methods to estimate wildlife population sizes. Individuals from the population are physically captured, marked with a unique tag, and released back into the population. The population is re-sampled multiple times and researchers identify individuals with marks from a previous capture. A capture history is generated for each captured individual indicating, for each sampling occasion, whether or not the individual was captured. The capture histories of all of the captured individuals in the study are entered into a capture history matrix (see Table 1.1 for example matrix). A variety of models with varying assumptions are available to analyze capture history matrices (Williams et al. 2002). When model assumptions are met, mark–recapture methods can provide researchers with accurate estimators of population size.

Table 1.1: Example capture history matrix for 8 individuals and 10 sampling occasions.

	sampling occasion									
individual	1	2	3	4	5	6	7	8	9	10
1	0	1	1	0	0	0	1	0	1	0
2	1	1	0	1	1	0	0	0	1	1
3	0	0	0	0	1	0	0	0	0	1
4	0	1	1	1	0	0	1	1	1	0
5	1	1	0	1	0	1	1	0	0	0
6	1	0	0	0	0	1	1	0	0	1
7	0	0	0	1	0	0	0	0	0	0
8	1	1	1	1	0	1	1	1	0	1
	0 indicates individual not captured. 1 indicates individual captured.									

There are three categories of mark–recapture models—closed population models, open population models, and combined open and closed models (see Figure 1.1). The closed population models assume that there are no births, no deaths, and no migration during the study period. The open models relax these assumptions, allowing births, deaths, or permanent migration. Neither open nor closed population models allow for temporary emigration (Pollock et al. 1990). Some combined models can yield unbiased population size estimates in the presence of temporary emigration (Kendall et al. 1997).

The simplest mark–recapture model is the Lincoln–Petersen model, which has two sampling occasions. This model has very restrictive assumptions: a closed population, equal capture probabilities for all individuals at each occasions, and marks are not lost (Williams et al. 2002). While the assumptions of the model are restrictive, it serves as a good model to understand the general mark–recapture concept. During the first sampling occasion, individuals are captured, marked, and released back into the population. During the second sampling occasion, individuals are captured and researchers take note of the

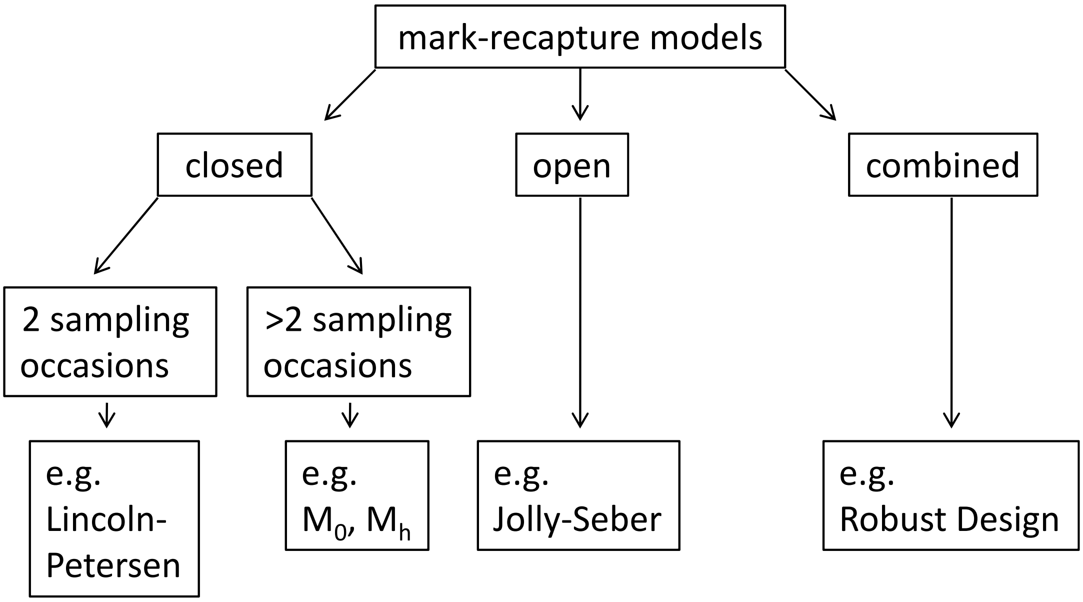


Figure 1.1: Types of mark–recapture models and examples of each type.

number of individuals that are marked, indicating that they were captured during the first sampling occasion. To estimate the population size, first researchers calculate the proportion of individuals captured in the second sampling occasion that were marked in the first sampling occasion. Then researchers determine the equation of the proportion of individuals in the entire population that are marked, which is the total number of marked individuals divided by the unknown population size. Under the assumptions of the model, these two proportions are equal. Equating them and solving for the population size will yield an estimate of population size (Williams et al. 2002).

More robust estimates can be obtained by using models that relax the assumptions and increase the number of sampling occasions. For more than two sampling occasions there is a standard suite of closed mark–recapture models. These models assume a closed population and that individuals are correctly identified using unique tags. The models vary in their assumptions regarding the capture probabilities, which can vary between sampling occasions, between individuals, and in response to prior trapping (Williams et al. 2002).

The baseline model in the suite of closed mark–recapture models is model M_0. This model makes the assumption that the capture probabilities are the same for each sampling occasion, for each individual, and regardless of whether the animal has been previously captured. Population size estimates can be obtained from model M_0 using maximum likelihood estimation. Estimates from model M_0 can be significantly biased if capture probabilities are not equal (Otis et al. 1978; Williams et al. 2002). The direction of the bias depends on the source of variation in the capture probabilities, with individual variation causing a negative bias (Pollock et al. 1990).

There are many sources of variation in capture probabilities. Capture probabilities can vary over time for various reasons, including varying weather conditions (Williams et al. 2002). Capture probabilities can vary between individuals in relation to many factors, including gender and age (Williams et al. 2002; Boulanger et al. 2004). An animal that has been previously captured may exhibit behavioral variation in capture probabilities due to a "trap–shy" or a "trap–happy" response when they encounter traps in subsequent sampling occasions (Pollock et al. 1990; Williams et al. 2002). The remainder of the models in the suite of closed mark–recapture models allow for different sources of variation in capture probabilities in all possible combinations (Table 1.2).

Table 1.2: Sources of variation in capture probabilities allowed in the closed mark–recapture models.

	source of variation in capture probabilities		
model	time (t)	individual (h)	trap response (b)
M_0			
M_t	$\surd$		
M_h		$\surd$	
M_b			$\surd$
M_{th}	$\surd$	$\surd$	
M_{tb}	$\surd$		$\surd$
M_{hb}		$\surd$	$\surd$
M_{thb}	$\surd$	$\surd$	$\surd$

$\surd$ indicates a source of variation allowed in the model.

The importance of heterogeneity in capture probabilities necessitates a more detailed discussion of model M_h. Model M_h allows for capture probabilities to vary between individuals, but not over time or due to prior trapping. Since each animal has its own capture probability, model M_h has a large number of parameters to estimate. For this reason a variety of methods are used to obtain estimates under model M_h (Otis et al. 1978; Williams et al. 2002). Three commonly used approaches are the finite mixture estimator (Pledger 2000), the jackknife estimator (Burnham and Overton 1978), and the Chao estimator (Chao 2006).

The finite mixture approach models heterogeneity by separating individuals into a small number of groups. Individuals within each group have equal capture probabilities. In many cases, separating the individuals into two groups is sufficient to model the heterogeneity (Pledger 2000). The jackknife estimator is an ad hoc method that assumes that the capture probabilities are a random sample from an unspecified distribution (Burnham and Overton 1978). The jackknife estimator, although it does not have a strong theoretical basis, is fairly robust (Otis et al. 1978), but population size estimates can be negatively biased if capture probabilities are very low (Chao 1989). The Chao estimator was developed to handle sparse data, due to low capture probabilities, where other heterogeneity models do not perform well (Chao 1989). It is based on the idea that rarely captured individuals contribute the most information about uncaptured individuals. Estimates from the Chao estimator can be negatively biased (Chao 1989) because it was intended as a lower bound on population size, not a point estimate (Chao 2006).

While using models that account for heterogeneous capture probabilities is often necessary, these models introduce some additional problems. These models reduce bias, but the cost is a decrease in precision (Otis et al. 1978; Boulanger et al. 2004). In addition, the estimates of population size depend on the specific model used due to an identifiability problem. Simulating two populations with very different sizes, each under a different model, can result in the same capture history for both populations (Link 2003). Regardless of the identifiability problem, heterogeneity models are often used to analyze mark–recapture data because there is extensive evidence of heterogeneous capture probabilities.

Researchers must also consider the assumption of population closure. When the closure assumption cannot be met, it is necessary to use an open population model. The Jolly–Seber model—which allows births, deaths, immigration, and permanent emigration—provides an estimate of population size at each sampling occasion. The model assumptions include equal capture probabilities of all individuals present during a sampling occasion and correct identification of individuals (Pollock et al. 1990). When there is substantial variation in capture probabilities between individuals and capture probabilities are not high, the estimates of population size can be negatively biased (Williams et al. 2002). The Jolly–Seber model can be extended to consider capture probabilities that are stratified by age class (Pollock et al. 1990), which may reduce some of the individual variation in capture probabilities.

The robust design combines open and closed models, with each type of model analyzing the data at a different scale. The robust design utilizes multiple primary sampling periods, each containing multiple secondary sampling periods. The secondary sampling periods within a single primary period are close enough in time that closure can be assumed, allowing analysis with closed population models that allow variation in capture probabilities. The primary periods are analyzed by combining the data from the secondary periods for each primary period. The data at this scale can be analyzed using an open population model (Pollock et al. 1990). When using the robust design, the data must meet the assumptions of the models used at the different scales (Williams et al. 2002); however, the model can be extended to allow for temporary emigration (Kendall et al. 1997).

There are a variety of models with varying assumptions that allow researchers to obtain accurate population size estimates. However, traditional mark–recapture has limitation, especially when sampling rare, elusive, or endangered species (Mills et al. 2000). If individuals are elusive, or if the density of the population is low, capture probabilities may not be high enough to obtain accurate population size estimates. In addition, physical capture can be stressful to an animal, decreasing the animal's ability to survive in the wild. Physical capture can also directly cause the injury or death of the animal (Langer 2006). These limitation have lead researchers to examine modifications of the traditional mark–recapture methods that have higher capture probabilities and do not rely on physically capturing the animals. I will focus on the use of non-invasive genetic samples to estimate wildlife population sizes under the closed mark–recapture models.

1.2 Non-Invasive Genetic Sampling

The limitations of low capture probabilities and stress on the animals that occur when using traditional mark–recapture methods can be addressed by utilizing natural tags instead of applied tags. In traditional mark–recapture, tags are physically applied to the animal to allow identification at a future capture. Instead of applying tags, animals can be identified without physical capture by using natural tags. Animals can be uniquely identified using photographs showing scars or color patterns (photo tags), or by genotypes from non-invasive DNA samples (DNA tags) (Yoshizaki 2007). I will focus on DNA tags because they are currently being used to study a wide variety of species.

To uniquely identify individual animals using DNA tags, non-invasive samples containing DNA, such as hair or scat, are collected and genotyped at multiple microsatellite markers. An individual's genotype serves as its DNA tag, uniquely identify that individual. When the same genotype is encountered in two different samples, the second sample is considered a "recapture". Non-invasive samples may be abundant, resulting in higher capture probabilities than when traditional physical capture is used (Mills et al. 2000). The higher capture probabilities result in larger sample sizes. In addition, the animals are not physically captured, so there is no stress on the animals and no risk of injury or death of the animals due to capture. Mark–recapture methods that utilize DNA tags often allow researchers to obtain accurate estimates of population sizes more quickly and more efficiently than when traditional tags are physically applied to the animals (Waits 2004).

1.2.1 Genotyping Non-Invasive Samples

Mark–recapture methods that utilize DNA tags require samples to be identified by genotype. The process of genotyping begins with the extraction of DNA from the samples. DNA can be extracted from a variety of non-invasive sources including hair, scat, urine, skin, feathers, and egg shells (see references in Waits (2004)). The two most common sources of non-invasive DNA used in wildlife studies are hair and scat (Waits 2004). Hair sampling has been used on a wide variety of species including bears, chimpanzees, and gibbons (see references in Waits (2004)). Hair is often collected by attracting animals to the sampling area using food or scent lures. In the sampling area, hair is collected on barbed wire hair snags. Shed hair can also be collected. Shed hair is often collected at nesting sites, provided the nest is occupied by a single individual (Gagneux et al. 1997). Animal hair contains follicular cells on the roots from which DNA can be extracted (Waits 2004). Samples with multiple hairs can be difficult to analyze since a single sample may contain hair from more than one individual. Scat sampling has been used on a wide variety of species including bears, elephants, wolves, bonobos, gibbons, chimpanzees, langurs, and rhinoceros (see references in Waits (2004)). Scat is a plentiful resource and it contains numerous epithelial cells from the intestinal mucosa from which DNA can be extracted (Wasser 1997; Waits 2004). It is easier to visually determine if a scat samples is from the target species than it is for a hair sample (Ruell and Crooks 2007). In addition, DNA extracted from scat has a higher concentration of DNA than DNA extracted from hair (Morin et al. 2001).

After DNA is extracted from the non-invasive sample, multiple microsatellite regions of the DNA are examined. Microsatellites, also known as simple sequence repeats (SSRs) or short tandem repeats (STRs), are regions of nuclear DNA that are non-coding and highly variable between individuals in a population. A microsatellite contains multiple tandem repeats of a small sequence of DNA, usually 1–5 base pairs in length (Tamaki 2007). The different alleles found in a population are due to differing number of tandem repeats, which result in differently sized DNA fragments.

For each DNA sample, multiple microsatellite regions are amplified using polymerase chain reactions (PCR). For each sample, each microsatellite region is usually PCR amplified multiple times for verification of the results. PCR amplification is a method used to increase the amount of specific DNA sequences using temperature cycles that denature the DNA, anneal the primers, and copy the DNA (Pierce 2003). The amplified DNA is labeled with fluorescent dye and visualized by capillary electrophoresis, resulting in an electropherogram for each PCR amplification. An electropherogram is a graph of the intensity of the fluorescent signals versus the size of the amplified DNA fragments (see Figure 1.2).

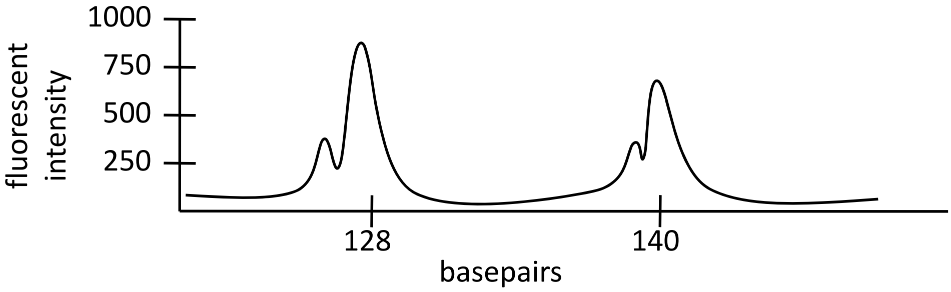


Figure 1.2: Example electropherogram from a single PCR replicate of a single locus for a single sample. The electropherogram shows a heterozygote with alleles 128 and 140. The two small peaks to the left of the two larger peaks are the stutter peaks. See section 1.2.2 for an explanation of stutter peaks.

Alleles are then called for each PCR amplification. Allele calling is the determination of which alleles are present in the sample by interpreting the peak heights and/or peak areas of the electropherogram (Butler 2005). The intensity of the fluorescent signal at a given fragment size, as measured by the peak height and/or peak area on the electropherogram, indicates how many DNA fragments of that size are present in the sample. The size of the amplified DNA fragment indicates how many tandem repeats are present, which indicates what allele is present. To determine a sample's genotype, alleles are called for each PCR replicate, of each locus, for that sample. Then, all of the PCR replicates at each locus for that sample are examined to determine the sample's genotype at each locus. A sample's locus genotype is the two alleles that are determined to be present in the sample at that locus. Finally, the overall genotype is determined by combining all of the locus genotypes for a sample (see Table 1.3 for example genotypes).

Table 1.3: Examples genotypes for three samples at eight loci.

sample 1		sample 2		sample 3	
locus	genotype	locus	genotype	locus	genotype
G1A	(188,190)	G1A	(180,180)	G1A	(192,200)
G10B	(150,162)	G10B	(142,166)	G10B	(148,166)
G10C	(101,101)	G10C	(105,117)	G10C	(111,113)
G10L	(165,171)	G10L	(139,143)	G10L	(161,161)
G10M	(202,220)	G10M	(196,200)	G10M	(210,212)
G10P	(155,159)	G10P	(143,143)	G10P	(141,167)
G10X	(145,161)	G10X	(125,169)	G10X	(131,131)
G1D	(180,188)	G1D	(176,182)	G1D	(174,190)

1.2.2 Misidentification

Non-invasive genetic mark–recapture methods solve some of the problems of traditional mark–recapture, but they also introduce some new problems. One major problem introduced by non-invasive genetic methods is the misidentification of samples. There are two forms of misidentification that occur when using DNA to identify samples, resulting in either a "shadow" individual or a "ghost" individual (see Figure 1.3). A shadow individual is generated when a sample from an individual is misidentified as belonging to a different individual (Mills et al. 2000). This occurs when the genetic markers do not have enough power to distinguish between individuals (Paetkau 2003; Petit and Valiere 2006). A ghost individual is generated when a sample from an individual is misidentified as belonging to a non-existent individual (Paetkau 2003). This occurs due to genotyping errors. A genotyping error almost always results in new genotype, rather than matching an already existing genotype. While misidentifications due to shadows and ghosts both have important impacts on population size estimates, I will focus on ghosts because shadows can be reduced by proper marker selection.

The DNA from non-invasive samples is often low in quantity, which results in low amounts of template DNA in the extracts. In addition, non-invasive samples are often low in quality, which decreases the amount of template DNA available for the genotyping process (Taberlet and Luikart 1999; Ladd et al. 2001). The lower the amount of template DNA, the higher the probability of genotyping errors (Taberlet and Luikart 1999; Gill 2001; Whitaker et al. 2001; Miller et al. 2002; Paetkau 2003; Buchan et al. 2005). There are two types of genotyping errors of concern—allelic drop out and false alleles (see Figure 1.4). False alleles are caused by two different mechanisms—stuttering and contamination (Taberlet et al. 1996; Miller et al. 2002).

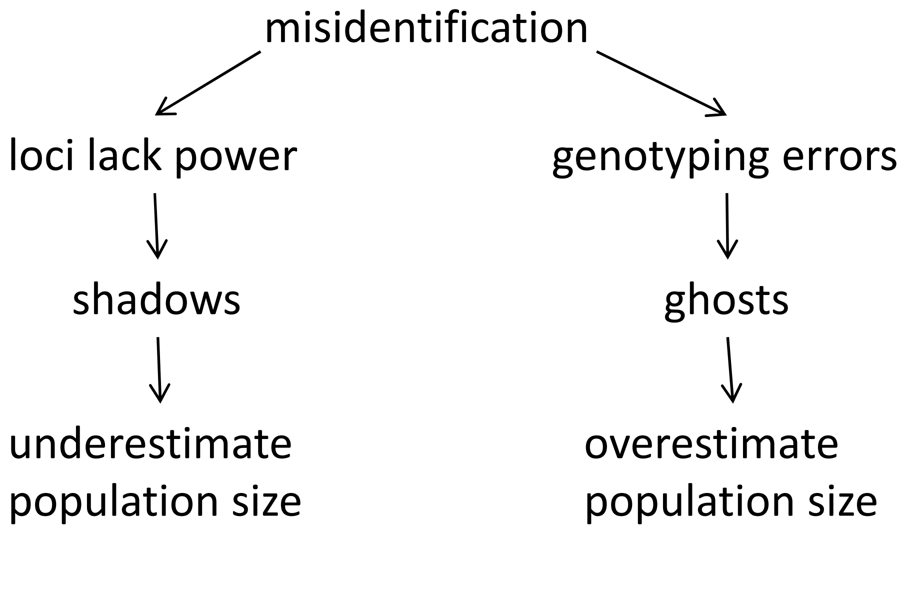


Figure 1.3: Types of misidentification that occurs in non-invasive genetic mark recapture studies and the resulting biases.

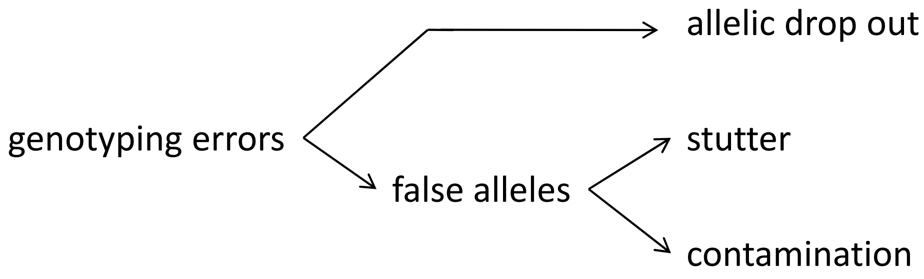


Figure 1.4: Types of genotyping errors

Genotyping low quantity and/or low quality DNA may result in allelic drop out, which is when an allele fails to amplify due to a stochastic sampling error (Taberlet and Luikart 1999; Taberlet et al. 1999; Miller et al. 2002; Butler 2005). The lower the quantity of template DNA, the higher the probability of drop out (Taberlet et al. 1996; Ladd et al. 2001). Estimates of the rate of allelic drop out vary between studies. Rates have been recorded as high as 48% (baboon scat, Smith et al. (2000)) and 31% (chimpanzee shed hair, Gagneux et al. (1997)) per PCR replicate of heterozygous genotypes. Simulation studies show that when at least one allele successfully amplifies from low quantity DNA, the probability of drop out can reach 50% per PCR replicate (Taberlet et al. 1996). Different loci have different PCR amplification success rates due to a number of factors including the size of the alleles present and the efficiency of the primers. As amplification success rate decreases, the rate of allelic drop out increases (Buchan et al. 2005). There is also a tendency for preferential amplification of smaller alleles (Gill et al. 1997; Clayton et al. 1998; Whitaker et al. 2001; Buchan et al. 2005), resulting in higher rates of allelic drop out for larger alleles (Tully 1993; Whitaker et al. 2001). This may be a result degradation having a larger effect on alleles with higher molecular weights (Taberlet et al. 1999; Ladd et al. 2001; Gill et al. 2006).

Genotyping low quantity and/or low quality DNA may result in false alleles. Estimates of the rate of false alleles vary between studies. False allele rates as high as 8% (badger scat, Frantz et al. (2003)) and 5.6% (chimpanzee shed hair, Gagneux et al. (1997)) per PCR replicate have been recorded. False alleles are caused by stuttering or contamination. A stutter peak is a false allele that appears to amplify due to a copy error during PCR amplification. The copy error is believed to be caused by a slipped–strand mispairing of the

Taq polymerase enzyme and the template strand during PCR amplification (Taberlet et al. 1996; Gill et al. 1997; Clayton et al. 1998; Gill et al. 1998; Butler 2005). The copy error results in molecules in the PCR product that are a different size than the template molecule, usually by multiples of the repeat length (Shinde et al. 2003). Stutter products that are smaller than the original molecule (contractions) are much more likely to occur than stutter products that are larger than the original molecule (expansions). For dinucleotide repeat loci, stutter contractions are 14 times more likely than stutter expansions (Shinde et al. 2003).

In addition to the smaller size of the molecules, stutter peaks are often identified by their peak heights relative to the parent allele. When there is a sufficient quantity of DNA at a tetranucleotide repeat locus, peak heights of stutters are usually less than 15% of the parent allele (Gill et al. 2000; Butler 2005). When a dinucleotide locus is examined, or there is a low quantity of DNA, the peak heights of stutters can be sufficiently large that interpretation of the electropherogram can be difficult (Taberlet and Luikart 1999; Gill et al. 2000; Butler 2005; Gill et al. 2006). Rates of stutter likely vary between loci (Clayton et al. 1998; Gill et al. 1998; Buchan et al. 2005). Different loci have alleles with differing number of repeats and stutter rates increase with an increase in the number of repeats (Lai and Sun 2004; Butler 2005).

Using low quantities of DNA may result in extra alleles on the electropherogram due to contamination. When attempting to amplify low quantities of DNA, additional PCR cycles are needed to ensure there is a sufficient signal to call alleles (Navidi et al. 1992; Gill et al. 2000; Butler 2005). Increasing the number of PCR cycles allows an allele from a single template molecule to be visualized (Taberlet and Luikart 1999; Gill et al. 2005). The additional PCR cycles increase the risk of amplifying DNA from contamination (Taberlet and Luikart 1999; Gill et al. 2000; Gill 2001). Sources of contamination include the environment where the sample was collected, other samples, and previously amplified DNA (Butler 2005). Contamination from other species or other populations can be inferred by the presence of alleles not known to occur in the study population. Contamination from within the study population is more difficult to detect and has been estimated at 0.8% (elephant scat) and 0.5% (baboon scat) (Buchan et al. 2005).

Another issue that makes analysis of electropherograms difficult is the presence of null alleles. Null alleles are alleles that fail to amplify due to a mutation at the primer binding site (Butler 2005). An individual that is heterozygous with one null allele, will always appear as homozygous for the other allele, unless a different primer is used. Null alleles consistently fail to amplify, so an individual with a null allele still can be identified by its unique genotype. However, the presence of null alleles in a population will make analysis of electropherograms difficult because the intensity of the fluorescent signal will be lower than expected when a null allele is present. Null alleles also reduce the power of a locus to distinguish between individuals (Eggert et al. 2003).

1.2.3 Reducing Misidentification of Samples

Different methods can be used to reduce the rate of misidentification. A pilot study is necessary to test if non-invasive genotyping is feasible for the population of interest (Taberlet and Luikart 1999). The quantity and quality of DNA varies between studies due to a number of factors including variation between species (Taberlet and Luikart 1999) and variation between samples types. A pilot study allows researchers to determine if there will be enough template DNA to accurately genotype samples (Taberlet and Luikart 1999). Careful marker selection, rigorous field and lab protocols, careful analysis of allele calls and genotypes, and replication of PCR amplifications are important steps to reduce the rate of misidentification.

Misidentification leading to a “shadow”, which is when two unique individuals share a genotype, can be reduced by careful marker selection (Paetkau 2003; Waits and Paetkau 2005). A pilot study can help researchers select enough highly polymorphic markers to ensure enough power to distinguish between closely related individuals (Taberlet and Luikart 1999). When determining how many markers to use, researchers need to bear in mind that the occurrence of genotyping errors increases as more markers are examined (Waits and Paetkau 2005).

Genotyping errors can be reduced by implementing rigorous field and lab protocols. The freshest possible samples should be collected from the field because DNA from fresh samples will be less degraded than DNA from older samples (Taberlet et al. 1999). Careful collection of samples from the field can help reduce the rate of contamination. When using additional PCR cycles to analyze low quantity and/or low quality DNA, separate labs should be designated for pre-PCR procedures and post-PCR procedures to reduce the risk of contamination. Positive controls should be PCR amplified and analyzed to ensure that appropriate procedures are being used to amplify samples and call alleles. Extract blanks and PCR blanks should be PCR amplified and analyzed to see if any erroneous DNA amplified, indicating possible contamination from reagents or other sources. (Navidi et al. 1992; Taberlet and Luikart 1999; Taberlet et al. 1999; Gill 2001; Paetkau 2003; Wasser et al. 2004; Butler 2005).

Early detection and culling of poor samples is important to efficiently obtaining accurate genotypes (Paetkau 2003). Species identification of samples, using mitochondrial DNA (mtDNA), should be done before genotyping to help detect poor samples. The mtDNA has more copies per cell than nuclear DNA, so mtDNA amplifies more readily than nuclear DNA (Waits and Paetkau 2005). Any sample not showing good amplification with mtDNA markers should be culled (Paetkau 2003; Wasser et al. 2004). Once amplification with microsatellite markers begins, any sample that fails to amplify at a large number of loci should be culled (Paetkau 2003).

Careful analysis of allele calls and genotypes can also help detect genotyping errors. Samples with three or more alleles at any locus should be scrutinized and possibly culled, since the sample may have DNA from more than one individual (Paetkau 2003). The presence of an allele outside the expected range for a locus may indicate a contaminated sample. A sample that has a very similar genotype to another sample should be reamplified to verify that the true genotypes are actually different (Paetkau 2003). Any locus showing deviations from Hardy–Weinberg proportions should be examined for the presence of null alleles (Butler 2005) or genotyping errors.

Lower genotyping error rates are obtained by performing PCR replications at each locus, for each sample. For each sample, a consensus genotype is determined from these multiple amplifications by calling alleles at each locus of each PCR replicate and comparing the allele calls to a preselected consensus threshold value. The consensus threshold value is the number of replications in which an allele needs to be called before the allele is considered part of the consensus genotype (Valiere et al. 2002).

The multiple–tubes approach uses extensive PCR replications to reduce genotyping error rates (Navidi et al. 1992; Taberlet et al. 1996). This method requires a minimum of 2 PCR replications for each heterozygote and a minimum of 7 PCR replications for each homozygote. The multiple–tubes approach does not completely eliminate genotyping errors, but it usually reduces the sample misidentification rate to below 1% (Waits 2004; Petit and

Valiere 2006). The multiple–tubes approach is widely accepted, but it is rarely adhered to because it is expensive, time consuming, and requires more DNA since the DNA must be spread across more PCR replicates (Morin et al. 2001; Miller et al. 2002; Valiere et al. 2002; Paetkau 2003). Reducing misidentification rates is an important step towards improving population size estimates, but complete elimination of misidentification is rarely achieved in non-invasive genetic mark–recapture studies.

1.3 Model Biases Resulting from Non-Invasive Genetic Sampling

When analyzing any mark–recapture data, it is important to understand whether the data collected meets the assumptions of the model, otherwise estimates may be biased. In addition to the violations of model assumptions that often occur in traditional mark–recapture, the use of non-invasive genetic mark–recapture introduces some additional sources of violations, which can result in significant biases in population size estimates.

Using models that allow for heterogeneous capture probabilities is often necessary in traditional mark–recapture. In non-invasive genetic mark–recapture it is often even more important to use heterogeneity models because an animal's capture probability has two components: the probability of collecting a sample from the animal and the probability of successfully amplifying DNA from the sample (Waits 2004; Lukacs and Burnham 2005). However, heterogeneity may be reduced because the data used to estimate population sizes relies on whether or not an animal was captured during a sampling occasion, not how many times it was captured during that sampling occasion. Since each animal can be captured multiple times during a given sampling occasion, the record for an animal captured multiple times in that occasion would be the same as the record for an animal that was captured only one time in that occasion (Boulanger et al. 2004).

The traditional mark–recapture models assume that individuals are correctly identified. Violations of this assumption result in biases in population size estimates (see Figure 1.3). If the genetic markers used in non-invasive mark–recapture do not have the power necessary to distinguish between individuals, the population size estimate will be negatively biased due to the "shadow" effect (Mills et al. 2000; Waits and Leberg 2000; Waits 2004). When individuals are misidentified due to genotyping errors, caused by allelic drop out, false alleles, and contamination, the population size estimate will be positively biased due to the excess "ghosts" created (Waits and Leberg 2000; Creel et al. 2003; Waits 2004; Yoshizaki 2007). Simulations show that non-invasive genetic mark–recapture methods can overestimate population sizes by 200% when amplifying 7–10 loci with an error rate of 5% per locus (Waits and Leberg 2000). A scat study estimated the population size of wolves in Yellowstone National Park to be 550% greater than the known size of the population (Creel et al. 2003). Researchers try and reduce misidentification error rates in order to decrease the bias in population size estimates. The methods used to reduce misidentification rates usually do not completely eliminate misidentification, so the resulting population size estimates are still biased. Even with sample misidentification rates as low as 5%, simulation studies show that the population size estimates are still biased when data are analyzed under traditional mark–recapture models (Yoshizaki 2007).

In addition to reducing the rate of misidentification, researchers also attempt to model misidentification in order to reduce the bias in population size estimates. These misidentification models may reduce some of the bias in population size estimates (Yoshizaki 2007), but these models are not ideal. Misidentification models often assume that the misidentification rate is constant between individuals and between samples and that the errors that occur are independent of each other. The rate of genotyping errors can vary between individuals (Gagneux et al. 1997). The rate of misidentification due to undetected allelic drop out probably also varies between individuals. Individuals with higher levels of heterozygosity have more loci where undetected allelic drop out could occur. Individuals with larger alleles may be more prone to genotyping errors since larger alleles are more prone to allelic drop out (Tully 1993; Whitaker et al. 2001) and the rate of stutter increases with the number of tandem repeats in the allele (Shinde et al. 2003). In addition to variation between individuals, misidentification rates vary between samples. The amount of template DNA varies considerably between samples (Taberlet and Luikart 1999), which results in a large variation in error rates between samples (Gagneux et al. 1997; Creel et al. 2003; Paetkau 2003). For scat samples, the rate of genotyping error decreases with fresher samples (Ruell and Crooks 2007). For hair samples, the rate of genotyping error decreases with an increase in the number of hairs from an individual (Gagneux et al. 1997; Waits and Paetkau 2005). Errors that occur in different samples from a single individual are not independent (Paetkau 2003). These repeated errors can be caused by undetected technical errors that affect all the samples in a run (Paetkau 2003). In addition, larger alleles are more prone to stutter and allelic drop out which may result in a lack of independence between errors in different samples from a single individual. Many attempts are made to reduce the bias in population size estimates; however, the bias is not completely eliminated in most non-invasive genetic mark–recapture studies.

1.4 The New Probabilistic Allele Calling Method

To achieve better population size estimates from non-invasive genetic mark–recapture analysis, I propose a new allele calling method that assigns probabilities to possible allele calls rather than declaring a definitive allele call. In the traditional allele calling method, definitive allele calls are made independently for each PCR replicate of a sample. Then the collection of definitive allele calls is examined to determine the sample's genotype. By independently calling alleles at each PCR replicate, the traditional method does not utilize all of the information available in the collection of PCR replicates for a sample. Genotyping errors can be reduced by examining the peak heights and/or peak areas on the electropherograms for each PCR replicate to determine the confidence in the allele calls. Then, for any PCR replicate with low confidence, information from the other PCR replicates for that sample can be used to determine the allele call at that PCR replicate.

In human criminal forensics, mixtures of DNA from multiple sources are often analyzed by assigning probabilities to possible genotypes; however, peak intensities are rarely formally considered. Probabilistic examination using peak intensities to assess allelic drop out and stutter can improve forensic analysis of mixtures when allele calls are ambiguous (Evett et al. 1998; Gill et al. 2006). In addition, when attempting to match a sample from a suspect to a sample from a crime scene, these single contributor samples can be analyzed probabilistically using peak heights to determine likelihood ratios or match probabilities when allelic drop out is likely (Buckleton and Triggs 2006).

When assessing stutter probabilistically, the ratio of the height of the possible stutter peak to the peak height of the parent allele indicates whether the peak is a stutter peak or a true allele—the smaller the ratio, the more likely it is that the peak is a stutter peak. When assessing allelic drop out probabilistically, the height of the remaining allele indicates the probability of drop out of its partner allele—the smaller the height of the remaining allele, the more likely it is that allelic drop out has occurred (Gill et al. 2000; Whitaker et al. 2001; Gill et al. 2006).

Figure 1.5 shows two examples of why assigning probabilities, rather than using definitive allele calls, is beneficial. It shows four example electropherograms, where (A) and (C) both show allele calls with no ambiguity, and (B) and (D) are both ambiguous. Using the traditional allele calling method (A) would be called identical to (B) and given equal weight, as would (C) and (D). Using a probabilistic method, the allele calls of electropherograms (B) and (D) would be allowed to remain uncertain until the other PCR replicates of that sample were examined.

I propose using probabilistic analysis of genotypes for non-invasive genetic mark–recapture data. The probabilistic analysis assigns probabilities to allele calls based on electropherogram peak heights, rather than using definitive allele calls. For each PCR replicate, probabilities are assigned to possible allele calls based on electropherogram peak heights. For cases of possible allelic drop out, a portion of the probability distribution for the PCR replicate is assigned to a heterozygous allele call with one undesignated allele. Using this probabilistic method, uncertainty remains in allele calls until all the PCR replicates of a sample are examined. At each locus, the probabilities for possible allele calls of a sample, including allele calls with undesignated alleles, are averaged from the PCR replicates. Then, possible allele calls with undesignated alleles are assigned based on the frequency of the alleles in the averaged probabilities. The sample's genotype is assigned as the genotype with the highest probability. The probabilistic method is a quantitative way to take into account all of the information from the PCR replicates for a sample to determine the sample's genotype.

To examine the effectiveness of the proposed probabilistic allele calling method, I coded computer simulations to compare population size estimates using the new probabilistic allele calling method to estimates using a traditional allele calling method. For each simulation scenario, a population was generated and sampled using non-invasive genetic mark–recapture methods. Each sample, which contained low quality and quantity DNA, was genotyped at multiple microsatellite loci, with multiple PCR replicates for each locus. Genotypes were determined for samples using the probabilistic method and the traditional method. The resulting genotypes were matched and the capture history data was analyzed using four traditional closed mark–recapture models.

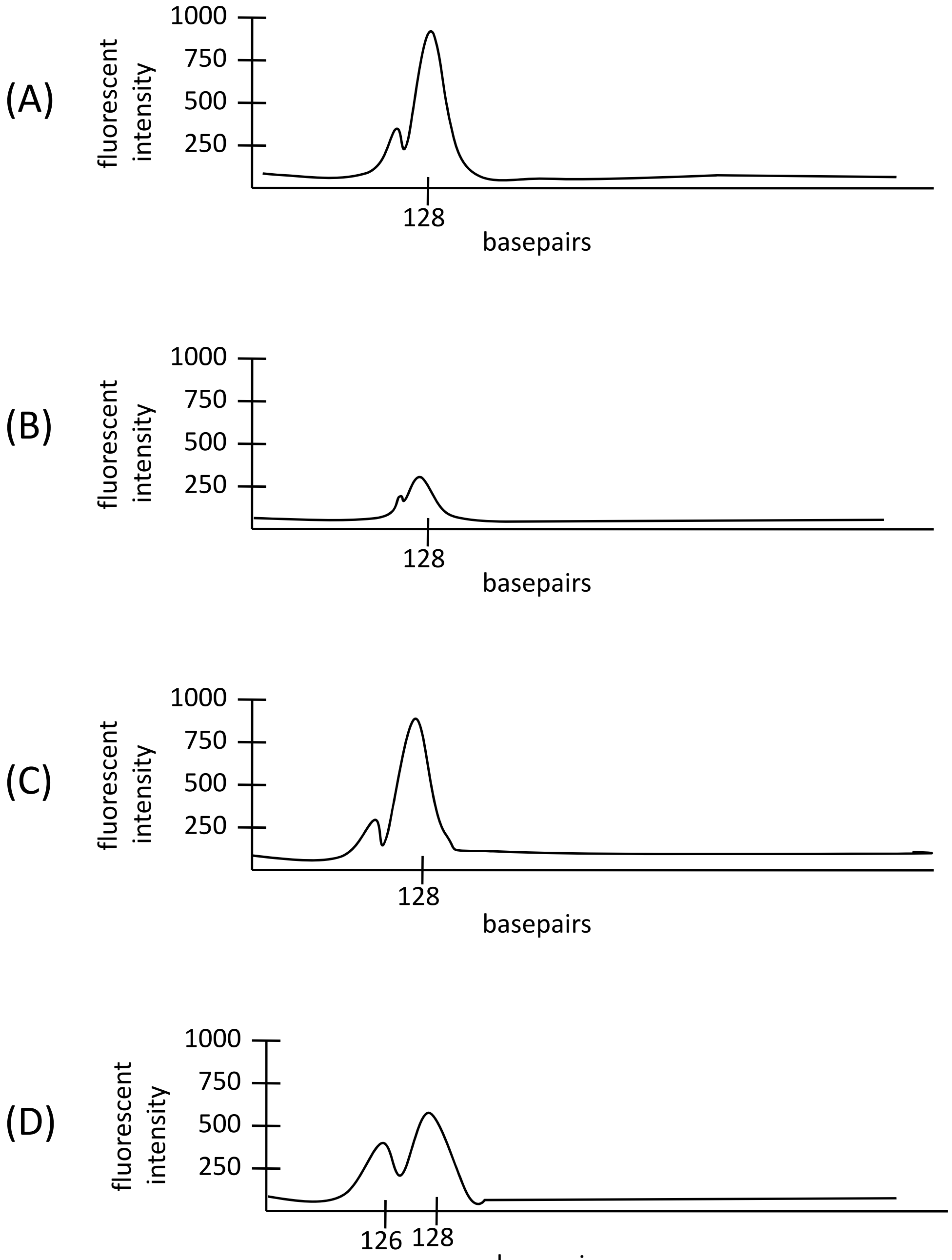


Figure 1.5: Example electropherograms. (A) shows a probable homozygote. (B) shows an ambiguous allele call—a possible homozygote or a possible case of allelic drop out. (C) shows a probable homozygote with a stutter peak. (D) shows an ambiguous allele call—a possible heterozygote or a possible homozygote with a large stutter peak.

Chapter 2

Methods

To simulate non-invasive genetic mark–recapture studies, I coded simulations in R (R Development Core Team 2005) (see Appendix A for code). The simulation code automatically reads an input file containing values specifying population parameters, details for the sampling procedures, and details for the analysis of the simulated data (see Appendix B for parameter input file). The code automatically reads in a separate file containing the sizes and frequencies of alleles in the population at each locus, including any null alleles (see Appendix C for allele input file). Based on these input files, non-invasive genetic mark–recapture data are simulated. The simulated data are analyzed using a traditional allele calling method and the new probabilistic allele calling method (see Figure 2.1 for an overview).

2.1 Data Simulation

Non-invasive genetic mark–recapture data are simulated in three sequential modules—the Population Generation Module, the Sampling Module, and the Genotyping Module. The Population Generation Module generates a closed population exhibiting heterogeneity in capture probabilities. Each individual is assigned a genotype and a capture probability. The Sampling Module selects samples based on non-invasive genetic mark–recapture methods. The Genotyping Module generates electropherograms for each PCR replicate based on a simulation of the biochemical processes of DNA extraction and PCR amplification modified from Gill et al. (2005).

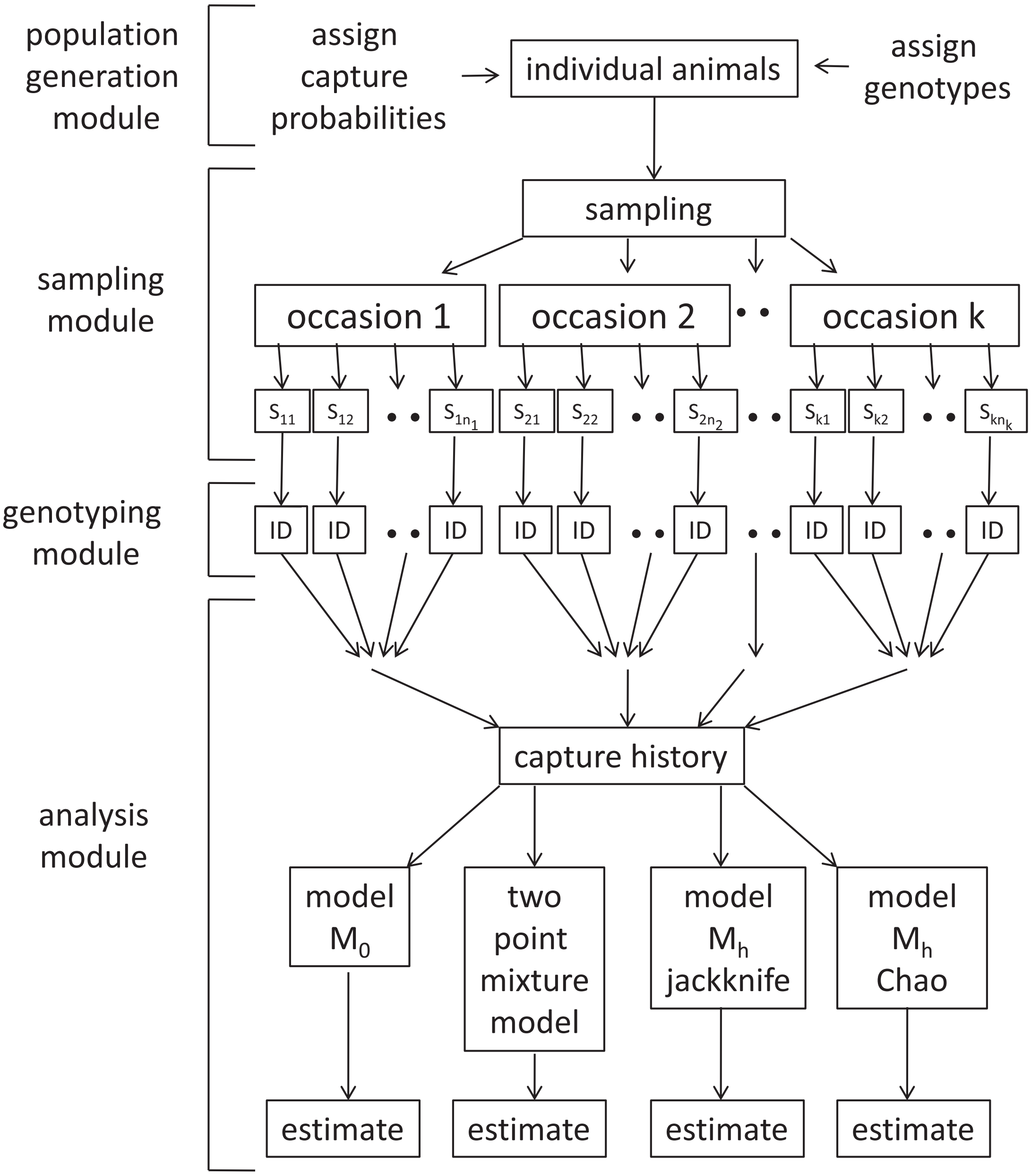


Figure 2.1: Overview of the modules used to generate and analyze data. The "S" denotes a non-invasive genetic sample, such as a single scat or hair sample, indexed by occasion number and sample number within the occasion. "ID" denotes identification by genotyping.

2.1.1 Population Generation Module

A closed population of a specified number of individuals (N) is generated. For each individual in the population a true genotype is randomly assigned based on input allele frequencies. For each individual, at each locus, two alleles are randomly sampled, with replacement, from the frequency distribution of alleles at that locus. Assigning alleles using this method generates a population with Hardy–Weinberg proportions.

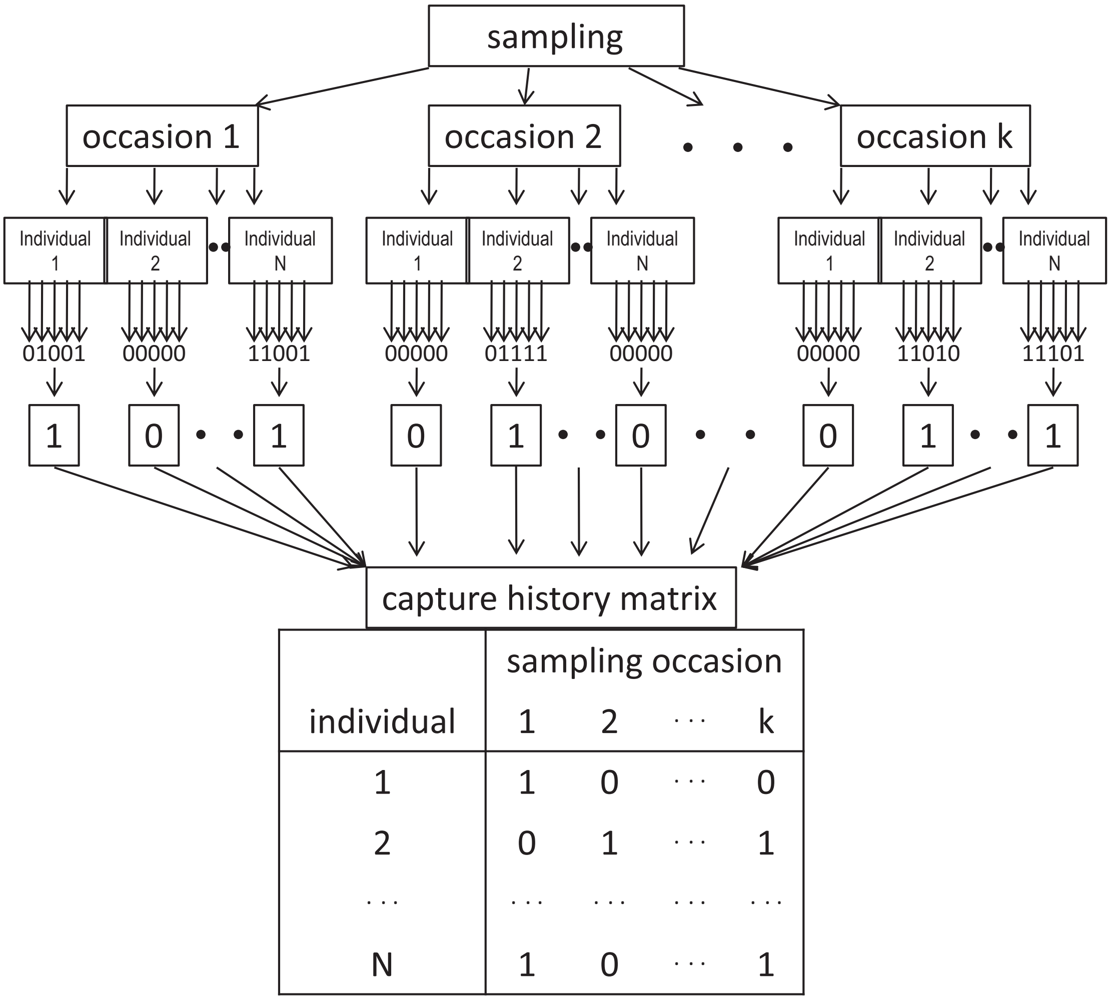
To simulate a population with heterogeneous capture probabilities, a capture probability (p) is randomly assigned to each individual in the population. This is the probability of capture per attempt that is tested against during each capture attempt in the Sampling Module to see if the animal is captured. The probabilities of capture per attempt are assigned to achieve a specified mean and variance of the probability of being captured at least one time during a sampling occasion. The mean and variance of the probability of capture per attempt are calculated by first determining the 2.5 percentile and 97.5 percentile for the probability of being captured at least once in a sampling occasion. These percentiles are then transformed to the 2.5 percentile and 97.5 percentile for the probability of being captured during a single capture attempt, from which the mean and variance for the probability of being captured at a single capture attempt are calculated. Each individual's capture probability is randomly assigned from a beta distribution with shape parameters calculated to achieve the calculated mean and variance of the probability of capture per attempt. The capture probabilities vary only by individual; the capture probabilities do not vary in time, nor do they vary in response to previous captures.

2.1.2 Sampling Module

For a specified number of sampling occasions (k), each individual generated in the Population Module is tested for capture for a specified number of possible capture attempts (s) (see Figure 2.2). Multiple capture attempts are used because, unlike traditional mark–recapture sampling where an individual can only be captured once, non-invasive sampling allows multiple captures within each sampling occasion. Each test is performed by sampling from a uniform (0,1) distribution and testing the resulting number to see if it is less than the individual's capture probability, which would result in a capture. For each capture that occurs, a single sample is collected.

2.1.3 Genotyping Module

For each sample collected in the Sampling Module, an electropherogram is generated for each PCR replicate, at each locus, based on a simulation of the biochemical processes of DNA extraction and PCR amplification modified from Gill et al. (2005). For each sample collected in the Sampling Module, the Genotyping Module simulates sample collection, DNA extraction, PCR amplification, and capillary electrophoresis (see Figure 2.3).



	sampling occasion			
individual	1	2	···	k
1	1	0	···	0
2	0	1	···	1
···	···	···	···	···
N	1	0	···	1

Figure 2.2: Diagram of the sampling process for N individuals, with k sampling occasions and five capture attempts per individual per occasion.

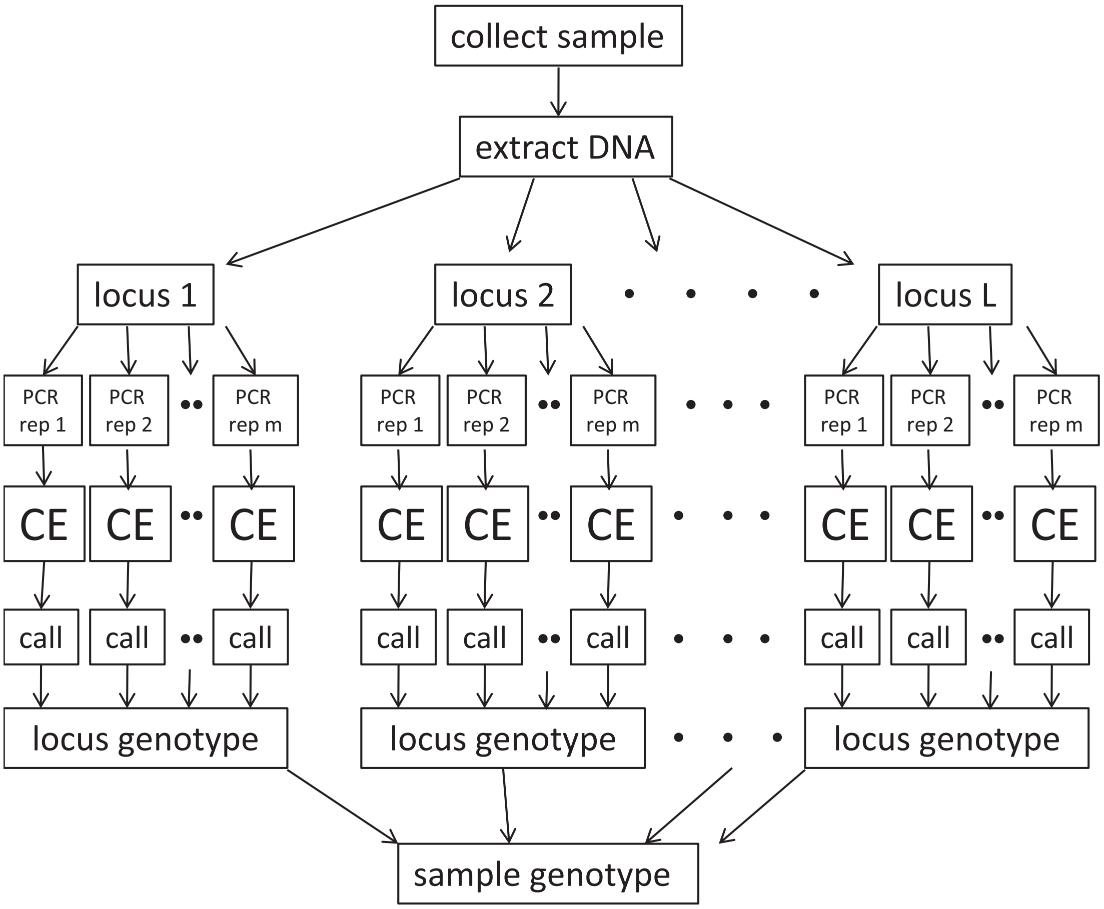


Figure 2.3: Diagram of the genotyping process for a single sample. L loci are examined, with m PCR replicates at each locus. “PCR rep” denotes a replication of the PCR amplification process, “CE” denotes visualization by capillary electrophoresis, and “call” denotes allele calling.

The quantity of DNA in each sample is simulated by assigning the number of cells present in the sample. The number of cells in a sample is randomly determined from a uniform distribution, with specified minimum and maximum number of cells possible. For each sample, at each locus, the number of double stranded DNA molecules of each allele type is calculated based on the number of cells present and the number of alleles of each type in the true genotype.

To simulate DNA degradation, a quality score is randomly assigned to each sample by sampling from a uniform (0,1) distribution. Degradation decreases the number of DNA molecules remaining in a sample. Lower quality samples will lose more DNA molecules than higher quality samples, and longer alleles will lose more DNA molecules than shorter alleles. The number of molecules available for the genotyping process depends on the degradation efficiency (the probability that a single molecule will not be affected by degradation and will thus be available for extraction).

I determined the degradation efficiency by assuming that any break in the DNA molecule results in the molecule not amplifying. Assuming that the probability of a break at any specific nucleotide is constant for a sample, then the probability of being unable to amplify a single DNA molecule is a function of the number of nucleotides in the molecule. I assumed that the probability of x number of breaks in the DNA follows a binomial distribution

$$P(X = x|seq, p_{break}) = \binom{seq}{x}(p_{break})^x(1 - p_{break})^{seq-x} \tag{2.1}$$

where seq is the number of nucleotides in the DNA molecule, and p_{break} is the probability of degradation causing a break in the DNA at a specific nucleotide. The probability of successfully amplifying a DNA molecule is equal to the probability of no breaks in the DNA molecule, which is given by

$$P(X = 0|seq, p_{break}) = (1 - p_{break})^{seq} \tag{2.2}$$

where p_{break} is again the probability of degradation causing a break in the DNA at a specific nucleotide. I assume p_{break} is proportional to one minus the quality score for the sample. This results in a probability of successful amplification of a DNA molecule equal to

$$P(X = 0|seq, p_{break}) = (1 - \alpha(1 - qual))^{seq} \tag{2.3}$$

where α is a constant of proportionality. To determine α, I solved equation 2.3 for p_{break} so that an average quality sample ($qual = 0.5$) showed the expected 12–15% difference in amplification success for allele lengths differing by 100 base pairs (bp) (Buchan et al. 2005). From the possible solutions, I chose a p_{break} so that small alleles from high quality samples ($size = 100$, $qual = .9$) would have amplification success rates greater than 90% and large alleles from low quality samples ($size = 300$, $qual = .1$) would have amplification success rates less than 50%. Based on these constraints, I chose the constant of proportionality to be $\alpha = 0.0035$. This resulted in a degradation efficiency of

$$deg_{eff} = (1 - (0.0035)(1 - qual))^{size} \tag{2.4}$$

where deg_{eff} is the degradation efficiency, $qual$ is the quality score, and $size$ is the length of the allele. The number of DNA molecules of each allele remaining in a sample is simulated by a random binomial distribution, where the number of trials is the number of DNA molecules of that allele, and the probability of success is the degradation efficiency.

Once the amount of usable DNA in each sample is determined, DNA extraction is simulated. Extraction is simulated for each allele of each PCR replicate. DNA extraction is simulated by randomly sampling from a binomial distribution, where the number of trials is the number of DNA molecules of that allele in the sample, and the probability of success is the specified extraction efficiency (the probability that a single molecule will be included in the extract). A portion of each simulated DNA extract is aliquoted for processing. The number of extracted DNA molecules of each allele in the aliquot is determined by randomly sampling from a binomial distribution, where the number of trials is the number of DNA molecules of that allele in the extract, and the probability of success is the specified aliquot efficiency (the probability that a single molecule will be included in the aliquot).

Undetected contamination is simulated by randomly generating a value from a uniform (0,1) for each PCR replicate and comparing it to the specified probability of undetected contamination. Undetected contamination occurs in the simulation when the random number is less than the probability of undetected contamination. If undetected contamination occurs, a single double stranded DNA molecule is added to the DNA aliquot at a random allele chosen from alleles present at that locus in the population. Contamination from detectable sources, such as contamination from lab materials or from unrelated species, is not simulated since it can be detected by standard laboratory procedures. Null alleles, which are alleles that do not amplify, are simulated by setting the number of DNA molecules present in the aliquot to zero for all null alleles.

To simulate PCR amplification of small quantities of DNA, 34 PCR cycles, including stutter, are simulated for each PCR replicate. Ten PCR replicates are simulated for each locus of each sample. PCR success rates vary between loci (Buchan et al. 2005), so for each locus a PCR efficiency (the probability that a molecule will be copied during a single PCR cycle) is randomly generated from a beta distribution with shape parameters set to achieve a specified mean and variance. Stutter rates vary between loci (Gill et al. 2005), so for each locus a stutter efficiency (the probability that a newly created molecule will be one repeat shorter than the molecule it was created from) is randomly generated from a uniform distribution with specified minimum and maximum possible stutter efficiencies.

For each PCR cycle, amplification of each allele present in the aliquot is simulated by a random binomial, where the number of trials is the number of DNA molecules of that allele, and the probability of success is the locus specific PCR efficiency. Some of the newly amplified DNA molecules will be in a stutter position, one repeat shorter than the actual allele. Stutter is simulated for each allele by randomly sampling from a binomial distribution, where the number of trials is the number of new DNA molecules of that allele amplified during that PCR cycle, and the probability of success is the locus specific stutter efficiency. Stutter molecules are available for PCR amplification during subsequent PCR cycles. Stutter expansions are not simulated because they are relatively rare (Shinde et al. 2003).

The simulation of the biochemical processes of DNA extraction and PCR amplification results in a simulated electropherogram for each PCR replicate. Genotypes are determined from the simulated data using a traditional allele calling method and the new probabilistic allele calling method.

2.1.4 Assumptions of Data Simulations

The simulation of non-invasive genetic mark–recapture data assumes that all individuals have a genotype and individuals can, by chance, have the same genotype. Random binomials were used to simulate the PCR process, so assumptions of the binomial distribution must be met. This includes constant efficiency parameters for each trial and independence of DNA molecules (Gill et al. 2005). It is assumed that degradation results in a decrease in the amount of template DNA. The number of DNA molecules in the PCR product is assumed to be a good indicator of peak height on an electropherogram.

Misidentification can occur on initial capture or recapture. Misidentification does not need to result in a new individual, although it is much more likely that it will. The same misidentification can occur more than once and two different individuals can be misidentified as the same non-existent ghost. Multiple ghosts can be produced by the same single individual at a single sampling occasion.

2.2 Analysis of Simulated Data

The simulated data are inputted into the analysis modules, which automatically analyze the data. The analysis consists of three sequential modules—the Allele Calling and Genotyping Module, the Sample Matching and Capture Histories Module, and the Population Size Estimation Module. The Allele Calling and Genotyping Module uses both a definitive allele calling method and a probabilistic allele calling method. The Sample Matching and Capture Histories Module matches samples and generates capture histories. The Population Size Estimation Module inputs the capture histories into four traditional mark–recapture models, each one yielding a population size estimate.

2.2.1 Allele Calling and Genotyping Module

Traditional and probabilistic allele calling both begin by assigning probabilities to allele calls for each PCR replicate based on electropherogram peak heights. Small stutters and noise are removed. Heterozygote balance, which is the ratio of the height of the potential stutter peak to the height of the parent peak, is calculated for potential stutter peaks. Any peak that is one repeat shorter than another peak, and whose heterozygote balance is less than a specified minimum value, is assumed to be a stutter peak, and the stutter peak height is reset to zero. Any peak below the specified noise threshold value is also reset to zero.

Next, the heterozygous and homozygous threshold values, which help control the amount of variability in the allele calls, need to be determined. The heterozygous threshold is the peak height above which an allele is assumed to be a true allele. The heterozygous threshold value of allele i is a linear function of the size of allele i with specified slope and intercept values. The homozygous threshold is the peak height above which an allele is assumed not to be associated with another allele that has dropped out. The heterozygous threshold value of allele i is a linear function of the size of allele i with specified slope and intercept values.

If no peaks are present for a PCR replicate, the allele call is assigned to *(Z,Z)* with a probability of one, where Z indicates an undesignated allele. If one or more peaks are present for a PCR replicate, the probabilities of potential allele calls are determined by the peak heights on the electropherogram. Calculating the probability of potential allele calls begins by calculating the probabilities of true alleles and allelic drop out. The probability that allele i is a true allele is calculated by

$$p(t_i) = \begin{cases} \frac{ht_i}{hrt_i} & \text{if } ht_i < hrt_i \\ 1 & \text{if } ht_i \geq hrt_i \end{cases}$$

where ht_i is the peak height for allele i, and hrt_i is the heterozygous threshold value for allele i. The probability that an unknown allele dropped out is a function of the height of allele i remaining in the electropherogram, and is calculated by

$$p(d_i) = \begin{cases} 1 - \frac{ht_i}{hmt_i} & \text{if } ht_i < hmt_i \\ 0 & \text{if } ht_i \geq hmt_i \end{cases}$$

where ht_i is the peak height for allele i, and hmt_i is the homozygous threshold value for allele i.

The following two equations (2.5–2.6) are used to calculate the probabilities of potential allele calls when only one peak is present in the electropherogram. The probability of a homozygous genotype, with allele i, is given by

$$p(i, i) = 1 - p(d_i) \tag{2.5}$$

and the probability of a heterozygous genotype, with allele i and an undesignated allele Z, is given by

$$p(i, Z) = p(d_i). \tag{2.6}$$

The following four equations (2.7–2.10) are used to calculate the probabilities of potential allele calls when more than one peak is present on the electropherogram. The probability of a homozygous genotype, with allele i, is given by

$$p(i, i) = (1 - p(d_i))p(t_i) \prod_{j \neq i}(1 - p(t_j)) \tag{2.7}$$

where j is any allele other than i. The probability of a heterozygous genotype, with allele i and an undesignated allele Z, is given by

$$p(i, Z) = p(d_i)p(t_i) \prod_{j \neq i}(1 - p(t_j)) \tag{2.8}$$

where j is any allele other than i. The probability of a heterozygote, with known alleles a and b, is initially set to

$$p_0(a,b) = \frac{ht_a}{hrt_a}\frac{ht_b}{hrt_b}. \tag{2.9}$$

The sum of the probabilities of all heterozygotes with known alleles (any combination of a and b except $a \neq b$) is given by

$$\sum_{a \neq b}\sum_{b \neq a} p(a,b) = 1 - \sum_i p(i,i) - \sum_i p(i,Z) \tag{2.10}$$

where i, a, b are any alleles. This summation value is used to normalize the initial probabilities for heterozygotes ($p_0(a,b)$), which were calculated using equation 2.9.

After the initial probabilities are assigned to potential allele calls, the probabilities are altered to take into account potential stutter peaks. A peak is determined to be a potential stutter peak, rather than a true allele, if the peak is at an allele position one repeat smaller than a taller peak. If allele i is a possible stutter peak, then each time the stutter allele occurs in a potential allele call, the probability of that allele call is multiplied by

$$\frac{ht_i}{ht_{i+1}}$$

where ht_i is the peak height for the possible stutter peak at allele i, and ht_{i+1} is the peak height for the allele *i+1*, which is the parent allele associated with the possible stutter peak. The probability removed from allele calls due to possible stutter peaks is reassigned to allele calls without potential stutter peaks in proportion to the probabilities of the allele calls without potential stutter peaks.

Traditional Allele Calling

To simulate traditional allele calling, a definitive allele call is made for each PCR replicate, at each locus, for each sample, based on the probabilities determined for that single PCR replicate. Undesignated alleles (Z) are assigned to known alleles by adding the probability of an allele call of (*i,Z*) to the probability of an allele call of (*i,i*). The probability of an allele call of (*Z,Z*) is not reassigned to known alleles.

The definitive allele call for a PCR replicate is determined by the allele call with the highest probability for that PCR replicate after undesignated alleles are assigned. When each PCR replicate has a definitive allele call, a consensus genotype is determined for each sample for the range of PCR replicates. For one PCR replicate, the consensus genotype for each sample will simply be the alleles called for the first PCR replicate. For two PCR replicates, a consensus genotype will be determined from the first and the second PCR replicates. For three or more PCR replicates a consensus genotype will be determined by that PCR replicate and all the previous replicates. I used an allele acceptance threshold, which indicates how many times an allele must be seen before the allele is called as part of the genotype (Valiere et al. 2002). The allele acceptance threshold was set to two-thirds the number of PCR replicates, rounded to the nearest integer.

Samples with exactly one allele, *i*, above the acceptance threshold will be assigned to a genotype of (*i*,*i*). Samples with exactly two alleles, *i* and *j*, above the acceptance threshold will be assigned to a genotype of (*i*,*j*). Samples with more than two alleles above the acceptance threshold will be assigned to a genotype of a heterozygote for the two alleles that were called most often for that sample during that round of PCR replications. If this results in the selection of more than two alleles, due to alleles being called an equal number of times, then the two largest size alleles are assigned as the genotype. Samples with no alleles above the acceptance threshold will be assigned based on the alleles that were called most often, but that did not make it to the acceptance threshold. If one allele was called most often, the genotype is assigned as a homozygote for that allele. If two alleles were called most often, the genotype is assigned as a heterozygote for those two alleles. If more than two alleles were called most often, then the two largest size alleles are assigned as the genotype. If no alleles were called, the genotype is assigned as a homozygote for the most common allele in the population.

Each sample's overall consensus genotype is the collection of consensus genotypes determined at each locus for that sample. From the overall consensus genotypes for all samples, a single capture history matrix is generated and inputted into four mark–recapture models, yielding four population size estimates.

Probabilistic Allele Calling

For each sample, the genotype probabilities are determined from the probabilities assigned to allele calls for all PCR replicates at each locus. The probabilities of allele calls are averaged for varying numbers of PCR replicates. For one PCR replicate, the genotype probabilities for each sample will simply be the allele call probabilities for the first PCR replicate. For two PCR replicates, the genotype probabilities will be determined by averaging the first and the second PCR replicates. For three or more PCR replicates, the genotype probabilities will be determined by averaging that PCR replicate and all the previous replicates. These genotype probabilities include probabilities for genotypes with undesignated alleles. The undesignated alleles are assigned to known alleles based on the allele frequencies calculated from the genotype probabilities of known alleles for that sample.

Each sample's probabilistically determined genotype is the genotype that is most probable based on the averaged allele call probabilities after undesignated alleles are assigned. From the probabilistically determined genotypes for all samples, a single capture history matrix is generated and inputted into four mark–recapture models, yielding four population size estimates.

2.2.2 Sample Matching and Capture Histories Module

Once each sample has been assigned two genotypes, one for each allele calling method, the samples are matched to generate a capture history matrix for each allele calling method. Two samples match when the samples have identical genotypes. Once samples are matched, two capture histories are determined for each individual, one history for each allele calling method. An individual is considered "captured" at a sampling occasion if at least one sample from that occasion belongs to the individual; if no samples from that occasion belong to the individual, the individual is considered "not captured" for that occasion. The capture histories of all individuals are then collected into two capture history matrices, one for each allele calling method. These two capture history matrices are inputted into four mark–recapture models, yielding four population size estimates for each allele calling method.

2.2.3 Population Size Estimation Module

The capture history matrices from the two methods are each inputted into four closed mark–recapture models—model M_0 (Williams et al. 2002), a two-point mixture model (Pledger 2000), model M_h first order jackknife (Burnham and Overton 1978), and model M_h modified first order Chao (Chao 2006). I do not consider models where capture probabilities varied in time or in response to previous capture.

Model M_0 has two parameters: the population size (N) and the probability of capture (p). Population size estimates from model M_0 (Williams et al. 2002) are obtained by maximizing the likelihood function given the data (x_ω). The likelihood function of model M_0 is given by

$$L(N, p \mid x_\omega) = \frac{N!}{(\prod\limits_{\omega} x_\omega!)(N - \sum x_\omega)!} p^n (1-p)^{(kN-n)} \tag{2.11}$$

where $\sum x_\omega$ is the number of unique individuals captured, n is the total number of captures (not including repeat captures of a single individual at a sampling occasion), and k is the number of sampling occasions (Williams et al. 2002). The likelihood is maximized when

$$\hat{p} = \frac{n}{kN} \tag{2.12}$$

(Williams et al. 2002). Plugging this equation for $\hat{p}$ (equation 2.12) into the likelihood function (equation 2.11), yields a likelihood function with a single parameter N. Removing factors that are functions of the data only, which are therefore constant, yields

$$L(N \mid x_\omega) \propto \frac{N!}{(N - \sum x_\omega)!} \left(\frac{n}{kN}\right)^n \left(1 - \frac{n}{kN}\right)^{(kN-n)}. \tag{2.13}$$

The estimate of the population size is constrained to being greater than or equal to the number of unique individuals captured by using a natural log transformation, which maps to the real line. In order to avoid boundary problems the population size is transformed using

$$N_{transformed} = ln(N - (\sum x_\omega - 1)) \tag{2.14}$$

where $\sum x_\omega$ is the number of unique individuals capture. The maximum likelihood estimator of N is obtained by minimizing the negative log likelihood of the transformation of equation 2.13 using the "nlm" function in R, with $\sum x_\omega$ (the number of unique individuals captured) as the initial estimate of N.

Estimates are obtained from the three models with individual heterogeneity by first calculating the capture frequencies. The capture frequencies are represented by a vector where the i^{th} coordinate is the number of individual animals that were captured exactly i times. which are the number of individuals captured exactly i times. The vector of capture frequencies is a sufficient statistic for the population size estimators from the heterogeneity models (Norris and Pollock 1996). The two point mixture model has four parameters: the population size (N), the probability of capture of group one (p_1), the probability of capture of group two (p_2), and the proportion of individuals in group one (π). Population size estimates from the two point mixture model (Pledger 2000) are obtained by maximizing the likelihood function given the data (x_ω). The likelihood function of the two point mixture model is given by

$$\begin{aligned} L(N, p_1, p_2, \pi \mid x_\omega) &= \frac{N!}{(\prod_\omega x_\omega!)(N - \sum x_\omega)!} \\ &\quad \left(\pi(1-p_1)^k + (1-\pi)(1-p_2)^k\right)^{(N-\sum x_\omega)} \\ &\quad \prod_{i=1}^{k} \left(\pi {p_1}^i (1-p_1)^{(k-i)} + (1-\pi){p_2}^i(1-p_2)^{(k-i)}\right)^{f_i} \end{aligned} \tag{2.15}$$

where k is the number of sampling occasions and f_i is the number of individuals captured exactly i times. Removing factors which are functions of the data only, which are therefore constant, yields

$$\begin{aligned} L(N, p_1, p_2, \pi \mid x_\omega) &= \frac{N!}{(N - \sum x_\omega)!} \\ &\quad \left(\pi(1-p_1)^k + (1-\pi)(1-p_2)^k\right)^{(N-\sum x_\omega)} \\ &\quad \prod_{i=1}^{k} \left(\pi {p_1}^i (1-p_1)^{(k-i)} + (1-\pi){p_2}^i(1-p_2)^{(k-i)}\right)^{f_i} . \end{aligned} \tag{2.16}$$

The estimate of the population size is constrained to being greater than or equal to the number of unique individuals captured by using the same natural log transformation as for model M_0 (see equation 2.14). The two capture probabilities and the proportion of animals in each group are constrained to be between 0 and 1 by using a logit transformation, which maps probabilities to the real line. The probabilities were transformed using

$$\hat{p}_{transformed} = ln\left(\frac{\hat{p}}{1-\hat{p}}\right) \tag{2.17}$$

where $\hat{p}$ is the probability estimate. The maximum likelihood estimator of N is obtained by minimizing the negative log likelihood of the transformation of equation 2.16 using the "nlm" function in R, with initial estimates of $\hat{N} = \sum x_\omega$, $\hat{p_1} = 0.4$, $\hat{p_2} = 0.6$, and $\hat{\pi} = 0.5$.

The population size estimate for model M_h first order jackknife (Burnham and Overton 1978) is given by

$$\hat{N} = \sum x_\omega + \frac{k-1}{k} f_1 \tag{2.18}$$

where $\sum x_\omega$ is the number of unique individuals captured, k is the number of sampling occasions, and f_1 is the number of individuals captured exactly one time (Burnham and Overton 1978).

The population size estimate for model M_h modified first order Chao (Chao 2006) is given by

$$\hat{N} = \sum x_\omega + \frac{f_1(f_1 - 1)}{2(f_2 + 1)} \tag{2.19}$$

where $\sum x_\omega$ is the number of unique individuals captured, and f_i is the number of individuals captured i times.

The simulations are run multiple times, resulting in multiple population size estimates for each allele calling method under each model. Any population size estimate greater than three times the true population size is considered unreasonable and is removed from further analysis. The number of simulations that did not converge on an estimate, or produced an unreasonable estimate, were recorded for each scenario, for each number of PCR replicates, under each model. For each allele calling method, under each model, a mean population size estimate and standard deviation are determined from the multiple simulations.

2.2.4 Assumptions of Data Analysis

The analysis of data assumes that the peak height is linearly proportional to the number of DNA molecules. It assumes that if an individual is a heterozygote, the two copies of the allele are independent and the contribution of the alleles is additive. The relationship between the probability of drop out and the height of the remaining peak is assumed to decline linearly until it reaches the threshold where the probability equals zero. The relationship between the probability of a true allele and the height of the peak is assumed to increase linearly until it reaches the threshold where the probability equals one. Peak heights are assumed to decrease in a linear fashion as the length of the allele increases. The analysis assumes that the assumptions of the mark–recapture models are met and that the genetic makers used have enough power to distinguish between individuals.

2.3 Simulation Scenarios

2.3.1 Simulation Inputs

To investigate how well the probabilistic allele calling method performs, I ran the simulation code under a number of different scenarios. The population size, capture probability (and its variance), and the number of sampling occasions varied. I used two sets of allele frequencies—one without null alleles and one with null alleles. All other input values did not change between scenarios (Table 2.1).

I ran the simulations with small and large population sizes, 200 or 500 individuals, respectively. I ran the simulations with low and high mean capture probabilities, 0.2 or 0.5, respectively. The low mean capture probability of $p = 0.2$ is similar to the mean capture probability estimate from a non-invasive genetic mark recapture study of bears using hair traps (Boulanger et al. 2004). I set the variance for the low capture probabilities to 0.002. I set the variance for the high capture probabilities to 0.020. Figure 2.4 shows the probability distribution functions from which the capture probabilities are sampled.

Samples were collected during 5 or 10 sampling occasions. To achieve good population size estimates using traditional closed mark–recapture models, it is recommended that at least 5 sampling occasions are used, preferably 7–10 sampling occasions (Otis et al. 1978). There were 5 capture attempts for each animal at each sampling occasion. The minimum and the maximum number of cells possible in the samples were set to 30 and 150 cells, respectively. This ensured that almost all of the samples had pre-PCR templates of less than 17 cells, which corresponds to 100 pg of DNA, which is considered low copy number and prone to allelic drop out (Gagneux et al. 1997; Morin et al. 2001; Whitaker et al. 2001; Butler 2005; Gill et al. 2005). I set the probability of undetected contamination to 0.008, which matches the probability of within population contamination found in a scat study of noninvasive genotyping errors in elephants (Buchan et al. 2005). I set the extraction efficiency to 0.46 and the aliquot efficiency to 0.30 (Gill et al. 2005). I used a PCR efficiency of 0.82 (Shinde et al. 2003; Gill et al. 2005), with a standard deviation of 0.06 to ensure that no locus had a PCR efficiency so low that it consistently failed to amplify. I set the stutter efficiencies to be between 0 and 0.4, which is the range found for dinucleotide repeat loci (Shinde et al. 2003).

I set the minimum heterozygote balance for a stutter peak to 32% (Ewen et al.

Table 2.1: Simulation input values.

parameter description	input value(s)	module
population size*	200, 500	population
mean probability of at least one capture per occasion*	0.2, 0.5	population
variance of probability of at least one capture per occasion*	0.002, 0.02	population
number of sampling occasions*	5, 10	sampling
number of capture attempts per sampling occasion*	5	sampling
minimum number of cells possible in the sample	30	genotyping
maximum number of cells possible in the sample	150	genotyping
probability of undetected contamination per PCR replicate	0.008	genotyping
extraction efficiency	0.46	genotyping
aliquot efficiency	0.30	genotyping
mean PCR efficiency	0.82	genotyping
standard deviation of PCR efficiency	0.06	genotyping
minimum possible stutter efficiency	0.0	genotyping
maximum possible stutter efficiency	0.04	genotyping
minimum heterozygote balance for a stutter peak	0.32	allele calling
noise threshold	$2\text{x}10^7$	allele calling
heterozygous threshold slope	$-1.1\text{x}10^7$	allele calling
heterozygous threshold intercept	$8.7\text{x}10^9$	allele calling
homozygous threshold slope	$-7.3\text{x}10^7$	allele calling
homozygous threshold intercept	$1.7\text{x}10^{10}$	allele calling

* indicates a parameter that varied between scenarios.

2000). I set the noise threshold to $2\text{x}10^7$ (Gill et al. 2005). I determined the heterozygous and homozygous threshold intercepts by starting with an input of 17 cells, which is at the high end of low copy number DNA. For both heterozygotes and homozygotes, I then calculated the expected number of amplified DNA molecules remaining after 34 PCR cycles when there was no degradation, the efficiency of the PCR cycle was 0.82, and 0.02 of new molecules were lost to stutter each cycle. The threshold intercepts were set to equal the expected number of DNA molecules (heterozygote: $8.7\text{x}10^9$, homozygote:$1.7\text{x}10^{10}$). I determined the heterozygous and homozygous threshold slopes using a similar method, but accounting for the increased effect of degradation on larger alleles. For both heterozygotes and homozygotes, the slopes were determined by calculating the number of amplified DNA molecules for a sample with a quality score of 0.5 and allele sizes of 100 bp and 300 bp. The difference between the the number of amplified molecules for the two different allele sizes

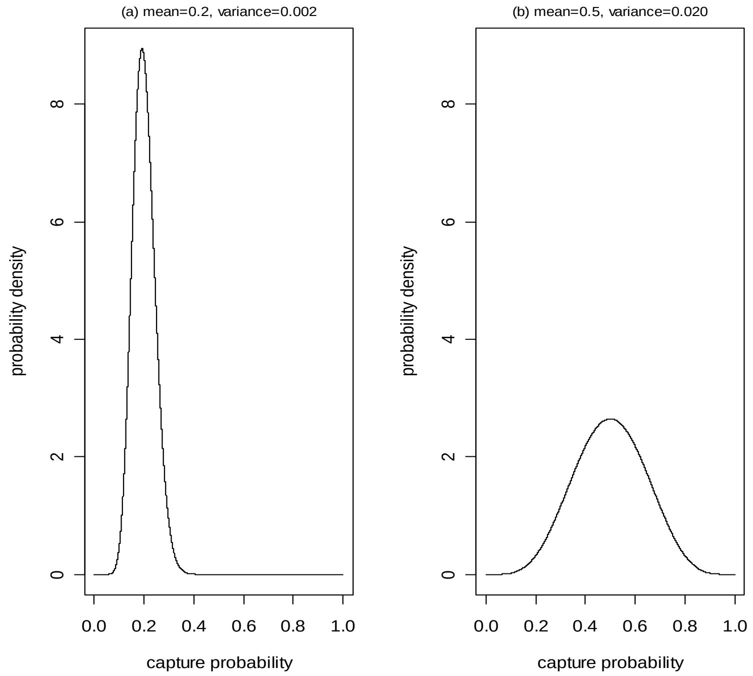


Figure 2.4: Probability distribution functions of beta distributions with shape parameters for (a) low capture probabilities (mean=0.2, variance=0.002) and (b) high capture probabilities (mean=0.5, variance=0.020).

was divided by the difference in allele sizes, which was 200, yielding the slopes (heterozygote: $-1.1\text{x}10^7$, homozygote: $-7.3\text{x}10^7$).

I used two sets of input allele frequencies (Appendix 2.2). I based both sets on a suite of 8 black bear nuclear microsatellite loci (G1A, G10B, G10C, G10L, G10M, G10P, G10X, G1D) (Paetkau et al. 1998), which are all dinucleotide repeat loci. For the first set, the allele frequencies were set to match data from a study of the "West Slope" population of Canadian black bears (Paetkau et al. 1998)(Table 2.2). The second set of input allele frequencies was modified from the first set to contain null alleles. I altered two alleles, at 2 different loci, to be null alleles. At locus G10B, allele 164 with frequency of 18% was modified to a null allele. At locus G1A, allele 198 with frequency of 8% was also modified to a null allele. Both sets of input allele frequencies have a high number of polymorphic loci, so each individual should have a unique genotype.

Table 2.2: Input allele frequencies.

locus	number of positive frequency alleles	range of alleles (bp)
G1A	7	180–200
G10B	5	140–166
G10C	8	99–117
G10L	12	135–171
G10M	10	196–222
G10P	7	139–167
G10X	12	125–163
G1D	9	172–190

2.3.2 Scenarios

I examined 16 different scenarios (Table 2.3), with each scenario simulated 100 times. For each scenario, I examined the mean population size estimate and the standard deviation, in the absence of genotyping errors, under the four models. For each scenario, under each model, I determined the number of PCR replicates required to achieve a mean population size estimate within 20% of the true population size estimate. For each scenario, under each model, I examined the mean population size estimates and the standard deviations for varying numbers of PCR replicates. I examined bias, calculated by

$$bias = \bar{\hat{N}} - N, \tag{2.20}$$

where $\bar{\hat{N}}$ is the mean of the population size estimates and N is the true population size. I also examined the root mean squared error (RMSE), calculated by

$$RMSE = \sqrt{(\bar{\hat{N}} - N)^2 + var(\hat{N})} \tag{2.21}$$

where $\bar{\hat{N}}$ is the mean of the population size estimates, N is the true population size, and $var(\hat{N})$ is the variance of the population size estimates.

I evaluated the performance of the four mark–recapture models by examining the bias, variance, and RMSE of the population size estimates under the models. I examined the effect that changing input parameters had on population size estimates. I examined changes in bias, variance, and RMSE for pairs of scenarios that had identical input value except for one varying parameter. For comparisons between simulations with large and

Table 2.3: Simulation scenarios.

simulation number	population size	mean capture probability	variance of capture probabilities	sampling occasions	presence of null alleles
1	200	0.2	0.002	5	no
2	200	0.2	0.002	5	yes
3	200	0.2	0.002	10	no
4	200	0.2	0.002	10	yes
5	200	0.5	0.020	5	no
6	200	0.5	0.020	5	yes
7	200	0.5	0.020	10	no
8	200	0.5	0.020	10	yes
9	500	0.2	0.002	5	no
10	500	0.2	0.002	5	yes
11	500	0.2	0.002	10	no
12	500	0.2	0.002	10	yes
13	500	0.5	0.020	5	no
14	500	0.5	0.020	5	yes
15	500	0.5	0.020	10	no
16	500	0.5	0.020	10	yes

small population sizes, I examined the relative bias, relative variance, and relative mean squared error (relative MSE), calculated by

$$relativeBIAS = (\bar{\hat{N}} - N)/N \tag{2.22}$$

$$relativeVAR = var(\hat{N})/N^2 \tag{2.23}$$

$$relativeMSE = RMSE^2/N^2 \tag{2.24}$$

where $\bar{\hat{N}}$ is the mean of the population size estimates, N is the true population size, $var(\hat{N})$ is the variance of the population size estimates, and $RMSE$ is the root mean squared error calculated from equation 2.21.

Chapter 3

Results

To compare the new probabilistic allele calling method to the traditional method, I ran simulations under different scenarios and the resulting capture history matrices were inputted into four closed mark–recapture models. To illustrate the inherent behavior of the traditional models, Table 3.1 shows the population size estimates in the absence of genotyping errors. As expected, model M_0 underestimated the true population size for all scenarios; however, the bias was small for many scenarios. The heterogeneity models performed well in most cases. The two point mixture model tended to have larger variances than the other two heterogeneity models. The performance of all the models varied somewhat with changes in the various parameters.

3.1 Comparison of Allele Calling Methods

The examination of the number of PCR replicates required to achieve a mean population size estimate within 20% of the true population size revealed that the probabilistic method always performed at least as well, and often times better, than the traditional method (Table 3.2). Examination of mean population size estimates, and the standard deviations of the estimates, for both allele calling methods under the four mark–recapture models at 5 and 10 PCR replicates show that the probabilistic method resulted in lower biases and lower variances than the traditional method in almost all situations (Tables 3.3 and 3.4). Even after extensive PCR replication, both methods, for almost all scenarios under all four models, had a considerable amount of residual bias.

Figures 3.1–3.2 show root mean squared errors (RMSEs) and mean population size estimates (±1 standard deviation) versus the number of PCR replicates examined for one simulation scenario under the four mark–recapture models. The qualitative results of most of the other scenarios were identical (see results for all simulation scenarios in Appendix D). The probabilistic method most often had lower biases, variances, and RMSEs than the traditional method when more than two PCR replicates were examined. The biases and the RMSEs of the probabilistic method usually exhibited a monotonic decrease as the number of PCR replicates increased. The traditional method usually showed an overall decline in biases and RMSEs as the number of PCR replicates increased; however, the decreases were not monotonic, rather they showed small increases when the consensus threshold value was increased.

There were some exceptions to the qualitative behavior exhibited by most of the scenarios. Figure 3.3 shows examples of some of the irregular results. For both allele calling methods, the estimates under model M_0 did not always show a decline as the number of PCR replicates increased (see top-right graph of Figure 3.3). Under model M_0 and the two point mixture model, the variances of the estimates showed no pattern with increasing number of PCR replicates, resulting in irregular RMSEs with respect to the number of PCR replicates (Figure 3.3). Under models M_h jackknife and M_h Chao, the RMSEs of the probabilistic method did not always exhibit a monotonic decrease. Instead, the RMSE occasionally increased slightly when the number of PCR replicates was increased by one (see bottom-left graph of Figure D.26 for an example).

Table 3.1: Mean population size estimates and standard deviations with no genotyping errors.

simulation	population	capture	sampling	presence of	M_0		mixture		jackknife		Chao	
number	size	probability	occasions	null alleles	est	sd	est	sd	est	sd	est	sd
true	small				200		200		200		200	
1	small	small	small	no	191.6	14.1	214.5	52.1	195.8	10.4	210.2	22.8
2	small	small	small	yes	192.3	14.3	210.0	44.5	195.4	9.9	209.8	23.1
3	small	small	large	no	194.1	6.9	200.5	12.7	222.4	8.9	199.3	9.3
4	small	small	large	yes	195.0	5.8	205.1	19.8	224.4	7.7	201.5	8.8
5	small	large	small	no	190.4	3.9	198.2	9.0	215.2	5.7	198.4	5.3
6	small	large	small	yes	191.5	4.1	197.5	7.2	216.1	6.3	198.8	6.0
7	small	large	large	no	197.1	1.6	198.5	1.9	205.7	3.0	199.8	2.2
8	small	large	large	yes	196.9	1.4	198.2	1.8	205.2	3.0	199.3	2.1
true	large				500		500		500		500	
9	large	small	small	no	354.3	178.2	512.2	58.8	490.6	17.5	528.3	34.5
10	large	small	small	yes	305.4	205.9	525.8	95.6	492.3	21.1	532.5	38.8
11	large	small	large	no	436.1	6.5	503.7	27.8	557.3	12.0	500.2	14.0
12	large	small	large	yes	437.3	7.7	509.0	33.1	559.5	13.8	502.3	16.0
13	large	large	small	no	464.3	5.6	494.3	10.4	539.5	7.8	497.0	7.5
14	large	large	small	yes	466.7	5.7	499.3	13.7	543.6	10.1	500.7	10.2
15	large	large	large	no	493.5	2.8	496.5	2.9	514.0	5.3	499.2	3.8
16	large	large	large	yes	493.6	2.4	496.5	2.5	513.6	5.1	499.2	3.7

Table 3.2: Number of PCR replications required to achieve a mean population size estimate within 20% of the true population size for various simulation scenarios. “trad” refers to the traditional method. “prob” refers to the probabilistic method.

			number of		number of PCR replications required							
simulation	population	capture	sampling	null alleles	M_0		mixture		jackknife		Chao	
number	size	probability	occasions	present	trad	prob	trad	prob	trad	prob	trad	prob
1	small	small	small	no	3	3	X	5*	2	2	8*	5
2	small	small	small	yes	3*	3	8*	6	2	2	8*	5
3	small	small	large	no	1	1	X	5	X	X	X	5
4	small	small	large	yes	1	1	X	10	X	X	8*	5
5	small	large	small	no	5	3	X	6	X	X	8*	5
6	small	large	small	yes	5*	4	X	6	X	X	X	6
7	small	large	large	no	8*	5	X	7	X	10	X	8
8	small	large	large	yes	X	5	X	8	X	X	X	9
9	large	small	small	no	X	X	8*	5	2	2	X	5
10	large	small	small	yes	X	X	X	9	2	2	X	9
11	large	small	large	no	1	1	X	6	X	X	5*	4
12	large	small	large	yes	2	1	X	7	X	X	8*	5
13	large	large	small	no	3	3	8*	5	X	8	8*	4
14	large	large	small	yes	3	3	X	8	X	X	X	7
15	large	large	large	no	8	5	X	6	X	9	X	7
16	large	large	large	yes	X	6	X	9	X	X	X	X

X indicates that the estimate was never within 20% of the true population size.

* indicates that the estimate reached below 20% of the true population size and then increased to above 20%.

Table 3.3: Mean population size estimates ("est") and standard deviations ("sd") at 5 PCR replicates for two allele calling methods.

	M_0				mixture				jackknife				Chao			
sim	traditional		probabilistic		traditional		probabilistic		traditional		probabilistic		traditional		probabilistic	
#	est	sd	est	sd	est	sd	est	sd	est	sd	est	sd	est	sd	est	sd
true	200		200		200		200		200		200		200		200	
1	216	31.4	213	25.0	242	61.5	240	62.8	210	17.0	209	14.3	243	44.8	238	36.5
2	219	31.4	214	28.3	246	66.3	241	68.1	211	18.0	208	17.0	246	45.9	240	40.2
3	212	53.4	212	42.0	248	42.5	238	40.1	267	38.6	256	27.9	251	51.8	236	34.7
4	207	61.6	210	48.3	255	51.3	257	77.9	267	49.8	260	45.4	243	37.0	236	32.3
5	224	32.7	218	27.2	255	67.1	248	65.3	262	43.6	254	36.5	249	50.9	238	40.8
6	232	40.5	222	31.6	265	70.3	244	46.2	272	50.6	259	39.6	259	59.9	243	43.8
7	248	38.1	233	24.6	284	74.9	253	41.1	291	64.7	266	39.7	293	77.9	261	43.2
8	251	42.0	236	40.2	284	70.8	260	50.7	296	68.2	274	49.7	295	76.6	268	50.5
true	500		500		500		500		500		500		500		500	
9	324	198.1	328	195.3	606	126.2	587	124.8	529	35.3	521	27.4	618	90.0	596	67.1
10	323	197.4	309	203.6	638	169.2	617	133.5	536	43.5	530	41.7	638	116.4	624	105.7
11	477	35.3	470	26.9	643	177.6	610	114.1	643	74.2	628	55.8	599	92.3	579	66.6
12	487	38.9	476	29.3	670	160.8	627	129.1	662	79.0	640	58.7	619	95.5	590	68.3
13	523	59.2	512	49.8	613	118.7	585	77.4	647	107.6	626	89.8	602	99.8	577	62.4
14	541	64.3	529	52.2	648	147.4	632	138.2	677	115.8	655	91.7	642	133.9	614	98.7
15	601	81.2	583	64.0	657	133.4	620	89.8	695	134.1	663	100.2	685	149.7	646	101.9
16	629	106.5	605	78.2	701	173.7	651	111.8	739	171.7	696	119.3	732	186.0	680	119.6

Table 3.4: Mean population size estimates (“est”) and standard deviations (“sd”) at 10 PCR replicates for two allele calling methods.

	M_0				mixture				jackknife				Chao			
sim	traditional		probabilistic		traditional		probabilistic		traditional		probabilistic		traditional		probabilistic	
#	est	sd	est	sd	est	sd	est	sd	est	sd	est	sd	est	sd	est	sd
true	200		200		200		200		200		200		200		200	
1	214	28.3	204	20.5	248	70.6	233	64.5	209	16.2	204	13.1	241	43.0	226	29.1
2	218	30.2	207	26.5	243	57.8	227	59.6	210	16.5	204	15.7	244	42.9	229	38.5
3	193	74.2	209	27.4	252	47.3	228	39.6	266	40.4	244	22.9	250	50.7	223	27.2
4	211	58.8	212	20.8	263	67.7	235	55.4	270	50.7	250	41.9	247	41.0	227	41.6
5	221	29.4	208	21.1	252	56.9	229	48.0	259	39.8	241	29.1	245	46.6	223	29.7
6	231	41.2	213	28.7	264	76.8	227	36.8	271	55.0	246	36.2	256	59.6	228	37.7
7	251	42.3	218	18.7	284	79.7	225	24.2	295	70.7	238	28.9	297	85.0	230	27.6
8	250	38.3	223	29.1	282	66.4	232	36.7	295	62.9	247	42.8	294	71.6	238	39.2
true	500		500		500		500		500		500		500		500	
9	288	212.6	332	192.6	604	131.4	556	87.4	529	36.9	510	25.9	618	95.9	572	60.3
10	322	197.2	314	201.5	634	172.2	591	114.7	534	40.3	521	38.2	632	97.8	599	92.1
11	476	36.5	458	23.7	630	150.2	568	89.6	640	76.3	603	48.7	594	94.7	549	54.8
12	484	40.5	465	27.0	658	164.7	581	84.6	657	81.4	616	54.1	614	97.5	563	60.0
13	526	62.0	495	50.7	620	130.1	539	50.6	651	112.5	593	91.2	606	103.9	541	50.7
14	540	60.7	509	45.7	659	155.8	579	89.0	676	109.4	619	80.2	642	129.8	577	83.2
15	594	82.3	545	48.9	644	129.8	560	60.0	682	133.8	595	72.2	670	143.6	571	65.9
16	625	106.6	568	66.7	691	161.5	592	87.1	734	175.0	631	99.5	721	173.7	608	93.6

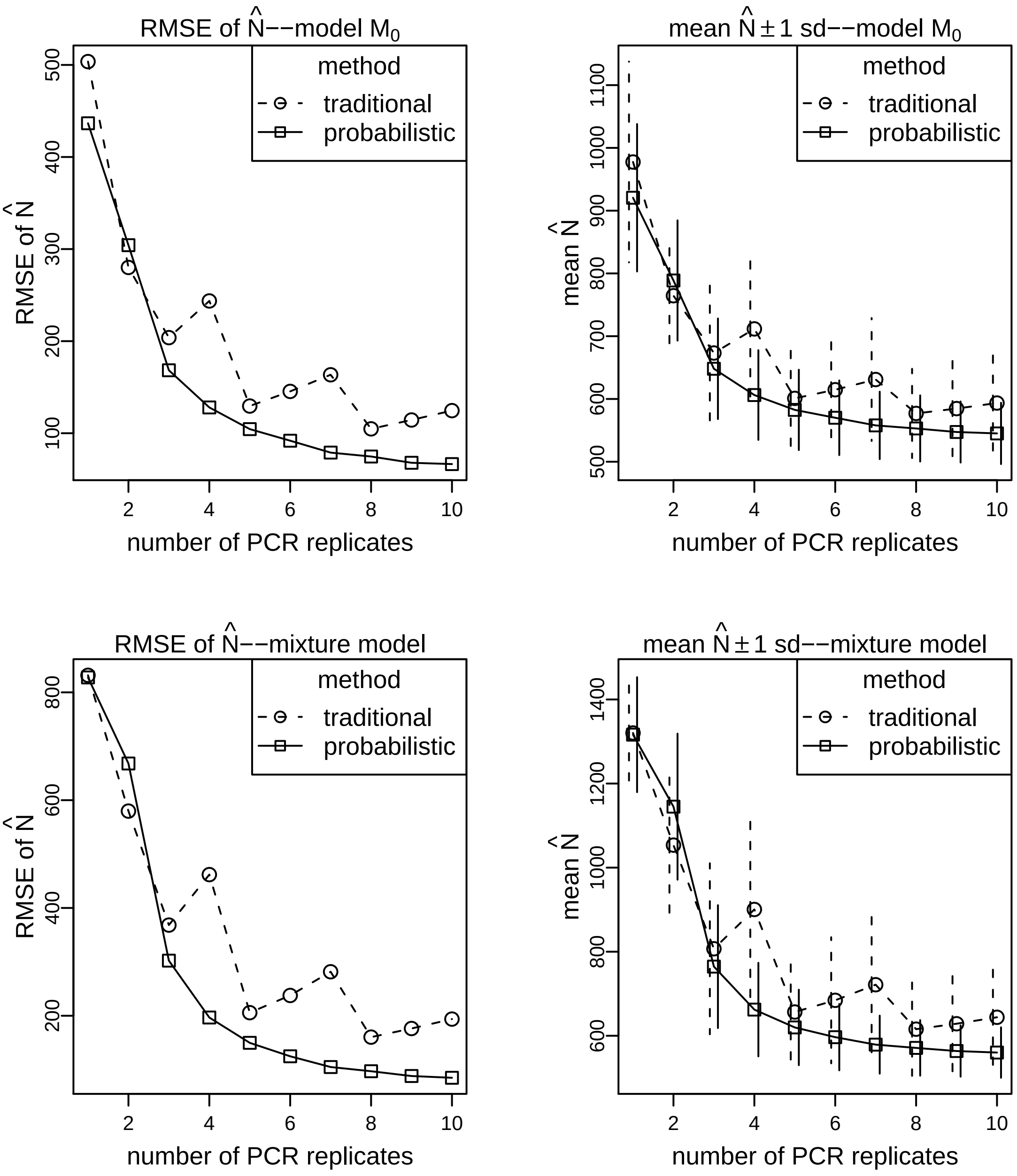


Figure 3.1: Results for simulation 15: N=500, capture probability=.5, sampling occasions=10, no null alleles. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_0 and the two point mixture model for two methods—the traditional consensus method ("traditional") and the probabilistic method ("probabilistic").

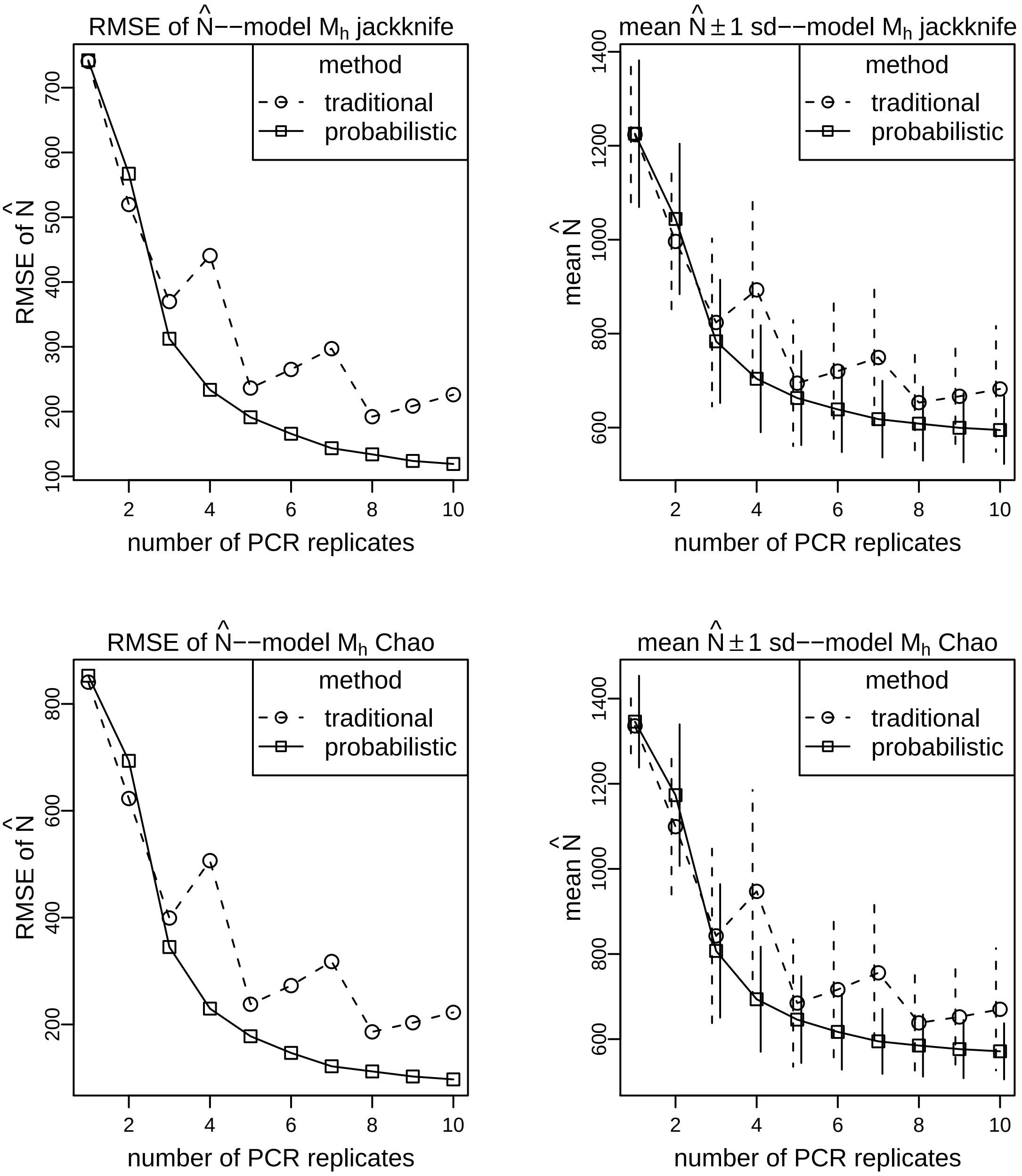


Figure 3.2: Results for simulation 15: N=500, capture probability=.5, sampling occasions=10, no null alleles. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_h jackknife and model M_h Chao for two methods—the traditional consensus method ("traditional") and the probabilistic method ("probabilistic").

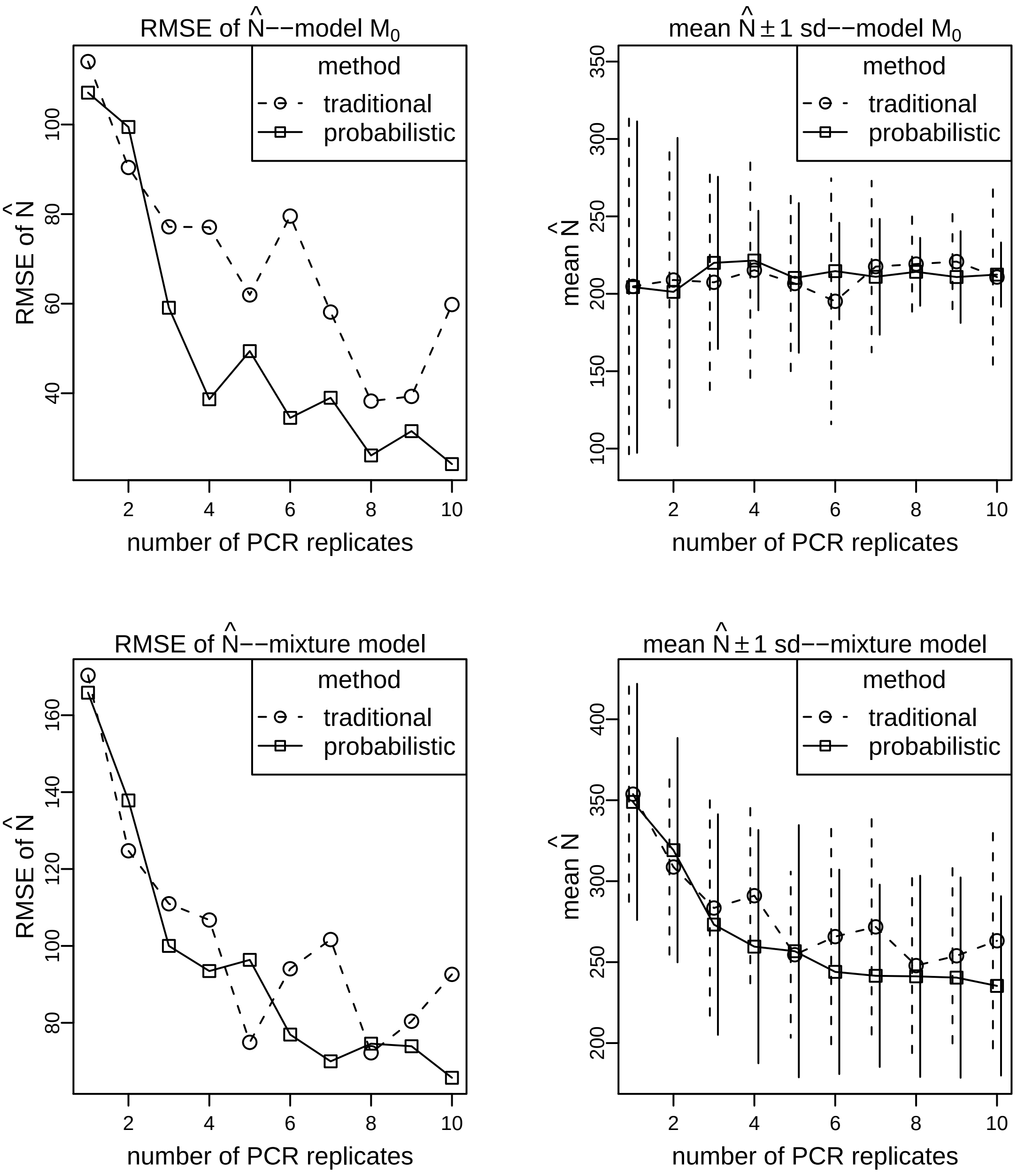


Figure 3.3: Results for simulation 4: N=200, capture probability=.2, sampling occasions=10, null alleles present. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_0 and the two point mixture model for two methods—the traditional consensus method ("traditional") and the probabilistic method ("probabilistic").

3.2 Performance of Mark–Recapture Models

3.2.1 Model M_0

Under model M_0, the probabilistic method performed better than the traditional method for almost all scenarios when more than two PCR replicates were examined (see top graphs of Figure 3.1 for an example). For these scenarios, the probabilistic method almost always had lower biases, variances, and RMSEs than the traditional method. A maximum likelihood estimate (MLE) of population size was obtained for almost all simulations for these scenarios.

For some of the simulations for scenarios 3 and 4 (small population sizes, small mean capture probabilities, and large numbers of sampling occasions), an MLE was not obtained under model M_0. This most often occurred at one or two PCR replicates and tended to occur less often with increasing number of PCR replicates for the probabilistic method. When more than two PCR replicates were examined, no estimate was obtained more often for the traditional method (mean=7.1% of attempted estimates, standard deviation=3.4%) than the probabilistic method (mean=2.1% of attempted estimates, standard deviation=2.1%). When genotyping errors were absent, an MLE was always obtained for these scenarios.

Model M_0 performed extremely poorly for simulation scenarios 9 and 10 (large population sizes, small mean capture probabilities, and small numbers of sampling occasions). The probabilistic method did not consistently perform better than the traditional method (see top graphs of Figures D.17 and D.19). For these scenarios, biases and variances were large for all estimates, including the estimates with no genotyping error. The biases did not show much difference between the two methods and did not decline with increasing number of PCR replicates. The variances did not change systematically with respect to the number of PCR replicates. This resulted in RMSEs that varied erratically as the number of PCR replicates increased. An MLE was not always obtained for all simulations for these scenarios. The probabilistic and the traditional method showed similar rates of not obtaining MLEs (traditional: mean=28.9% of attempted estimates, standard deviation=3.3%; probabilistic: mean=29.1% of attempted estimates, standard deviation=4.1%). When genotyping errors were absent, no MLE was obtained in 20% of simulations in scenario 9 and 31% of simulations in scenario 10.

3.2.2 Two Point Mixture Model

Under the two point mixture model, when more than two PCR replicates were examined, the probabilistic method performed better than the traditional method in almost all situations (see bottom graphs of Figure 3.1 for an example). The biases of the probabilistic method were most often less than, but occasionally equal to, the biases of the traditional method when more than two PCR replicates were examined. The biases tended to decrease monotonically with increasing numbers of PCR replicates for the probabilistic method and the biases showed an overall decrease for the traditional method; however, the decrease was not monotonic. The variances were usually lower for the probabilistic method than for the traditional method. The variances tended to decrease with increasing numbers of PCR replicates. The RMSEs of the probabilistic method tended to decline with increasing numbers of PCR replicates, but the decrease was not always monotonic. The RMSEs of the probabilistic method were lower than the RMSEs of the traditional method, with the exception of a couple situations. In scenarios 1 and 10, there were a couple of data points at different numbers of PCR replicates where the RMSE of the traditional method was lower than the RMSE of the probabilistic method (see bottom graphs of Figures D.1 and D.19).

A maximum likelihood estimate (MLE) of population size was obtained for all simulations; however, all scenarios had some estimates that were unreasonably large. The number of unreasonably large estimates tended to decline with the number of PCR replicates. Estimates were frequently unreasonably large at one or two PCR replicates (mean=23.8% of estimates, standard deviation=21.8%). When more than two PCR replicates were examined, unreasonably large estimates were obtained more often for the traditional method (mean=2.3% of estimates, standard deviation=2.5%) than the probabilistic method (mean= 1.2% of estimates, standard deviation=1.9%). Simulations 1 and 2 (small population sizes, small mean capture probabilities, and small numbers of sampling occasions) had the highest rates of estimates that were unreasonably large. Scenarios 1 and 2 were the only two scenarios to have any estimates that were unreasonably large when genotyping errors were absent (scenario 1: 1% of estimates, scenario 2: 2% of estimates).

3.2.3 Model M_h Jackknife

Under model M_h jackknife, when more than two PCR replicates were examined, the probabilistic method always performed better than the traditional method. It exhibited lower biases, variances, and RMSEs (see top graphs of Figure 3.2 for an example). All scenarios showed the same qualitative behavior: biases and RMSEs decreased with increasing numbers of PCR replicates. The decreases were monotonic for the probabilistic method and not monotonic for the traditional method. This model occasionally produced estimates that were unreasonably large. This most often occurred at one PCR replicate for simulations 7,8,15, and 16 (large mean capture probabilities and large numbers of sampling occasions). The probabilistic method had less unreasonably large estimates than the traditional method. Using the probabilistic method, no scenario had any unreasonably large estimates at more than two PCR replicates. The traditional method rarely had unreasonably large estimates at more than two PCR replicates. For both methods, a reasonable estimate was always obtained when genotyping errors were absent.

3.2.4 Model M_h Chao

Under model M_h Chao, when more than two PCR replicates were examined, the probabilistic method always performed better than the traditional method. It exhibited lower biases, variances, and RMSEs (see bottom graphs of Figure 3.2 for an example). Most scenarios showed the same qualitative behavior: biases and RMSEs decreased with increasing numbers of PCR replicates. The decreases were monotonic for the probabilistic method and not monotonic for the traditional method. In a few scenarios, the probabilistic method showed an occasional increase in RMSE when the number of PCR replicates increased by one (see bottom-left graph of Figure D.26 for an example). This model occasionally produced estimates that were unreasonably large. This occurred more often under model M_h Chao than it did under model M_h jackknife. Unreasonably large estimates usually occurred at one or two PCR replicates. The probabilistic method had less unreasonably large estimates than the traditional method. For both methods, a reasonable estimate was always obtained when genotyping errors were absent.

3.3 Comparisons of Estimators for Varying Parameters

3.3.1 Population Size

To examine the effect of population size, I examined changes in relative mean biases, relative variances, and relative MSEs of the population size estimates for pairs of simulations that had identical input values, except for whether the population size was 200 or 500. For all four models, there was no clear trend in response to changes in population size. The changes in relative biases and relative MSEs due to changes in population size depended on other simulation parameters. The relative variances tended to decrease in response to increasing population size in most situations, but this also depended on other parameters. The probabilistic method performed better than the traditional method, regardless of the population size.

3.3.2 Capture Probabilities

To examine the effect of capture probabilities, I examined changes in mean biases, variances, and RMSEs of the population size estimates for pairs of simulations that had identical input values, except for whether the mean capture probability was 0.2 (variance=0.002) or 0.5 (variance=0.020). Biases, variances, and RMSEs most often increased with increasing capture probabilities. Overall, the increase in capture probabilities tended to result in poorer estimates, especially at low numbers of PCR replicates. While increasing capture probabilities resulted in poorer estimates for both the traditional and probabilistic method, the effect was greater on the traditional method. The probabilistic method performed better than the traditional method, regardless of the mean capture probability.

Under model M_0, there were no clear trends in changes in biases, variances, nor RMSEs for changes in capture probabilities. Under the two point mixture model, biases and RMSEs tended to either increase or stay the same with increasing capture probabilities. The magnitude of these differences in RMSEs tended to be larger for the traditional method and at low numbers of PCR replicates. The variances showed no clear trend in response to changes in capture probabilities under model M_0.

The biases, variances, and RMSEs tended to increase with increasing capture probabilities under model M_h jackknife (Figure 3.4). The magnitude of these differences tended to be larger for the traditional method, as compared to the probabilistic method.

The magnitude of these differences in biases and RMSEs tended to be larger for low numbers of PCR replicates. The magnitude of these differences in variances tended to be slightly larger for low numbers of PCR replicates.

Under model M_h Chao, biases and RMSEs tended to either increase or stay the same with increasing capture probabilities. The magnitude of these differences tended to be larger for the traditional method and for lower numbers of PCR replicates. Under model M_h Chao, the variances tended to increase for increasing capture probabilities when more than one PCR replicate was examined. The magnitude of these differences in variances tended to be larger for the traditional method, as compared to the probabilistic method.

3.3.3 Sampling Occasions

To examine the effect of the number of sampling occasions, I examined changes in mean biases, variances, and RMSEs of the population size estimates for pairs of simulations that had identical input values, except for whether there were 5 or 10 sampling occasions. The effect that changing the number of sampling occasions had on the biases, variances, and RMSEs of population size estimates varied with the model used and other input parameters. Biases and RMSEs most often increased with increasing number of sampling occasions. Overall, an increase in the number of sampling occasions usually resulted in poorer estimates, especially at low numbers of PCR replicates. While increasing the number of sampling occasions resulted in poorer estimates for both the traditional and probabilistic method, the effect was greater for the traditional method than the probabilistic method. The probabilistic method performed better than the traditional method, regardless of the number of sampling occasions.

Under model M_0, the changes in biases, variances, and RMSEs in response to increasing the number of sampling occasions varied between comparisons. For most comparisons, these changes in bias showed an increase in magnitude for the traditional method and as the number of PCR replicates decreased. Under the two point mixture model, biases and RMSEs tended to either increase or stay the same for increasing number of sampling occasions. The magnitude of these differences tended to be larger for the traditional method and at low numbers of PCR replicates. The variances showed no clear trend in response to changes in the number of sampling occasions under the two point mixture model. Under model M_h jackknife, the biases, variances, and RMSEs tended to increase for increasing

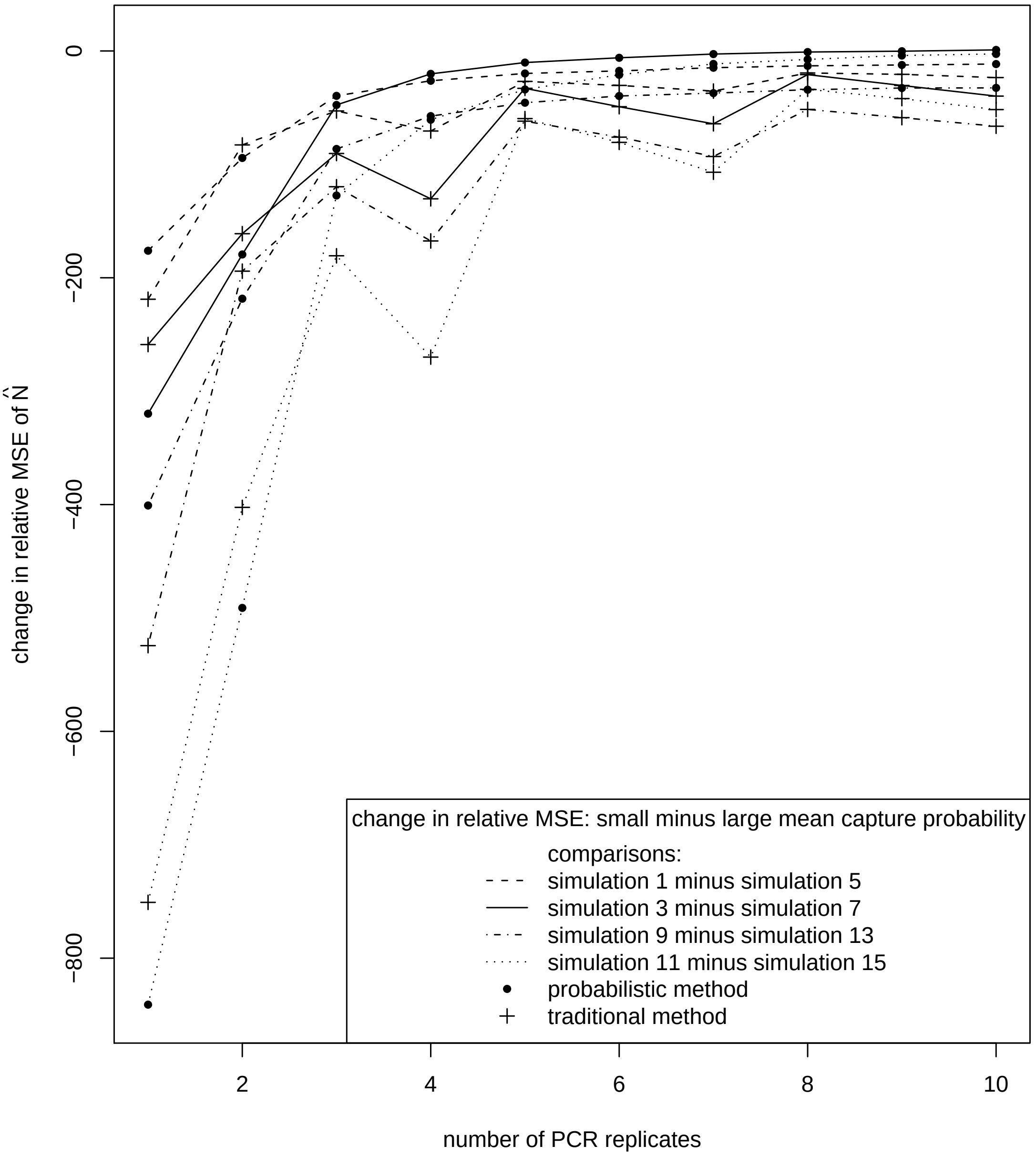


Figure 3.4: Effect of change in mean capture probability on relative MSE of $\hat{N}$ for model M_h jackknife. Differences in relative MSEs for four simulation pairs, where each pair had identical input values, except for the mean and variance of the capture probabilities. Each point was obtained by subtracting the relative MSE of the large mean capture probability scenario from the corresponding relative MSE of the small mean capture probability scenario.

numbers of sampling occasions. The magnitude of these differences tended to be larger for the traditional method and at low numbers of PCR replicates. Under model M_h Chao, the biases and RMSEs tended to either increase or stay the same for increasing numbers of sampling occasions. The magnitude of these differences tended to be larger for the traditional method and at low numbers of PCR replicates. The variances showed no clear trend in response to changes in the number of sampling occasions under model M_h Chao.

3.3.4 Null Alleles

To examine the effect of null alleles, I examined changes in mean biases, variances, and RMSEs of the population size estimates for pairs of simulations that had identical input values, except for whether or not null alleles were present. The probabilistic method performed better than the traditional method, even in the presence of null alleles. The two methods showed very similar changes in response to the presence of null alleles. In many situations, the presence of null alleles caused increases in biases, variances, and RMSEs of estimates, regardless of the model used. These increases in biases and variances occurred less often under model M_h Chao than the other models. There was not much change in RMSEs under model M_0. The magnitudes of the changes in RMSEs under the heterogeneity models tended to be larger for lower numbers of PCR replicates.

Chapter 4

Discussion

4.1 Comparison of Allele Calling Methods

Overall, the probabilistic method performed better than the traditional method. Population size estimates showed less bias and more precision when more than two PCR replicates were examined. Using the probabilistic method, good population size estimates can be achieved using fewer PCR replicates. The probabilistic method resulted in better estimates because uncertainty was allowed to remain in allele calls until all the PCR replicates were examined. This allowed utilization of more information from the electropherograms than is traditionally used.

I did not examine the performance of the two allele calling methods when only one or two PCR replicates were examined. The low quality of the samples in the simulations requires more extensive PCR replication to reduce the rate of misidentification and achieve good population size estimates. While the probabilistic method accounts for genotyping errors due to allelic drop out and stutter, it does not account for contamination. Both allele calling methods require PCR replication to detect contamination. I did note that when only one PCR replication was examined, the probabilistic method tended to do slightly better. This is most likely due to the fact that undesignated alleles are assigned differently for the two allele calling methods. Both methods of assigning undesignated alleles were designed to be used with multiple PCR replications, not for single PCR replications.

The probabilistic allele calling method relies on electropherogram peak heights to assign probabilities; however, electropherogram peak areas are often a better metric for interpreting electropherograms because peak areas include information on peak morphology (Clayton et al. 1998; Gill et al. 1998; Butler 2005). When using the probabilistic method, peak areas can be substituted for peak heights. In addition, relying solely on measures of signal intensity may not be sufficient (Paetkau 2003). Allele calling is very subjective and relies on the judgment and experience of the technician to examine all of the information in the electropherogram. The objective probabilistic method can be used in conjunction with subjective allele calling by allowing the technician to manually alter allele call probabilities based on other information in the electropherogram.

Using the probabilistic method resulted in better estimates than the traditional method; however, the neither method completely eliminated the biases that result from genotyping errors. Simulations with more than 10 PCR replications reveal that as the number of PCR replicates continues to increase, an asymptotic estimate is reached that retains a residual bias (results not shown). Simulations with higher efficiency parameters, and therefore lower error rates, reveal that this residual bias is most likely due to the high genotyping error rates (results not shown). As the efficiency parameters increased, the asymptotic estimate approached the value of the estimate in the absence of genotyping errors. In the presence of genotyping errors, the additional information gained from further PCR replication is balanced by the probability of having genotyping errors in the new PCR replicates.

Residual genotyping errors are also seen with the multiple-tubes approach (Waits 2004; Petit and Valiere 2006), which relies on extensive PCR replication to reduce genotyping errors. Residual genotyping errors, even if small, can still cause biased population size estimates (Yoshizaki 2007). In the presence of high genotyping error rates, I would expect that the residual genotyping error would be large enough to substantially bias population size estimates, even with extensive PCR replication.

The residual genotyping error rates in the simulations are possibly higher than the actual rates because poor samples were not culled and all samples were PCR amplified the same number of times at each locus. In practice, samples that consistently amplify poorly are completely removed from analysis. Removing poor samples increases the overall probability of obtaining correct genotypes. In addition, all samples are not usually PCR amplified the same number of times at each locus. Samples with uncertain genotypes at specific loci are typically selected for additional PCR replications at those loci. This ensures that any additional PCR replications elucidate ambiguous genotypes, rather than introducing uncertainty to samples with unequivocal genotypes.

The residual bias also could have been caused by undue influence of outlier allele call probabilities caused by very poor PCR replicates. To address this possibility, I examined simulations where the final allele call probabilities were determined by using the median, rather than the mean, of the probabilities from the PCR replicates. The results did not change (results not shown), indicating that outlier allele call probabilities are not a likely cause of the residual bias.

4.2 Comparison of Estimators for Various Models and Parameters

In the absence of genotyping errors, model M_0 did not substantially underestimate the population size for many scenarios, even though there was individual variation in capture probabilities, which causes underestimation under model M_0. While there was heterogeneity in capture probabilities, the amount of heterogeneity may not have been high enough to cause substantial biases under model M_0. In addition, the lack of substantial bias may be due to the reduction in variation between individuals that occurs because the analysis is based on whether or not an animal was capture during a sampling occasion, not how many times it was capture during that occasion. Since each animal can be captured multiple times during a given sampling occasion, an animal captured multiple times during that occasion would be recorded the same as an animal that was captured only one time during that occasion.

In the presence of genotyping errors, model M_0 usually resulted in overestimation of population size and the three heterogeneity models always resulted in overestimation. The only time the population size was underestimated was when the population size was considerably underestimated in the absence of genotyping errors. All simulations were run with eight highly variable loci, with genotypes assigned so the populations were in Hardy–Weinberg proportions. This ensured that the loci had the power to distinguish between individuals, preventing the shadow effect as a source of underestimation of population size. On some occasions, model M_0 seemed to perform extremely well in the presence of high levels of genotyping errors. It is unlikely that it is a better estimator in these cases; rather, in the absence of genotyping errors, the model underestimates the population size. The model's inherent underestimation of the population size cancels out the overestimation of population size caused by genotyping errors.

An increase in the number of samples analyzed—due to larger capture probabilities or more sampling occasions—often resulted in poorer population size estimates. This is likely due to additional samples allowing more chances for errors to occur. Each sample analyzed has the potential to produce multiple ghosts, which cause overestimation of population size. The increased potential for ghosts is balanced against the increased information gained by collecting more samples. As the number of PCR replicates increased, the genotyping error rate decreased, and the increase in the number of samples no longer resulted in poorer estimates. The poorer estimates for simulations with high mean capture probabilities could also be due to the higher input capture probability variances used in those scenarios.

Both methods showed similar changes in response to the presence of null alleles. I expected that the probabilistic method would be more negatively affected by null alleles because peak heights are lower when null alleles are present. An individual that is heterozygous for a null allele and a regular allele would consistently appear as homozygous for the regular allele with a lower peak height than expected. This should not affect the traditional allele calling method. The probabilistic method will be affected by the lower peak height because it gives the appearance of possible allelic drop out. The fact that the two methods showed similar responses to null alleles is most likely an artifact of using the assigned probability of a single PCR replicate to determine the definitive allele call for that PCR replicate.

4.3 Future Work

Simulation results show that the probabilistic allele calling method performed better than the traditional method. The simulated traditional method examines the same number of PCR replicates for all samples at each locus. In addition to this traditional method, the probabilistic method should be tested against a traditional method that utilizes sequential stopping rules. Sequential stopping rules are commonly used in practice where, for example, a heterozygote would be confirmed with agreement between two PCR replicates and a homozygote would be confirmed with agreement between seven PCR replicates. Sequential stopping rules utilize limited resources more efficiently and may reduce misidentification rates by targeting ambiguous genotypes. In addition to adding sequential stopping rules, the simulations should be modified to include in the culling of poor samples. Culling of poor samples is an important procedure that helps reduce misidentification rates.

The simulation performance of the probabilistic allele calling method shows that the method may improve population size estimates from non-invasive genetic mark–recapture studies. In order to validate the probabilistic method, it will need to be tested on actual non-invasive genetic mark–recapture data. This data should come from a population of known size or from a study that estimated the population size using a previously validated estimation method.

While the probabilistic allele calling method did reduce overestimation of population size due to genotyping errors, it did not eliminate the overestimation. Continued examination of this method is needed, as well as continued examination of methods to further reduce genotyping errors and the development of models that account for them. The probabilistic allele calling method can be used in conjunction with new laboratory methods and new mark–recapture models. Researchers are continuing to investigate new models specific to non-invasive genetic mark–recapture data. These new models include models that allow multiple captures from a single individual per sampling occasion and continuous sampling models.

Bibliography

Boulanger, J., Stenhouse, G., Munro, R., and Bradley, R. D. (2004). Sources of heterogeneity bias when DNA mark–recapture sampling methods are applied to grizzly bear (Ursus Arctos) populations. *Journal of Mammalogy*, 85(4):618–624.

Buchan, J. C., Archie, E. A., Van Horn, R. C., Moss, C. J., and Alberts, S. C. (2005). Locus effects and sources of error in noninvasive genotyping. *Molecular Ecology Notes*, 5(3):680–683.

Buckleton, J. and Triggs, C. (2006). Is the 2p rule always conservative? *Forensic Science International*, 159(2–3):206–209.

Burnham, K. P. and Overton, W. S. (1978). Estimation of the size of a closed population when capture probabilities vary among animals. *Biometrika*, 65(3):625–633.

Butler, J. M. (2005). *Forensic DNA Typing : Biology, Technology, and Genetics of STR Markers*. Burlington, MA : Elsevier Academic Press, Amsterdam.

Chao, A. (1989). Estimating population size for sparse data in capture–recapture experiments. *Biometrics*, 45(2):427–438.

Chao, A. (2006). Species estimation and applications. In Balakrishnan, N., Read, C. B., and Vidakovic, B., editors, *Encyclopedia of Statistical Sciences*, number 12, pages 7907–7916. Wiley, New York, second edition.

Clayton, T. M., Whitaker, J. P., Sparkes, R., and Gill, P. (1998). Analysis and interpretation of mixed forensic stains using DNA STR profiling. *Forensic Science International*, 91(1):55–70.

Creel, S., Spong, G., Sands, J. L., Rotella, J., Zeigle, J., Joe, L., Murphy, K. M., and Smith, D. (2003). Population size estimation in Yellowstone wolves with error–prone noninvasive microsatellite genotypes. *Molecular Ecology*, 12(7):2003–2009.

Eggert, L. S., Eggert, J. A., and Woodruff, D. S. (2003). Estimating population sizes for elusive animals: the forest elephants of Kakum National Park, Ghana. *Molecular Ecology*, 12(6):1389–1402.

Evett, I. W., Gill, P. D., and Lambert, J. A. (1998). Taking account of peak areas when interpreting mixed DNA profiles. *Journal of Forensic Sciences*, 43(1):62–69.

Ewen, K. R., Bahlo, M., Treloar, S. A., Levinson, D. F., Mowry, B., Barlow, J. W., and Foote, S. J. (2000). Identification and analysis of error types in high-throughput genotyping. *The American Journal of Human Genetics*, 67(3):727–736.

Frantz, A. C., Pope, L. C., Carpenter, P. J., Roper, T. J., Wilson, G. J., Delahay, R. J., and Burke, T. (2003). Reliable microsatellite genotyping of the Eurasian bager (*Meles meles*) using faecal DNA. *Molecular Ecology*, 12(6):1649–1661.

Gagneux, P., Boesch, C., and Woodruff, D. S. (1997). Microsatellite scoring errors associated with noninvasive genotyping based on nuclear DNA amplified from shed hair. *Molecular Ecology*, 6(9):861–868.

Gill, P. (2001). Application of low copy number DNA profiling. *Croatian Medical Journal*, 42(3):229–232.

Gill, P., Brenner, C., Buckleton, J., Carracedo, A., Krawczak, M., Mayr, W., Morling, N., Prinz, M., Schneider, P., and Weir, B. (2006). DNA commission of the International Society of Forensic Genetics: Recommendations on the interpretation of mixtures. *Forensic Science International*, 160(2–3):90–101.

Gill, P., Curran, J., and Elliot, K. (2005). A graphical simulation model of the entire DNA process associated with the analysis of short tandem repeat loci. *Nucleic Acids Research*, 33(2):632–643.

Gill, P., Sparkes, R., and Kimpton, C. (1997). Development of guidelines to designate alleles using an STR multiplex system. *Forensic Science International*, 89(3):185–197.

Gill, P., Sparkes, R., Pinchin, R., Clayton, T., Whitaker, J., and Buckleton, J. (1998). Interpreting simple STR mixtures using allele peak areas. *Forensic Science International*, 91(1):41–53.

Gill, P., Whitaker, J., Flaxman, C., Brown, N., and Buckleton, J. (2000). An investigation of the rigor of interpretation rules for STRs derived from less than 100 pg of DNA. *Forensic Science International*, 112(1):17–40.

Kendall, W. L., Nichols, J. D., and Hines, J. E. (1997). Estimating temporary emigration using capture–recapture data with Pollock's robust design. *Ecology*, 78(2):563–578.

Ladd, C., Lee, H., Yang, N., and Bieber, F. (2001). Interpretation of complex forensic DNA mixtures. *Croatian Medical Journal*, 42(3):244–246.

Lai, Y. and Sun, F. (2004). Sampling distribution for microsatellites amplified by PCR: mean field approximation and its applications to genotyping. *Journal of Theoretical Biology*, 228(2):185–194.

Langer, T. J. (2006). *Population estimates with age and genetic structure of a harvested bear population in eastern North Carolina.* PhD thesis, North Carolina State University, Raleigh, North Carolina, USA.

Link, W. A. (2003). Nonidentifiability of population size from capture–recapture data with heterogeneous detection probabilities. *Biometrics*, 59(4):1123–1130.

Lukacs, P. M. and Burnham, K. P. (2005). Research notes: Estimating population size from DNA-based closed capture–recapture data incorporating genotyping error. *Journal of Wildlife Management*, 69(1):396–403.

Miller, C. R., Joyce, P., and Waits, L. P. (2002). Assessing allelic dropout and genotype reliability using maximum likelihood. *Genetics*, 160(1):357–366.

Mills, L. S., Citta, J. J., Lair, K. P., Schwartz, M. K., and Tallmon, D. A. (2000). Estimating animal abundance using noninvasive DNA sampling: Promise and pitfalls. *Ecological Applications*, 10(1):283–294.

Morin, P. A., Chambers, K. E., Boesch, C., and Vigilant, L. (2001). Quantitative polymerase chain reaction analysis of DNA from noninvasive samples for accurate microsatellite genotyping of wild chimpanzees (Pan troglodytes verus). *Molecular Ecology*, 10(7):1835–1844.

Navidi, W., Arnheim, N., and Waterman, M. S. (1992). A multiple–tubes approach for accurate genotyping of very small DNA samples by using PCR: statistical considerations. *American Journal of Human Genetics*, 50(2):347–359.

Norris, James L., I. and Pollock, K. H. (1996). Nonparametric MLE under two closed capture–recapture models with heterogeneity. *Biometrics*, 52(2):639–649.

Otis, D. L., Burnham, K. P., White, G. C., and Anderson, D. R. (1978). Statistical inference from capture data on closed animal populations. *Wildlife Monographs*, 62:3–135.

Paetkau, D. (2003). An empirical exploration of data quality in DNA-based population inventories. *Molecular Ecology*, 12(6):1375–1387.

Paetkau, D., Waits, L. P., Clarkson, P. L., Craighead, L., Vyse, E., Ward, R., and Strobeck, C. (1998). Variation in genetic diversity across the range of North American brown bears. *Conservation Biology*, 12(2):418–429.

Petit, E. and Valiere, N. (2006). Estimating population size with noninvasive capture–mark–recapture data. *Conservation Biology*, 20(4):1062–1073.

Pierce, B. A. (2003). *Genetics: A Conceptual Approach.* W.H. Freeman and Company, New York, NY.

Pledger, S. (2000). Unified maximum likelihood estimates for closed capture–recapture models using mixtures. *Biometrics*, 56(2):434–442.

Pollock, K. H., Nichols, J. D., Brownie, C., and Hines, J. E. (1990). Statistical inference for capture–recapture experiments. *Wildlife Monographs*, 107:3–97.

R Development Core Team (2005). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria.

Ruell, E. W. and Crooks, K. R. (2007). Evaluation of noninvasive genetic sampling methods for felid and canid populations. *Journal of Wildlife Management*, 71(5):1690–1694.

Shinde, D., Lai, Y., Sun, F., and Arnheim, N. (2003). Taq DNA polymerase slippage mutation rates measured by PCR and quasi-likelihood analysis: $(\mathrm{CA/GT})_n$ and $(\mathrm{A/T})_n$ microsatellites. *Nucleic Acids Research*, 31(3):974–980.

Smith, K., Alberts, S., Bayes, M., Bruford, M., Altmann, J., and Ober, C. (2000). Cross-species amplification, non-invasive genotyping, and non-mendelian inheritance of human STRPs in Savannah baboons. *American Journal of Primatology*, 51(4):219–227.

Taberlet, P., Griffin, S., Goossens, B., Questiau, S., Manceau, V., Escaravage, N., Waits, L., and Bouvet, J. (1996). Reliable genotyping of samples with very low DNA quantities using PCR. *Nucleic Acids Research*, 24(16):3189–3194.

Taberlet, P. and Luikart, G. (1999). Non-invasive genetic sampling and individual identification. *Biological Journal of the Linnean Society*, 68(1–2):41–55.

Taberlet, P., Waits, L., and Luikart, G. (1999). Noninvasive genetic sampling: look before you leap. *Trends in Ecology & Evolution*, 14(8):323–327.

Tamaki, K. (2007). Minisatellite and microsatellite DNA typing analysis. In Rapley, R. and Whitehouse, D., editors, *Molecular Forensics*, pages 71–89. John Wiley & Sons, Ltd., West Sussex, England, first edition.

Tully, G. (1993). Analysis of 6 VNTR loci by multiplex PCR and automated fluorescent detection. *Human Genetics*, 92(6):554–562.

Valiere, N., Berthier, P., Mouchiroud, D., and Pontier, D. (2002). GEMINI: software for testing the effects of genotyping errors and multitubes approach for individual identification. *Molecular Ecology Notes*, 2(1):83–86.

Waits, J. L. and Leberg, P. L. (2000). Biases associated with population estimation using molecular tagging. *Animal Conservation*, 3(3):191–199.

Waits, L. P. (2004). Using noninvasive genetic sampling to detect and estimate abundance of rare wildlife species. In Thompson, W. L., editor, *Sampling Rare or Elusive Species*, pages 211–228. Island Press, Washington, DC, first edition.

Waits, L. P. and Paetkau, D. (2005). Noninvasive genetic sampling tools for wildlife biologists: A review of applications and recommendations for accurate data collection. *Journal of Wildlife Management*, 69(4):1419–1433.

Wasser, S. K. (1997). Techniques for application of faecal DNA methods to field studies of Ursids. *Molecular Ecology*, 6:1091–1097.

Wasser, S. K., Davenport, B., Ramage, E. R., Hunt, K. E., Parker, M., Clarke, C., and Stenhouse, G. (2004). Scat detection dogs in wildlife research and management: application to grizzly and black bears in the Yellowhead Ecosystem, Alberta, Canada. *Canadian Journal of Zoology*, 82(3):475–492.

Whitaker, J. P., Cotton, E. A., and Gill, P. (2001). A comparison of the characteristics of profiles produced with the AMPFlSTR SGM $Plus_{(TM)}$ multiplex system for both standard and low copy number (LCN) STR DNA analysis. *Forensic Science International*, 123(2–3):215–223.

Williams, B. K., Nichols, J. D., and Conroy, M. J. (2002). *Analysis and Management of Animal Populations*. Academic Press, San Diego, CA.

Yoshizaki, J. (2007). *Use of natural tags in closed population capture–recapture studies: modeling misidentification*. PhD thesis, North Carolina State University, Raleigh, North Carolina, USA.

Appendices

Appendix A

Simulation Code

```
#function to run full simulation
sims=function(param_file, allele_file)
{
  #read parameter file
  params=readin_params(param_file)
  #read allele file
  alleles=read.table(allele_file)
  rm(param_file,allele_file)
  #designate scenario
  simulation=params["simid"]
  #set seed
  seed=as.numeric(params["seed"])
  set.seed(seed)

  loci=length(alleles[,1])
  num_alleles=.5*length(alleles[1,])

  #record params values as proper structure
  num_sims=as.numeric(params["num_sims"])
  num_models=as.numeric(params["num_models"])
    PCR_reps=as.numeric(params["PCR_reps"])
    occasions=as.numeric(params["occasions"])

    num_samples=vector(length=num_sims)
  estimates=array(0,dim=c(num_models,PCR_reps+1,num_sims))
  likely_estimates=array(0,dim=c(num_models,PCR_reps+1,num_sims))
  trad_estimates=array(0,dim=c(num_models,PCR_reps+1,num_sims))
  results=array(0,dim=c((num_models+1),(PCR_reps+1),6))
  quants=array(0,dim=c(num_models,PCR_reps+1,9))
```

```
for (i in 1:num_sims)
 {
true_genos=gen_pop(params,alleles)
capt_probs=gen_capt_probs(params)
true_capt_hx=gen_sample(params, capt_probs)
samples=sum(true_capt_hx)
num_samples[i]=samples
samp_occ=vector(length=samples)#tracks which sample from which occassion
samp=1

for (a in 1:(length(true_capt_hx[,1])))
 {
   for (b in 1:(length(true_capt_hx[1,])))
    {
      x=(true_capt_hx[a,b])
      while (x>0)
       {
      samp_occ[samp]=b
      x=x-1
      samp=samp+1
       }
    }
 }

rm(capt_probs,a,b,x)
genotypes=genotype(params,alleles,true_genos, true_capt_hx, num_alleles,
 loci,samples)
rm(true_genos)
geno_rep_probs=assign_probs(params,genotypes,num_alleles,loci,samples,
 PCR_reps)
rm(genotypes)
av_geno_probs=average_geno_probs(alleles,num_alleles,loci,geno_rep_probs,
      samples,PCR_reps)
likely_genos=most_likely_genos(num_alleles,loci,av_geno_probs,samples,
 PCR_reps)
rm(av_geno_probs)
likely_capture_hx=gen_capt_hx(num_alleles,loci,likely_genos,samples,
 samp_occ,PCR_reps,occasions)
likely_freqs=calc_freqs(params,likely_capture_hx,true_capt_hx)
rm(likely_genos,likely_capture_hx)

print("likely")
```

```
  likely_estimates[1,,i]=jackknife(params,likely_freqs)
  likely_estimates[2,,i]=chao(params,likely_freqs)
  likely_estimates[3,,i]=m0(params,likely_freqs)
  likely_estimates[4,,i]=twopt(params,likely_freqs)

  trad_genos=consensus(loci, num_alleles, geno_rep_probs,samples,PCR_reps,
   alleles)
  rm(geno_rep_probs)
  trad_capture_hx=gen_capt_hx(num_alleles,loci,trad_genos,samples,samp_occ,
   PCR_reps,occasions)
  trad_freqs=calc_freqs(params,trad_capture_hx,true_capt_hx)
  rm(trad_genos,trad_capture_hx)

  print("trad")
  trad_estimates[1,,i]=jackknife(params,trad_freqs)
  trad_estimates[2,,i]=chao(params,trad_freqs)
  trad_estimates[3,,i]=m0(params,trad_freqs)
  trad_estimates[4,,i]=twopt(params,trad_freqs)
  rm(likely_freqs,trad_freqs)
}

  print("average number of samples")
  print(mean(num_samples))
  print(sd(num_samples))
  rm(true_capt_hx,params,alleles,loci,samp,samp_occ,num_alleles,samples,i,
   occasions,num_sims,num_samples)

  for (m in 1:num_models)
   {
     for (r in 1:(PCR_reps+1))
      {
       results[m,r,1]=mean(estimates[m,r,],na.rm=TRUE)
       results[m,r,2]=sd(estimates[m,r,],na.rm=TRUE)
       quants[m,r,1:3]=quantile(estimates[m,r,],p=c(.025,.5,.975),
        na.rm=TRUE)
       results[m,r,3]=mean(likely_estimates[m,r,],na.rm=TRUE)
       results[m,r,4]=sd(likely_estimates[m,r,],na.rm=TRUE)
        quants[m,r,4:6]=quantile(likely_estimates[m,r,],
         p=c(.025,.5,.975),na.rm=TRUE)
       results[m,r,5]=mean(trad_estimates[m,r,],na.rm=TRUE)
       results[m,r,6]=sd(trad_estimates[m,r,],na.rm=TRUE)
       quants[m,r,7:9]=quantile(trad_estimates[m,r,],
        p=c(.025,.5,.975),na.rm=TRUE)
      }
   }

  #keep track of median of two pt for likely and trad models
  results[(m+1),,]=quants[4,,4:9]
  rm(estimates, likely_estimates, trad_estimates,PCR_reps,r)
  print("results")
  print(results[1,,])
  print(results[2,,])
  print(results[3,,])
  print(results[4,,])
  print(results[5,,])

  for (m in 1:(num_models+1))
   {
  filename=paste(simulation,"output",m,".txt",sep="")
  write.table(results[m,,],filename)
}
  rm(simulation,quants,num_models,filename,m)
  return(results)
}

##############################################################
#function to read parameter file
readin_params=function(param_file)
{
  params=scan(param_file, what=list("",""), comment.char="#")
  rm(param_file)
  names(params[[2]])=params[[1]]
  params[[2]]
}

##############################################################
#function to generate true population genotypes
gen_pop=function(params,alleles)
{
  N=as.numeric(params["N"])
  loci=length(alleles[,1])
  num_alleles=.5*length(alleles[1,])

  allele_sizes=matrix(0,nrow=loci, ncol=num_alleles)
  allele_freqs=matrix(0,nrow=loci, ncol=num_alleles)
     true_genos=array(data=0, dim=c(N, loci, num_alleles,2))
  #format allele sizes and frequencies
  for (k in 1:loci)
{
  for (m in 1:num_alleles)
{
     allele_sizes[k,m]=alleles[k,((2*m)-1)]
     allele_freqs[k,m]=alleles[k,(2*m)]
}
}

  #generate genotype based on allele frequencies
  for (i in 1:N)
{
  for (j in 1:loci)
{
  allele1=sample(allele_sizes[j,], 1, prob=allele_freqs[j,])
  allele2=sample(allele_sizes[j,], 1, prob=allele_freqs[j,])
  for (r in 1:num_alleles)
{
  true_genos[i,j,r,1]=allele_sizes[j,r]
  if (allele1==allele_sizes[j,r])
true_genos[i,j,r,2]=true_genos[i,j,r,2]+1
  if (allele2==allele_sizes[j,r])
true_genos[i,j,r,2]=true_genos[i,j,r,2]+1
}
}
}
  rm(N,loci,num_alleles,allele_sizes,allele_freqs,k,m,i,j,allele1,allele2,
   r,params,alleles)
  return(true_genos)
}

##############################################################
#function to generate capture probabilities
gen_capt_probs=function(params)
{
  N=as.numeric(params["N"])
  capt_dist=params["capt_dist"]
  capt_mean=as.numeric(params["capt_mean"])
  capt_var=as.numeric(params["capt_var"])
  capt_attempts=as.numeric(params["capt_attempts"])
  #calculate probability of capture per attempt
  #from probability of capture at least once in occasion
  high=capt_mean+2*(capt_var^(1/2))
  low=capt_mean-2*(capt_var^(1/2))
  per_high=1-((1-high)^(1/capt_attempts))
  per_mean=1-((1-capt_mean)^(1/capt_attempts))
  per_low=1-((1-low)^(1/capt_attempts))
  high_se=(per_high-per_mean)/2
  low_se=(per_mean-per_low)/2
  per_var=((high_se+low_se)/2)^2

  #generate capt probs from beta
  if (capt_dist=="beta")
   {
     alpha=per_mean*((per_mean*(1-per_mean)/per_var)-1)
  beta=(1-per_mean)*((per_mean*(1-per_mean)/per_var)-1)
     capt_probs=rbeta(N, alpha, beta)
   }
  else
{
  #generate capt probs from uniform
  if (capt_dist=="uniform") print("uniform")
     else print("error: unknown capture probability distribution")
   }
      rm(N,capt_dist,capt_mean,capt_var,capt_attempts,high,low,per_high,
       per_mean,per_low,high_se,low_se,per_var,alpha,beta,params)
  return(capt_probs)
}

##############################################################
#function to generate sample
gen_sample=function(params,capt_probs)
{
  N=as.numeric(params["N"])
  occasions=as.numeric(params["occasions"])
  capt_attempts=as.numeric(params["capt_attempts"])

  true_capt_hx=matrix(0,nrow=N, ncol=occasions)

  #generate capture histories
  for (i in 1:occasions)
{
  for (j in 1:N)
{
  n=0
  for (k in 1:capt_attempts)
{
  test=runif(1,0,1)
     if (test<capt_probs[j]) n=n+1
}
  true_capt_hx[j,i]=n
}
}
  rm(N, occasions,capt_attempts,i,j,k,n,test,params,capt_probs)
  return(true_capt_hx)
}

##############################################################
#function to prep and call genotype function
genotype=function(params,alleles,true_genos, true_capt_hx, num_alleles,
loci,samples)
{
  N=as.numeric(params["N"])
  occasions=as.numeric(params["occasions"])
  capt_attempts=as.numeric(params["capt_attempts"])
  PCR_reps=as.numeric(params["PCR_reps"])
  PCR_eff_mean=as.numeric(params["PCR_eff_mean"])
  PCR_eff_sd=as.numeric(params["PCR_eff_sd"])
  stutter_min=as.numeric(params["stutter_min"])
  stutter_max=as.numeric(params["stutter_max"])
  cells_min=as.numeric(params["cells_min"])
  cells_max=as.numeric(params["cells_max"])
  qual_max=as.numeric(params["qual_max"])
  qual_min=as.numeric(params["qual_min"])

  genotypes=array(0,dim=c(samples, PCR_reps, loci, num_alleles,2))

  #generate PCR efficiencies
  alpha=PCR_eff_mean*((PCR_eff_mean*(1-PCR_eff_mean)/(PCR_eff_sd^2))-1)
  beta=(1-PCR_eff_mean)*((PCR_eff_mean*(1-PCR_eff_mean)/(PCR_eff_sd^2))-1)
  PCR_effs=rbeta(n=loci, alpha, beta)

  #generate stutter efficiencies
  stutter_effs=runif(n=loci,stutter_min,stutter_max)

  #count of samples genotyped
  n=1
  for (i in 1:N)
{
  for (j in 2:(occasions+1))
{
  k=true_capt_hx[i,(j-1)]
  while (k>0)
{
  num_cells=sample(c(cells_min:cells_max),size=1)
     quality=runif(n=1,qual_min,qual_max)
  for (l in 1:PCR_reps)
{
  for (m in 1:loci)
{
     genotypes[n,l,m,,]=gen_geno(params,alleles[m,],true_genos[i,m,,],
      loci,num_alleles, num_cells,quality,PCR_effs[m],stutter_effs[m])
   }
}
  k=k-1
  n=n+1
}
}
}
  rm(N,occasions,capt_attempts,PCR_reps,PCR_eff_mean,PCR_eff_sd,
   stutter_min,stutter_max,cells_min,cells_max,qual_max,qual_min,
   alpha,beta,PCR_effs,stutter_effs,i,j,k,num_cells,quality,l,m,
   params,alleles,true_genos,true_capt_hx, num_alleles,loci,samples,n)
  return(genotypes)
}

##############################################################
#function to generate genotypes
gen_geno=function(params,locus_alleles,geno,loci,num_alleles,num_cells,
quality,PCR_eff,stutter_eff)
{
  extract_eff=as.numeric(params["extract_eff"])
  aliquot_eff=as.numeric(params["aliquot_eff"])
  PCR_cycles=as.numeric(params["PCR_cycles"])
  prob_contam=as.numeric(params["prob_contam"])

     alls=vector("numeric",length=num_alleles)

  geno[,2]=geno[,2]*num_cells
  #extraction
  geno[,2]=rbinom(n=num_alleles, size=geno[,2], prob=extract_eff)
  #aliquot
  geno[,2]=rbinom(n=num_alleles, size=geno[,2], prob=aliquot_eff)
  #degredation
  geno[,2]=rbinom(n=num_alleles, size=geno[,2],
   prob=(1-(1-quality)*.0035)^geno[,1])
  #contamination
  test_contam=runif(1,0,1)
  if (test_contam<prob_contam)
{
  for (a in 1:num_alleles)
   {
  if (locus_alleles[(2*a)]>0) alls[a]=1
}
  y=sample(c(1:num_alleles), size=1,prob=alls)
  geno[y,2]=geno[y,2]+1
  rm(a,y)
}

  #zero out null
  for (i in 1:num_alleles)
   {
  if (geno[i,1]==999) geno[i,2]=0
}

  #PCR with sutter
  for (j in 1:PCR_cycles)
{
  new=rbinom(n=num_alleles, size=geno[,2], prob=PCR_eff)
  stutter=rbinom(n=num_alleles, size=new[], prob=stutter_eff)
  x=new-stutter
  final=x[1:(num_alleles-1)]+stutter[2:num_alleles]
     final[num_alleles]=x[num_alleles]
  geno[,2]=geno[,2]+final[]
}

  rm(params,locus_alleles,loci,num_alleles,num_cells,quality,PCR_eff,
   stutter_eff,extract_eff,aliquot_eff,PCR_cycles,prob_contam,alls,
   test_contam,i,j,new,stutter,x,final)
  return(geno)
}
```

```
###############################################################
#function to prep and call function to assign probs
assign_probs=function(params,genotypes, num_alleles,loci,samples,PCR_reps)
{
  rep_probs=array(0,dim=c(samples, PCR_reps, loci, num_alleles+1,
   num_alleles+1))

  #call fn to assign probs
  for (i in 1:samples)
{
  for (l in 1:PCR_reps)
{
  for (m in 1:loci)
{
     rep_probs[i,l,m,,]=probs(params,genotypes[i,l,m,,],num_alleles)
}
}
}

  rm(params,genotypes, num_alleles,loci, PCR_reps,i,l,m,samples)
  return(rep_probs)
}

##############################################################
#function to assign probs
probs=function(params,genotype_rep,num_alleles)
{
  noise_thresh=as.numeric(params["noise_thresh"])
  homoz_thresh_int=as.numeric(params["homoz_thresh_int"])
  homoz_thresh_slope=as.numeric(params["homoz_thresh_slope"])
  heteroz_thresh_int=as.numeric(params["heteroz_thresh_int"])
  heteroz_thresh_slope=as.numeric(params["heteroz_thresh_slope"])
  hb_thresh=as.numeric(params["hb_thresh"])
  p_stut=as.numeric(params["p_stut"])
  stut_size_diff=as.numeric(params["stut_size_diff"])

  prob=array(0, dim=c(num_alleles+1,num_alleles+1))
  hmt=vector(length=num_alleles)
  hrt=vector(length=num_alleles)
  pd=vector(length=num_alleles)
  pt=vector(length=num_alleles)
      het_prob=array(data=0,dim=c(num_alleles,num_alleles))
      stut=vector(mode="numeric",length=(num_alleles+1))

  #remove small stutters
      for (f in 2:num_alleles)
{
  if (genotype_rep[f,2]>0)
{
  if ((genotype_rep[(f-1),2]/genotype_rep[f,2])<hb_thresh)
   genotype_rep[(f-1),2]=0
}
}

  #homoz thresh
  hmt[]=homoz_thresh_slope*genotype_rep[,1]+homoz_thresh_int-noise_thresh
  #hetero thresh
  hrt[]=heteroz_thresh_slope*genotype_rep[,1]
   +heteroz_thresh_int-noise_thresh
  #subtract noise
  genotype_rep[,2]=genotype_rep[,2]-noise_thresh

  bands=0
  for (i in 1:num_alleles)
{
  #count bands, zero noise
  if (genotype_rep[i,2]<=0) genotype_rep[i,2]=0
   else bands=bands+1
  #p of drop out
     if (genotype_rep[i,2]<hmt[i]) pd[i]=1-(genotype_rep[i,2]/hmt[i])
       else pd[i]=0
     #p of true allele
     if (genotype_rep[i,2]<hrt[i]) pt[i]=genotype_rep[i,2]/hrt[i]
       else pt[i]=1
}

  #assign probabilites if no bands amplified--geno NN
  if (bands==0)  prob[(num_alleles+1), (num_alleles+1)]=1

  #assign probabilites if 1 band amplified--genos ii,iN
  if (bands==1)
{
  for (i in 1:num_alleles)
    {
      if (genotype_rep[i,2]>0)
```

```
        {
      prob[i,i]=1-pd[i]
      prob[i,(num_alleles+1)]=pd[i]
  break
        }
    }
}

  #assign probabilites if more than one band amplified
  if (bands>1)
   {
     #genos ii,iN
     for (i in 1:num_alleles)
      {
        if (genotype_rep[i,2]>0)
{
  prob[i,i]=(1-pd[i])*pt[i]
           prob[i,(num_alleles+1)]=pd[i]*pt[i]
  for (j in 1:num_alleles)
  {
    if (i!=j)
     {
       prob[i,i]=prob[i,i]*(1-pt[j])
       prob[i,(num_alleles+1)]=prob[i,(num_alleles+1)]*(1-pt[j])
     }
  }
}
      }
     #genos ij
      heteros=1-sum(prob)
      for (i in 1:(num_alleles-1))
       {
         for (j in (i+1):num_alleles)
          {
            #relative heteroz proportion
            het_prob[i,j]=genotype_rep[i,2]*genotype_rep[j,2]
             /(hrt[i]*hrt[j])
          }
       }
      sum_het=sum(het_prob)

      #normalize heteroz to sum all genos to one
      for (i in 1:(num_alleles-1))
```

```
       {
         for (j in (i+1):num_alleles)
          {
            prob[i,j]=heteros*het_prob[i,j]/sum_het
          }
       }

  #reassign probabilites if stutter positioning
      for (i in 1:(num_alleles-1))
{
  if (genotype_rep[i,2]>0)
   {
     if (genotype_rep[i,2]<genotype_rep[(i+1),2])
   {
     #track stutter alleles
     stut[i]=1
     for (j in 1:(num_alleles+1))
      {
        #decrease probabilities of genos with possible stutter alleles
         prob[i,j]=prob[i,j]*genotype_rep[i,2]/genotype_rep[(i+1),2]
         prob[j,i]=prob[j,i]*genotype_rep[i,2]/genotype_rep[(i+1),2]
      }
   }
   }
}

  #redistribute probs from stutter to non-stutter
  redistr_prob=1-sum(prob)
  non_stut_sum=0
  #calc total prob of genos with no stutter
  for (i in 1:(num_alleles+1))
   {
     if (stut[i]==0)
      {
        for (j in 1:(num_alleles+1))
         {
           if (stut[j]==0)
            {
              non_stut_sum=non_stut_sum+prob[i,j]
            }
         }
      }
   }
  #reassign probabilites to non-stutter positioning
  for (i in 1:(num_alleles+1))
   {
     if (stut[i]==0)
      {
        for (j in 1:(num_alleles+1))
         {
           if (stut[j]==0)
            {
              if (non_stut_sum!=0)
             {
             prob[i,j]=prob[i,j]+prob[i,j]*redistr_prob/non_stut_sum
             }
            }
         }
      }
   }
  rm(heteros,sum_het,redistr_prob,non_stut_sum)
}

  #normalize if not sum to 1
  x=round(sum(prob),digits=2)
  if (x!=1)
   {
     prob=prob/sum(prob)
}

  #diagonalize matrix so geno j,i is added to geno i,j
  for (i in 1:num_alleles)
   {
     for (j in (i+1):(num_alleles+1))
      {
        prob[i,j]=prob[i,j]+prob[j,i]
        prob[j,i]=0
      }
   }

  rm(params,genotype_rep,num_alleles,noise_thresh,homoz_thresh_int,
   homoz_thresh_slope,heteroz_thresh_int,heteroz_thresh_slope,
   hb_thresh,p_stut,stut_size_diff,hmt,hrt,pd,pt,het_prob,stut,f,
   bands,i,j,x)
  return(prob)
}
```

```
###############################################################
#calls function to average reps
average_geno_probs=function(alleles,num_alleles,loci, geno_rep_probs,
samples,PCR_reps)
{
  av_geno_probs=array(0, dim=c(samples,loci,num_alleles,num_alleles,
   PCR_reps))

  #call fn  to average geno probs
  for (i in 1:samples)
   {
  for (m in 1:loci)
{
  for (l in 1:PCR_reps)
   {
     av_geno_probs[i,m,,,l]=av_genos(alleles,num_alleles,
      loci,geno_rep_probs[i,1:l,m,,],l,m)
       }
}
   }
  rm(alleles,num_alleles,loci, geno_rep_probs,PCR_reps,i,m,l,samples)
  return(av_geno_probs)
}

###############################################################
#function to average genotype probs from reps
av_genos=function(alleles,num_alleles,loci,rep_probs,num_reps,locus)
{
  pre_genos_av=array(0, dim=c((num_alleles+1),(num_alleles+1)))
  al_freq=vector(mode="numeric",length=(num_alleles+1))
  norm_al_freq=vector(mode="numeric",length=num_alleles)
  genos_av=array(0,dim=c(num_alleles,num_alleles))

  #if one rep--just use those calls
  if (num_reps==1)
   {
     pre_genos_av=rep_probs
   }

  #if more than one rep--average
  if (num_reps>1)
   {
```

```
  for (i in 1:num_reps)
   {
     pre_genos_av=pre_genos_av+rep_probs[i,,]
   }
  pre_genos_av=pre_genos_av/num_reps
}

  #calculate allele frequency for a sample's reps from genotype frequency
  for (i in 1:(num_alleles+1))
   {
     for (j in i:(num_alleles+1))
      {
        al_freq[i]=al_freq[i]+(.5*pre_genos_av[i,j])
        al_freq[j]=al_freq[j]+(.5*pre_genos_av[i,j])
      }
   }

  #recalculate allele frequencies without N alleles
  al_freq_total=1-al_freq[(num_alleles+1)]
  for (i in 1:num_alleles)
   {
     if (al_freq_total>0)
      {
     norm_al_freq[i]=al_freq[i]/al_freq_total
      }
     #use population allele freqs if no alleles amplified
     if (al_freq_total==0)
      {
  a=1
  b=1
  while(a<(2*num_alleles))
   {
     norm_al_freq[b]=alleles[locus,(a+1)]
     if (alleles[locus,a]==999) norm_al_freq[b]=0
     a=a+2
     b=b+1
   }
  norm_al_freq=norm_al_freq/sum(norm_al_freq)
  rm(a,b)
      }
   }

 #redistribute probs from N alleles based on sample allele freqs
 for (i in 1:num_alleles)
  {
    for (j in i:num_alleles)
     {
       pre_genos_av[i,j]=pre_genos_av[i,j]
        +pre_genos_av[i,(num_alleles+1)]*norm_al_freq[j]
     }
  pre_genos_av[i,(num_alleles+1)]=0
  }

  #redistribute probs from NN genos
  if (pre_genos_av[(num_alleles+1),(num_alleles+1)]>0)
   {
     for (i in 1:num_alleles)
      {
        #redistribute prob to homozygotes
         pre_genos_av[i,i]=pre_genos_av[i,i]
          +pre_genos_av[(num_alleles+1),(num_alleles+1)]
          *norm_al_freq[i]*norm_al_freq[i]
        if ((i+1)<=num_alleles)
         {
          for (j in (i+1):num_alleles)
          {
            #redistribute prob to heterozygotes
            pre_genos_av[i,j]=pre_genos_av[i,j]
             +pre_genos_av[(num_alleles+1),(num_alleles+1)]
             *2*norm_al_freq[i]*norm_al_freq[j]
          }
       }
      }
      pre_genos_av[(num_alleles+1),(num_alleles+1)]=0
   }

  genos_av=pre_genos_av[1:num_alleles,1:num_alleles]

  rm(alleles,num_alleles,loci,rep_probs,num_reps,locus,pre_genos_av,
   al_freq,norm_al_freq,i,j,al_freq_total)
  return(genos_av)
}

################################################################
#function to sample from geno probs
geno_sample=function(num_alleles,loci,av_geno_probs,samples,PCR_reps)
```

```
{
  sample_genos=array(0, dim=c(samples,PCR_reps,loci))
  #matrix of 2 digit genos
  genotypes=matrix(nrow=num_alleles,ncol=num_alleles)

  #make matrix of possible genos
  for (a in 1:num_alleles)
   {
     for (b in 1:num_alleles)
      {
        genotypes[a,b]=100*a+b
      }
   }

  for (i in 1:samples)
   {
  for (l in 1:PCR_reps)
   {
     for (m in 1:loci)
      {
  sample_genos[i,l,m]=sample(genotypes,size=1,p=av_geno_probs[i,m,,,l])
     }
      }
   }

  rm(num_alleles,loci,av_geno_probs,PCR_reps,genotypes,a,b,i,l,m,samples)
  return(sample_genos)
}

###############################################################
#function to create capture hx
gen_capt_hx=function(num_alleles,loci,sample_genos,samples,samp_occ,
PCR_reps,occasions)
{
  unique_genos=vector("numeric",length=PCR_reps)
  #array tracking which samples matched
  matched=array(1,dim=c(PCR_reps,samples))
  capture_hx=array(0,dim=c(PCR_reps,samples,occasions))

  for (e in 1:PCR_reps)
   {
     #choose sample
  for (i in 1:samples)
```

```
   {
  if (matched[e,i]==1)
      {
          matched[e,i]=0
         unique_genos[e]=unique_genos[e]+1
        capture_hx[e,unique_genos[e],samp_occ[i]]
         =capture_hx[e,unique_genos[e],samp_occ[i]]+1
        #compare to all other samples
     for (a in 1:samples)
   {
     if (matched[e,a]==1)
            {
        if(identical(sample_genos[i,e,],sample_genos[a,e,])==TRUE)
    {
    capture_hx[e,unique_genos[e],samp_occ[a]]
     =capture_hx[e,unique_genos[e],samp_occ[a]]+1
     matched[e,a]=0
         }
           }
      }
      }
   }
}

  #remove multiple captures in an occasion
  for (i in 1:PCR_reps)
   {
     for (j in 1:samples)
      {
        for (k in 1:occasions)
         {
           if (capture_hx[i,j,k]>1) capture_hx[i,j,k]=1
         }
      }
   }

  rm(num_alleles,loci,sample_genos,occasions,PCR_reps,unique_genos,
   samples,matched,i,j,k,e,a,samp_occ)
  return(capture_hx)
}

###############################################################
#function to calculate capture frequencies
```

```
calc_freqs=function(params,capture_hx,true_capt_hx)
{
  occasions=as.numeric(params["occasions"])
    PCR_reps=as.numeric(params["PCR_reps"])
      N=as.numeric(params["N"])
      #number of unique genotypes identified
  samples=nrow(capture_hx[1,,])

  captures=matrix(0,nrow=PCR_reps+1,ncol=max(samples,N))
  freqs=matrix(0,nrow=PCR_reps+1,ncol=occasions)

  #calculate number of occasions in which a unique geno captured
  for (i in 1:PCR_reps)
   {
     for (j in 1:samples)
      {
  captures[i,j]=sum(capture_hx[i,j,])
      }
   }

  #calc for true capt hx, store in at end of matrix
  for (i in 1:N)
   {
     for (j in 1:occasions)
      {
        if (true_capt_hx[i,j]>=1)
         captures[PCR_reps+1,i]=captures[PCR_reps+1,i]+1
      }
   }

  #calculate frequencies
  for (i in 1:(PCR_reps+1))
   {
     for (j in 1:max(samples,N))
      {
        k=captures[i,j]
        if (k>0) freqs[i,k]=freqs[i,k]+1
      }
   }

  rm(params,capture_hx,true_capt_hx,occasions,PCR_reps,N,samples,
   captures,i,j,k)
  return(freqs)
```

```
}

###############################################################
#function to calculate first order jackknife est
jackknife=function(params,freqs)
{
  occasions=as.numeric(params["occasions"])
    PCR_reps=as.numeric(params["PCR_reps"])
     N=as.numeric(params["N"])

    ests=vector("numeric",length=PCR_reps+1)

  for (i in 1:(PCR_reps+1))
   {
     ests[i]=sum(freqs[i,])+((occasions-1)/occasions)*freqs[i,1]
     if (ests[i]>(3*N))
      {
        ests[i]=NA
     print(paste("jack too big PCR rep",i))
   }
   }

  rm(params,occasions,PCR_reps,freqs,i)
  return (ests)
}

##############################################################
#function to calculate first order chao est
chao=function(params,freqs)
{
  occasions=as.numeric(params["occasions"])
    PCR_reps=as.numeric(params["PCR_reps"])
     N=as.numeric(params["N"])

      ests=vector("numeric",length=PCR_reps+1)

  for (i in 1:(PCR_reps+1))
   {
     ests[i]=sum(freqs[i,])+(freqs[i,1]*(freqs[i,1]-1))/(2*(freqs[i,2]+1))
     if (ests[i]>(3*N))
      {
        ests[i]=NA
     print(paste("chao too big PCR rep",i))
   }
   }

  rm(params,occasions,PCR_reps,freqs,i)
    return (ests)
}

##############################################################
#function to calculate m0
m0=function(params,freqs)
{
  occasions=as.numeric(params["occasions"])
    PCR_reps=as.numeric(params["PCR_reps"])
     N=as.numeric(params["N"])

  ests=vector("numeric",length=PCR_reps+1)

  for (i in 1:(PCR_reps+1))
   {
  n=0
  for (j in 1:occasions)
{
  n=n+j*freqs[i,j]
}
  M=sum(freqs[i,])
  likelihood=function(NN)
{
     -(lfactorial(exp(NN)+M-1)-lfactorial(exp(NN)+M-1-M)
     +log((n/(occasions*(exp(NN)+M-1)))^n)
     +log((1-(n/(occasions*(exp(NN)+M-1))))^(occasions*(exp(NN)+M-1)-n)))
}
  ln_ests_temp=try(nlm(likelihood,p=log(1))$estimate)
  if (is.numeric(ln_ests_temp)==FALSE)
   {
     print(paste("no estimate for mo PCR rep",i))
   }
   else
   {
        ests[i]=exp(ln_ests_temp)+M-1
        if (ests[i]>(3*N))
      {
        ests[i]=NA
       print(paste("mo too big PCR rep",i))
    }
      }
   }
  ests=as.numeric(ests)

  rm(params,occasions,PCR_reps,freqs,i,j,n,M,likelihood,ln_ests_temp)
  return (ests)
}

##############################################################
#function to calculate two point mixture
twopt=function(params,freqs)
{
  occasions=as.numeric(params["occasions"])
    PCR_reps=as.numeric(params["PCR_reps"])
     N=as.numeric(params["N"])

  ests=vector("numeric",length=PCR_reps+1)

  for (i in 1:(PCR_reps+1))
{
  M=sum(freqs[i,])
  likelihood=function(p)
{
  f=0
  f=lfactorial(exp(p[4])+M-1)-lfactorial(exp(p[4])+M-1-M)
  +(exp(p[4])+M-1-M)*log((exp(p[3])/(1+exp(p[3])))
  *((1-(exp(p[1])/(1+exp(p[1]))))^occasions)+(1-(exp(p[3])/(1+exp(p[3]))))
  *((1-(exp(p[2])/(1+exp(p[2]))))^occasions))
  for (j in 1:occasions)
   {
     f=f+freqs[i,j]*log((exp(p[3])/(1+exp(p[3])))
     *((exp(p[1])/(1+exp(p[1])))^j)
     *((1-(exp(p[1])/(1+exp(p[1]))))^(occasions-j))
     +(1-(exp(p[3])/(1+exp(p[3]))))*((exp(p[2])/(1+exp(p[2])))^j)
     *((1-(exp(p[2])/(1+exp(p[2]))))^(occasions-j)))
   }
        f=-f
  return(f)
}

  ln_ests_temp=try(nlm(likelihood,p=c(-.4,.4,0,log(1)))$estimate[4])
  if (is.numeric(ln_ests_temp)==FALSE)
```

```
   {
     print("no estimate for two pt")
     print("PCR rep")
     print(i)
   }
   else
   {
     ests[i]=exp(ln_ests_temp)+M-1
        if (ests[i]>(3*N))
      {
        ests[i]=NA
      print(paste("2pt too big PCR rep",i))
    }
      }
}
  ests=as.numeric(ests)

  rm(params,occasions,PCR_reps,freqs,i,M,likelihood,ln_ests_temp)
 return (ests)
}

##############################################################
#find most likely geno
most_likely_genos=function(num_alleles,loci,av_geno_probs,samples,PCR_reps)
{
  likely_genos=array(0,dim=c(samples,PCR_reps,loci))

  for (i in 1:PCR_reps)
   {
     for (j in 1:samples)
      {
  for (m in 1:loci)
         {
            p=0
           for (a in 1:num_alleles)
            {
               for (b in a:num_alleles)
                {
                  if (av_geno_probs[j,m,a,b,i]>p)
                   {
                    p=av_geno_probs[j,m,a,b,i]
                    z=a
                   y=b
```

```
                    }
               }
            }
            likely_genos[j,i,m]=100*z+y
         }
      }
   }

  rm(num_alleles,loci,av_geno_probs,PCR_reps,i,j,m,p,a,b,z,y,samples)
  return(likely_genos)
}

##############################################################
#function to determine consensus genotype
consensus=function(loci, num_alleles, geno_rep_probs,samples,
PCR_reps,alleles)
{
  trad_alleles=array(0,dim=c(samples,PCR_reps,loci,2))
  trad_genos=array(0,dim=c(samples,PCR_reps,loci))
  count=array(0,dim=c(samples,loci,PCR_reps,num_alleles))
  thresh=vector(length=PCR_reps)
  call=array(0,dim=c(samples,loci,PCR_reps,num_alleles))

  #calculate consensus thresholds
  thresh=c(1:PCR_reps)*2/3
  thresh=round(thresh, digits=0)

  for (i in 1:samples)
{
  for (l in 1:PCR_reps)
{
  for (m in 1:loci)
{
  #reassign i,N genos to i,i
     for (n in 1:num_alleles)
      {
        geno_rep_probs[i,l,m,n,n]=geno_rep_probs[i,l,m,n,n]
         +geno_rep_probs[i,l,m,n,(num_alleles+1)]
   }
     #find most likely geno
     p=0
     z=0
     y=0
     for (a in 1:num_alleles)
   {
        for (b in a:num_alleles)
       {
          if (geno_rep_probs[i,l,m,a,b]>p)
            {
           p=geno_rep_probs[i,l,m,a,b]
           z=a
              y=b
            }
         }
   }
     trad_alleles[i,l,m,1]=z
     trad_alleles[i,l,m,2]=y
}
}
}

  for (i in 1:samples)
{
  for (m in 1:loci)
{
  for (l in 1:PCR_reps)
{
  #count number of times an allele called at each rep and the previous
  for (f in 1:l)
   {
     if (trad_alleles[i,f,m,1]>0)
      count[i,m,l,trad_alleles[i,f,m,1]]
      =count[i,m,l,trad_alleles[i,f,m,1]]+1
     if (trad_alleles[i,f,m,2]>0)
      count[i,m,l,trad_alleles[i,f,m,2]]
      =count[i,m,l,trad_alleles[i,f,m,2]]+1
   }
}

  #compare counts to consensus
  for (g in 1:PCR_reps)
   {
     for (h in 1:num_alleles)
      {
        if (count[i,m,g,h]>=thresh[g])
           call[i,m,g,h]=1
         else
           call[i,m,g,h]=0
   }
   }

  #determine geno
  for (d in 1:PCR_reps)
   {
     called=sum(call[i,m,d,])
     if (called==0)
      {
           #count how many alleles amplified
           amp=0
           for (j in 1:num_alleles)
            {
              if (count[i,m,d,j]>0)
                amp=amp+1
            }
  #if no alleles amplified
  #assign to homozygote of most common allele in population
  if (amp==0)
   {
  even=c(1:num_alleles)*2
  als=as.numeric(alleles[m,even])
  z=which.max(als)
  trad_genos[i,d,m]=100*z+z
  rm(even,als)
   }
  #if one allele amplified assign to homozygote of that allele
  if (amp==1)
   {
     z=which.max(count[i,m,d,])
     trad_genos[i,d,m]=100*z+z
   }
  #if more than one allele amplified assign geno
  if (amp>1)
   {
  for (aa in 1:num_alleles)
   {
         counts=count[i,m,d,]
         counts[aa]=count[i,m,d,(num_alleles-aa+1)]
       }
     z=num_alleles-which.max(counts)+1
```

```
     counts[num_alleles-z+1]=0
     y=num_alleles-which.max(counts)+1
     if (count[i,m,d,z]==count[i,m,d,y])
      {
         if (z<y)
          trad_genos[i,d,m]=100*z+y
         else
          trad_genos[i,d,m]=100*y+z
      }
      else
        trad_genos[i,d,m]=100*z+z
   }
  rm(amp,j,aa)
}

    if (called==1)
      {
  z=which.max(call[i,m,d,])
  trad_genos[i,d,m]=100*z+z
      }

     if (called==2)
     {
  counts=call[i,m,d,]
  z=which.max(counts)
  counts[z]=0
  y=which.max(counts)
  if (z>y)  trad_genos[i,d,m]=100*y+z
  else trad_genos[i,d,m]=100*z+y
      }

     if (called>2)
      {
     counts=count[i,m,d,]
        z=which.max(counts)
        counts[z]=0
       y=which.max(counts)
        twomax=counts[y]
        counts[y]=0
  if (z>y)  trad_genos[i,d,m]=100*y+z
  else trad_genos[i,d,m]=100*z+y
        for (a in 1:num_alleles)
         {
          if (counts[a]==twomax)
           {
              for (aa in 1:num_alleles)
                counts[aa]=count[i,m,d,(num_alleles-aa+1)]
           z=num_alleles-which.max(counts)+1
        counts[num_alleles-z+1]=0
       y=num_alleles-which.max(counts)+1
    if (z>y)  trad_genos[i,d,m]=100*y+z
  else trad_genos[i,d,m]=100*z+y
              break
}
         }
  rm(twomax)
      }
   }
}
}

  rm(alleles,loci, num_alleles,geno_rep_probs,PCR_reps,trad_alleles,
    count,thresh,call,z,y,i,l,m,p,a,b,f,g,h,d,called,counts,n,samples)
  return(trad_genos)
}

#####################################################################
#run full simulation
x=sims("params.txt", "allele.txt") #run full simulation
```

Appendix B

Parameter Input File

```
simid 1D # simulation identification number (1A-4D)
seed 719034 # simulation seed
N 200 # true population size (200,500)
occasions 5 # number of sampling occasions (5,10)
capt_mean 0.2 # mean probability of at least one capture per occasion (0.2,0.5)
capt_var 0.002 # variance of probabilities of at least one capture per occasion (0.002,0.02)
capt_dist beta # distribution to generate capture probabilities
capt_attempts 5 # number of capture attempts per sampling occasion
PCR_reps 10 # maximum number of PCR reps
cells_min 30 # minimum number of cells possible in the sample
cells_max 150 # maximum number of cells possible in the sample
PCR_cycles 34 # number of PCR cycles
extract_eff 0.46 # extraction efficiency
qual_min 0 # minimum quality score
qual_max 1 # maximum quality score
aliquot_eff 0.3 # aliquot efficiency
PCR_eff_mean 0.82 # mean PCR efficiency
PCR_eff_sd 0.06 # standard deviation of PCR efficiencies
stutter_min 0 # minimum possible stutter efficiency
stutter_max 0.04 # maximum possible stutter efficiency
prob_contam 0.008 # probability of undetected contamination per locus per PCR replicate
```

```
noise_thresh 20000000 # noise threshold
homoz_thresh_int 17000000000 # homozygous threshold intercept
homoz_thresh_slope -75000000 # homozygous threshold slope
heteroz_thresh_int 8700000000 # heterozygous threshold intercept
heteroz_thresh_slope -11000000 # heterozygous threshold slope
hb_thresh 0.32 # value relative to parent allele below which assumed to be stutter
stut_size_diff 1 # number of array positions over to stutter peak
num_sims 100 # number of simulations
num_models 4 # number of mark-recapture models used to analyze data
```

Appendix C

Allele Input Files

```
# Allele sizes and frequencies for populations with no null alleles
156 .15 158 .19 160 .26 162 .22 164 .18 166 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 #G10B
99 .31 101 0.0 103 .25 105 .13 107 .09 109 .02 111 .14 113 .04 115 .03 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 #G10C
135 .08 137 .15 139 .01 141 .05 143 0.0 145 .09 147 0.0 149 .15 151 0.0 153 0.0 155 .02 157 .05 159 .32 161 0.0 163 .02 165 .03 167 0.0 169 .03 998 0.0 998 0.0 #G10L
200 .01 202 0.0 204 .02 206 .04 208 .11 210 .14 212 .24 214 .10 216 .23 218 .06 220 .03 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 #G10M
149 0.0 151 .05 153 .18 155 .18 157 .15 159 .27 161 .10 163 0.0 165 0.0 167 .06 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 #G10P
125 .03 127 .12 129 .01 131 .01 133 .07 135 0.0 137 0.0 139 .06 141 0.0 143 0.0 145 .16 147 .25 149 .19 151 .03 153 .01 155 0.0 157 0.0 159 0.0 161 0.0 163 .04 #G10X
184 .16 186 0.0 188 .04 190 .03 192 .25 194 .41 196 .03 198 .08 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 #G1A
172 .14 174 .10 176 .47 178 .11 180 .01 182 .06 184 0.0 186 .03 188 .05 190 .01 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 #G1D
```

```
# Allele sizes and frequencies for populations with null alleles
156 .15 158 .19 160 .26 162 .22 999 .18 166 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 #G10B
99 .31 101 0.0 103 .25 105 .13 107 .09 109 .02 111 .14 113 .04 115 .03 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 #G10C
135 .08 137 .15 139 .01 141 .05 143 0.0 145 .09 147 0.0 149 .15 151 0.0 153 0.0 155 .02 157 .05 159 .32 161 0.0 163 .02 165 .03 167 0.0 169 .03 998 0.0 998 0.0 #G10L
200 .01 202 0.0 204 .02 206 .04 208 .11 210 .14 212 .24 214 .10 216 .23 218 .06 220 .03 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 #G10M
149 0.0 151 .05 153 .18 155 .18 157 .15 159 .27 161 .10 163 0.0 165 0.0 167 .06 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 #G10P
125 .03 127 .12 129 .01 131 .01 133 .07 135 0.0 137 0.0 139 .06 141 0.0 143 0.0 145 .16 147 .25 149 .19 151 .03 153 .01 155 0.0 157 0.0 159 0.0 161 0.0 163 .04 #G10X
184 .16 186 0.0 188 .04 190 .03 192 .25 194 .41 196 .03 999 .08 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 #G1A
172 .14 174 .10 176 .47 178 .11 180 .01 182 .06 184 0.0 186 .03 188 .05 190 .01 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 998 0.0 #G1D
# alleles ordered in increasing size
# size 998 are zero frequency alleles
# size 999 indicates a null allele
# all loci need to have same number of entries; if different, add zero frequency alleles at end of list
# first allele set is from West Slope black bears from Paetkau et al. (1998)
# second allele set is modified from West Slope black bears in Paetkau et al. (1998) to include null alleles
```

Appendix D

Results for All Simulation Scenarios

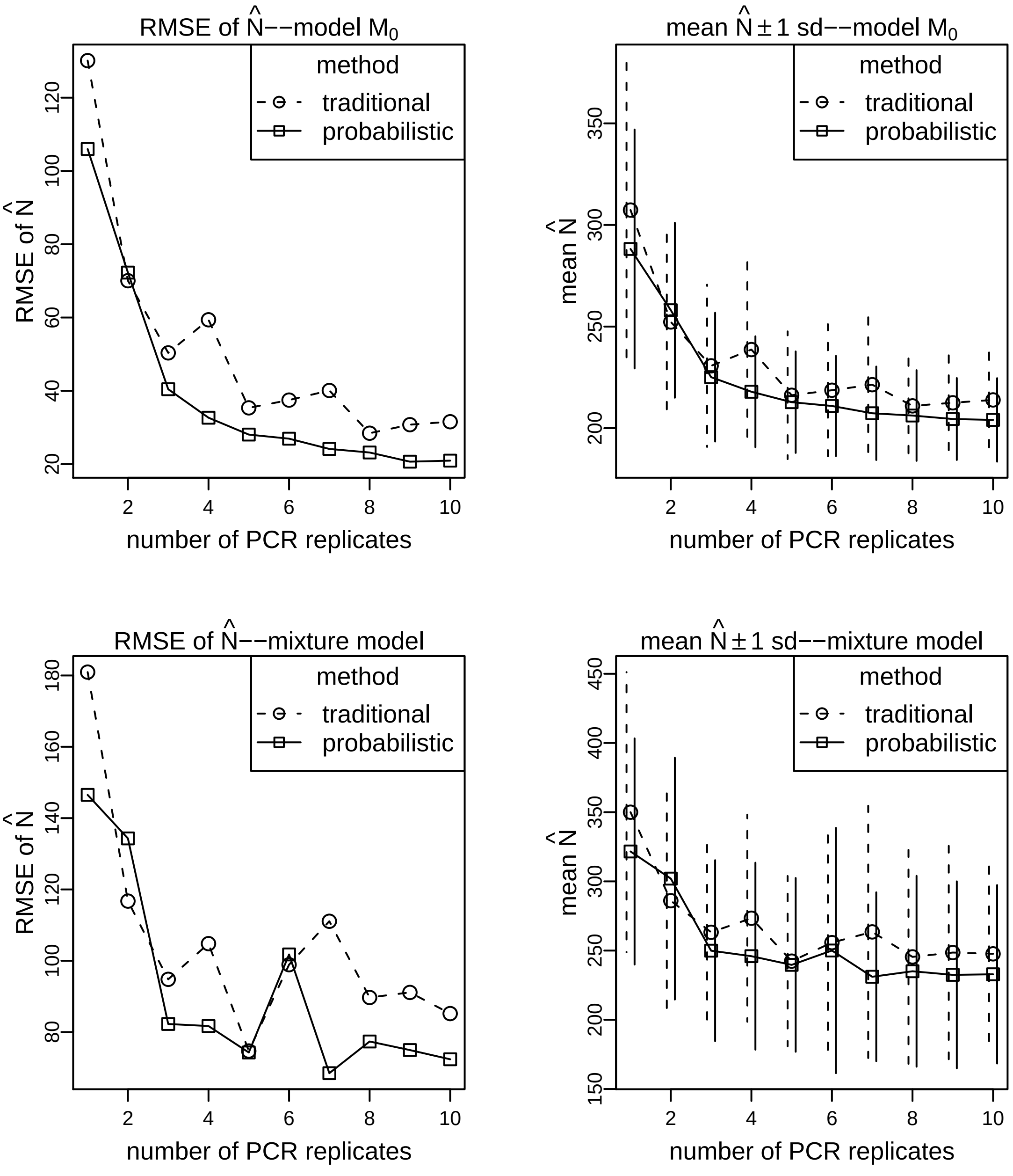


Figure D.1: Results for simulation 1: N=200, capture probability=.2, sampling occasions=5, no null alleles. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_0 and the two point mixture model for two methods—the traditional consensus method ("traditional") and the probabilistic method ("probabilistic").

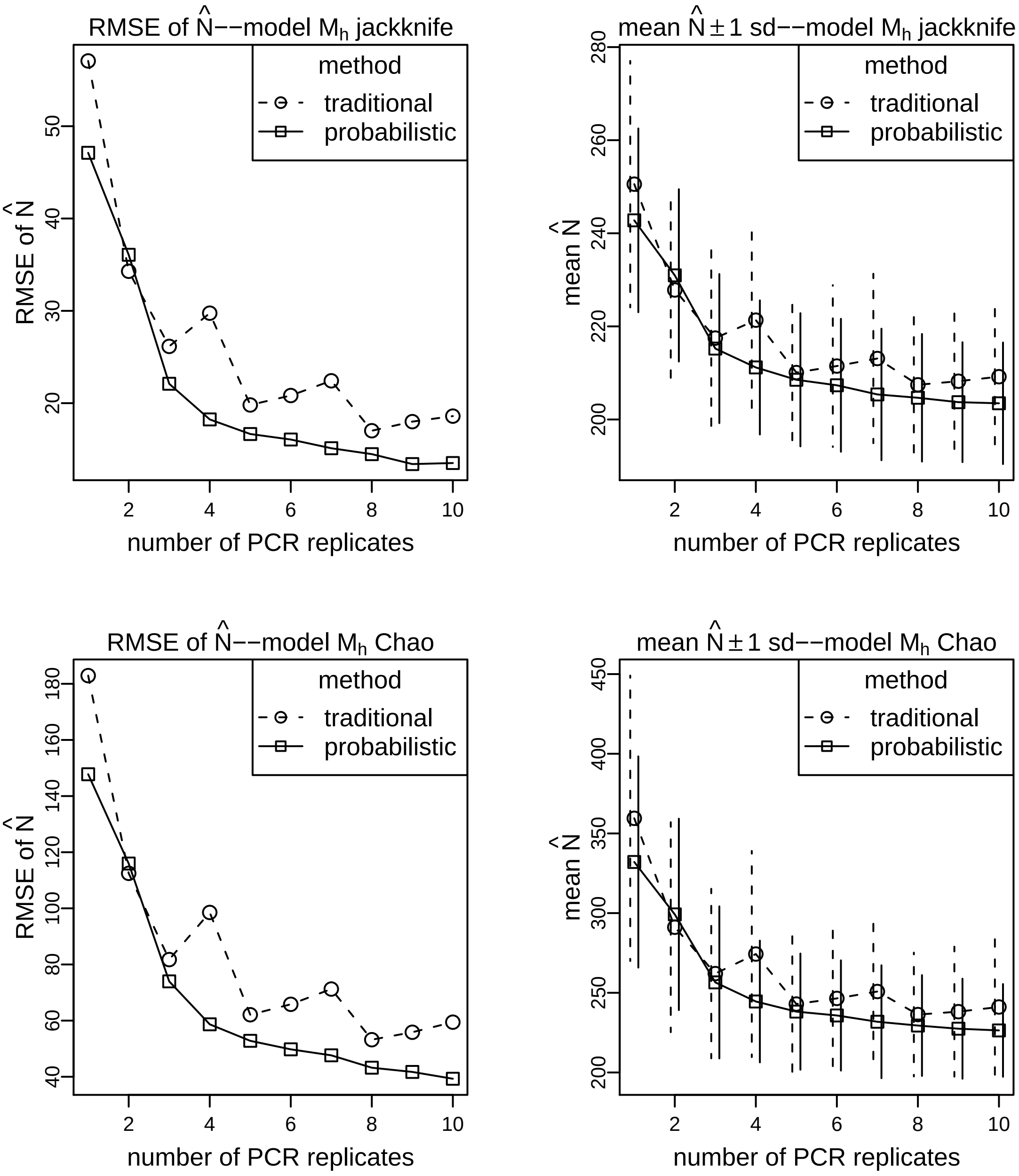


Figure D.2: Results for simulation 1: N=200, capture probability=.2, sampling occasions=5, no null alleles. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_h jackknife and model M_h Chao for two methods—the traditional consensus method ("traditional") and the probabilistic method ("probabilistic").

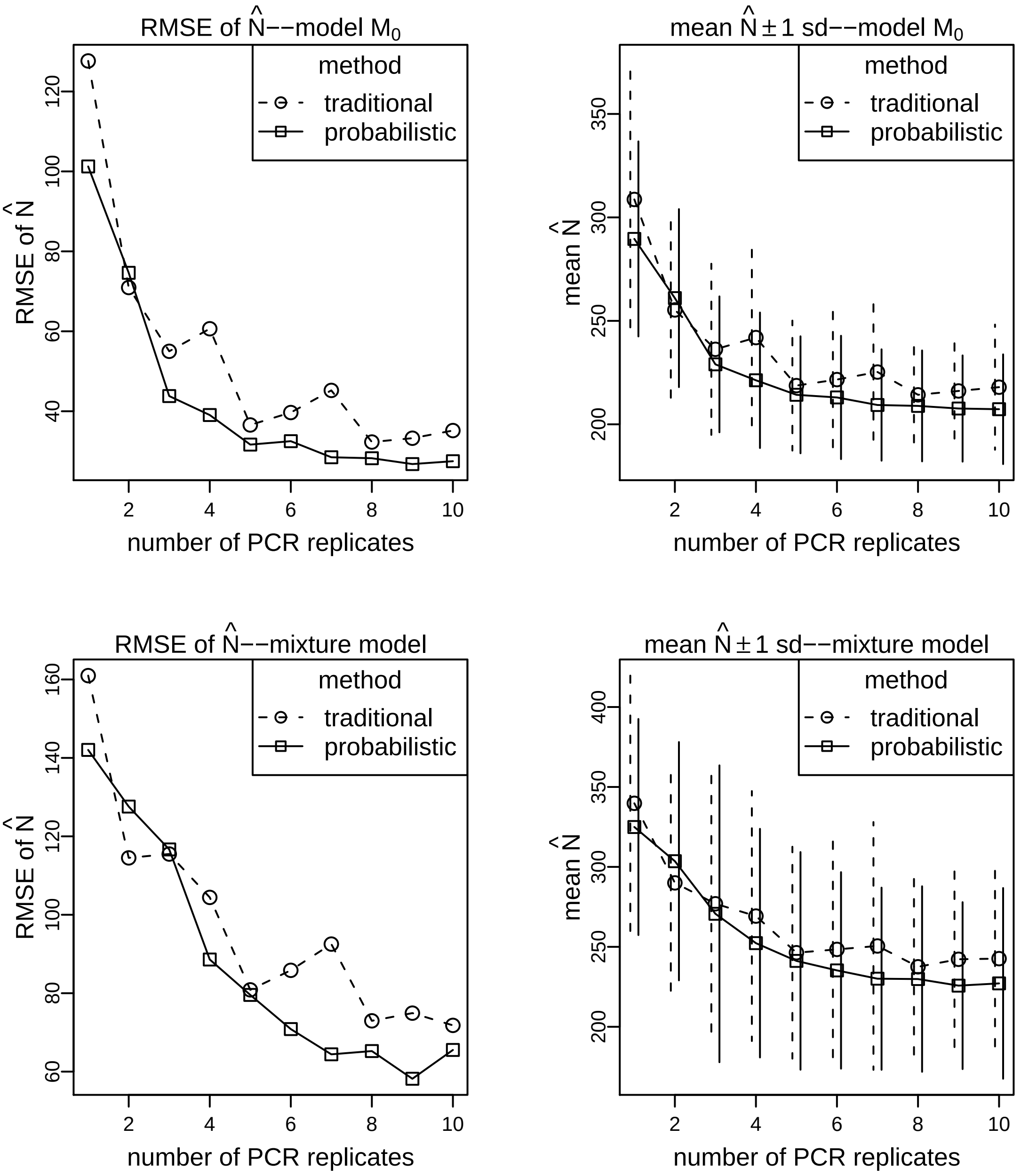


Figure D.3: Results for simulation 2: N=200, capture probability=.2, sampling occasions=5, null alleles present. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_0 and the two point mixture model for two methods—the traditional consensus method ("traditional") and the probabilistic method ("probabilistic").

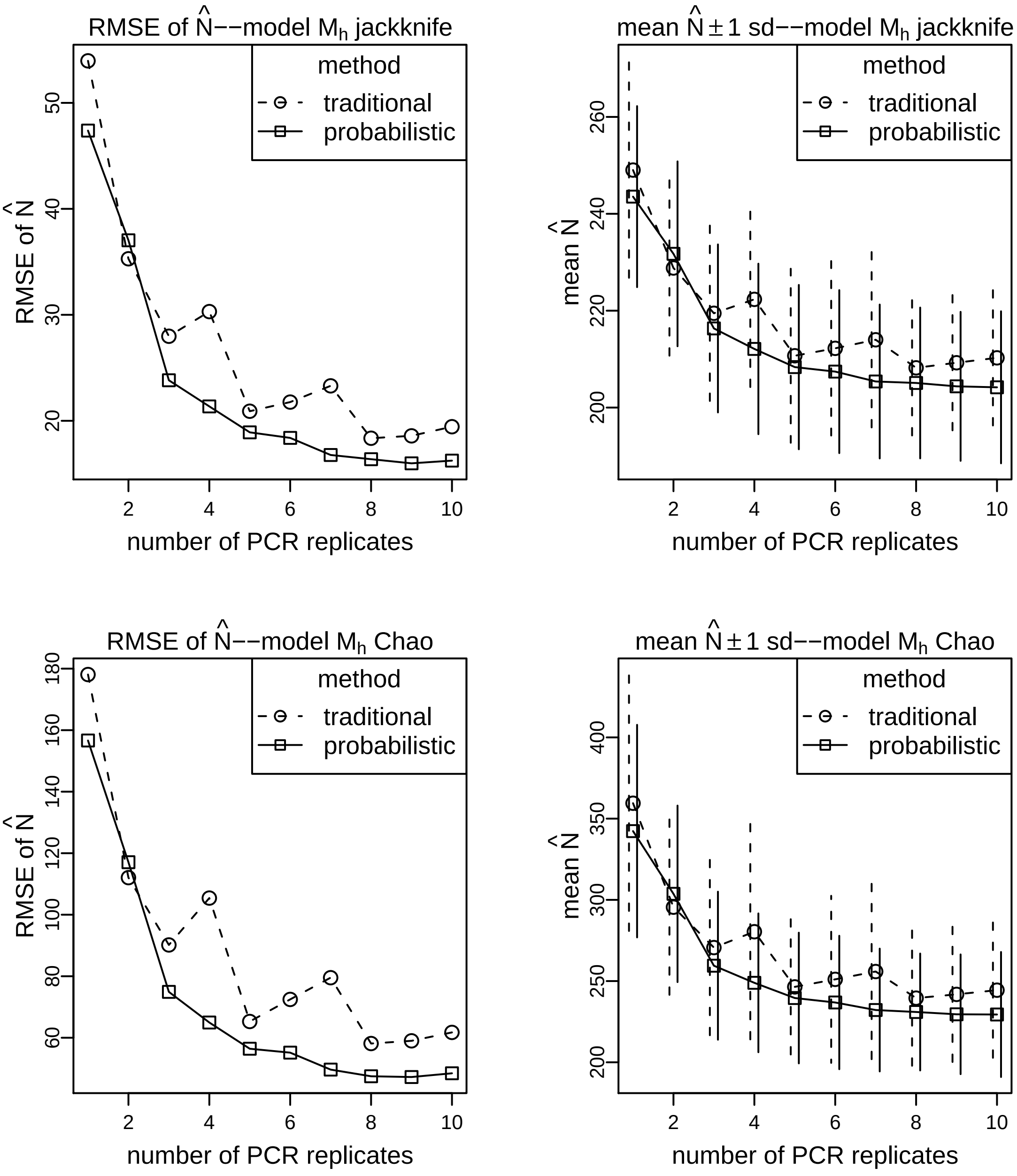


Figure D.4: Results for simulation 2: N=200, capture probability=.2, sampling occasions=5, null alleles present. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_h jackknife and model M_h Chao for two methods—the traditional consensus method ("traditional") and the probabilistic method ("probabilistic").

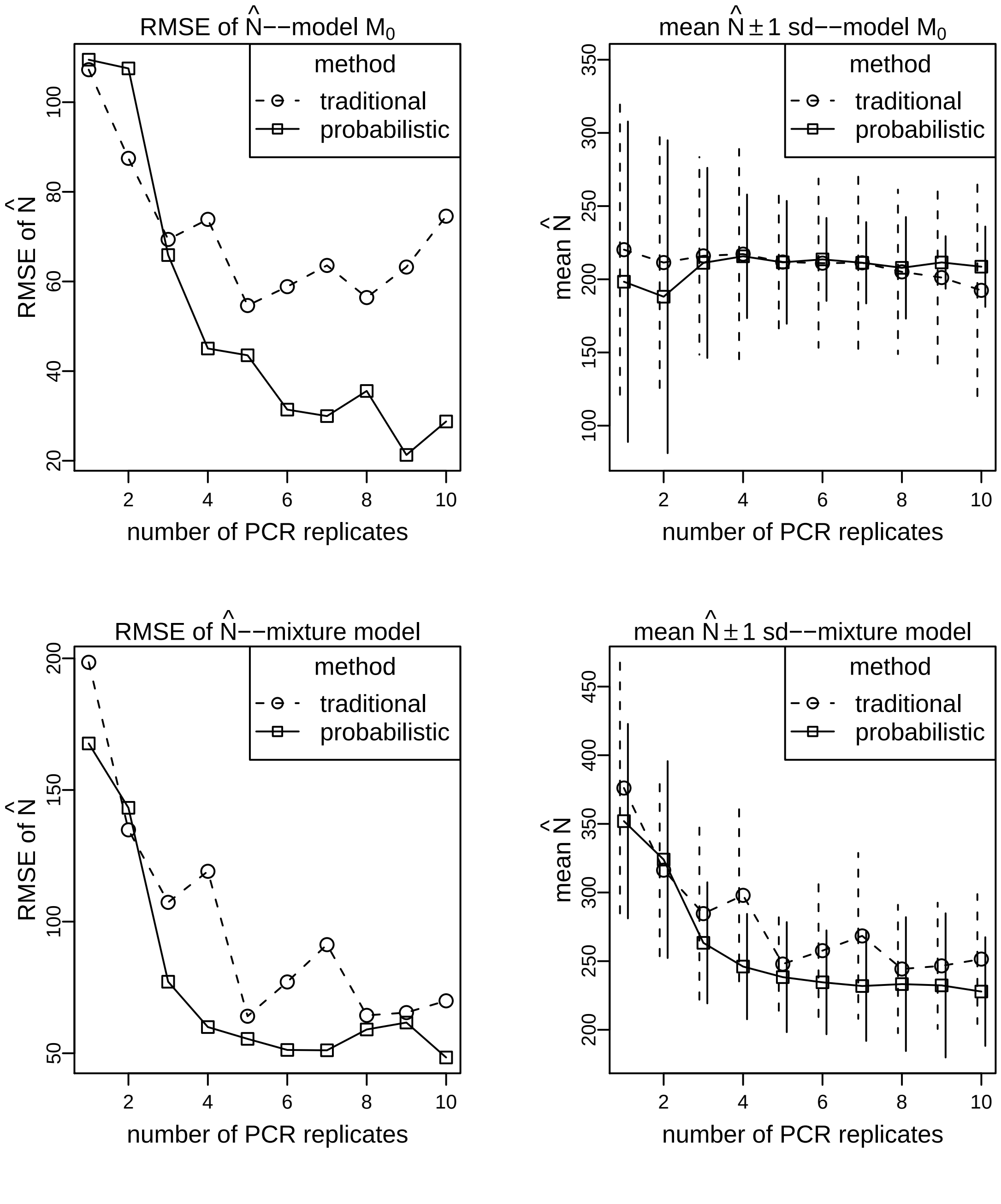


Figure D.5: Results for simulation 3: N=200, capture probability=.2, sampling occasions=10, no null alleles. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_0 and the two point mixture model for two methods—the traditional consensus method ("traditional") and the probabilistic method ("probabilistic").

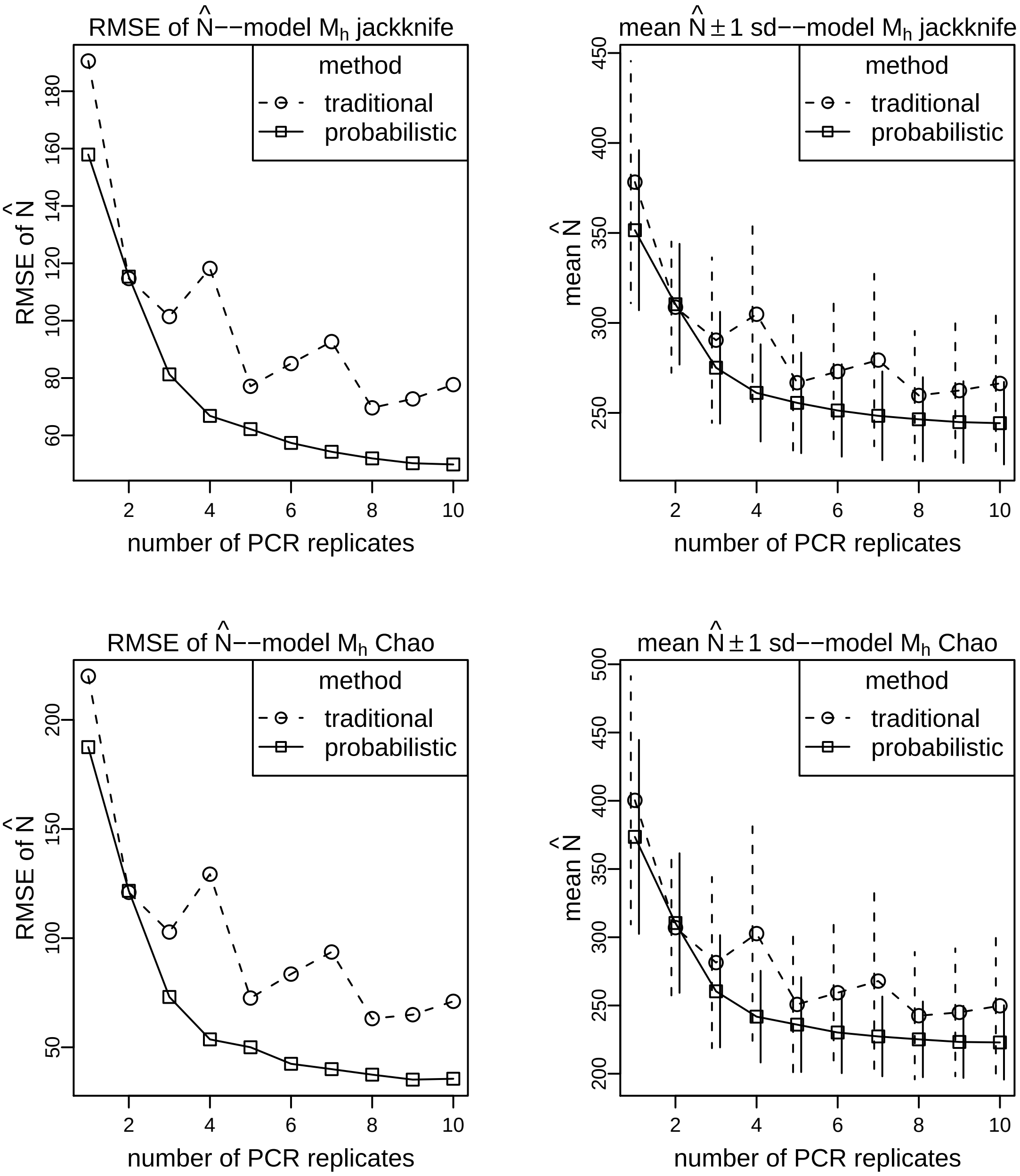


Figure D.6: Results for simulation 3: N=200, capture probability=.2, sampling occasions=10, no null alleles. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_h jackknife and model M_h Chao for two methods—the traditional consensus method ("traditional") and the probabilistic method ("probabilistic").

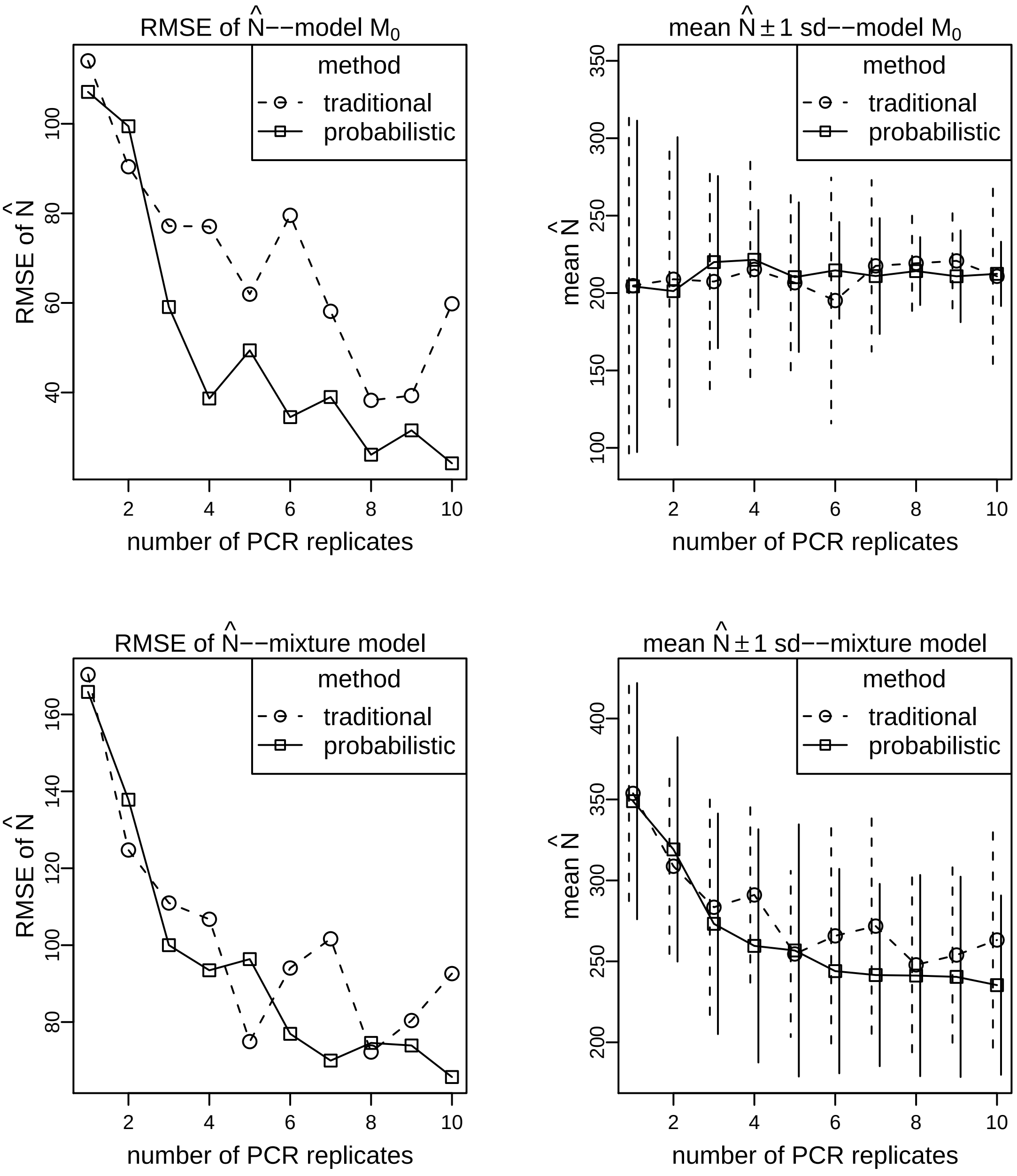


Figure D.7: Results for simulation 4: N=200, capture probability=.2, sampling occasions=10, null alleles present. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_0 and the two point mixture model for two methods—the traditional consensus method ("traditional") and the probabilistic method ("probabilistic").

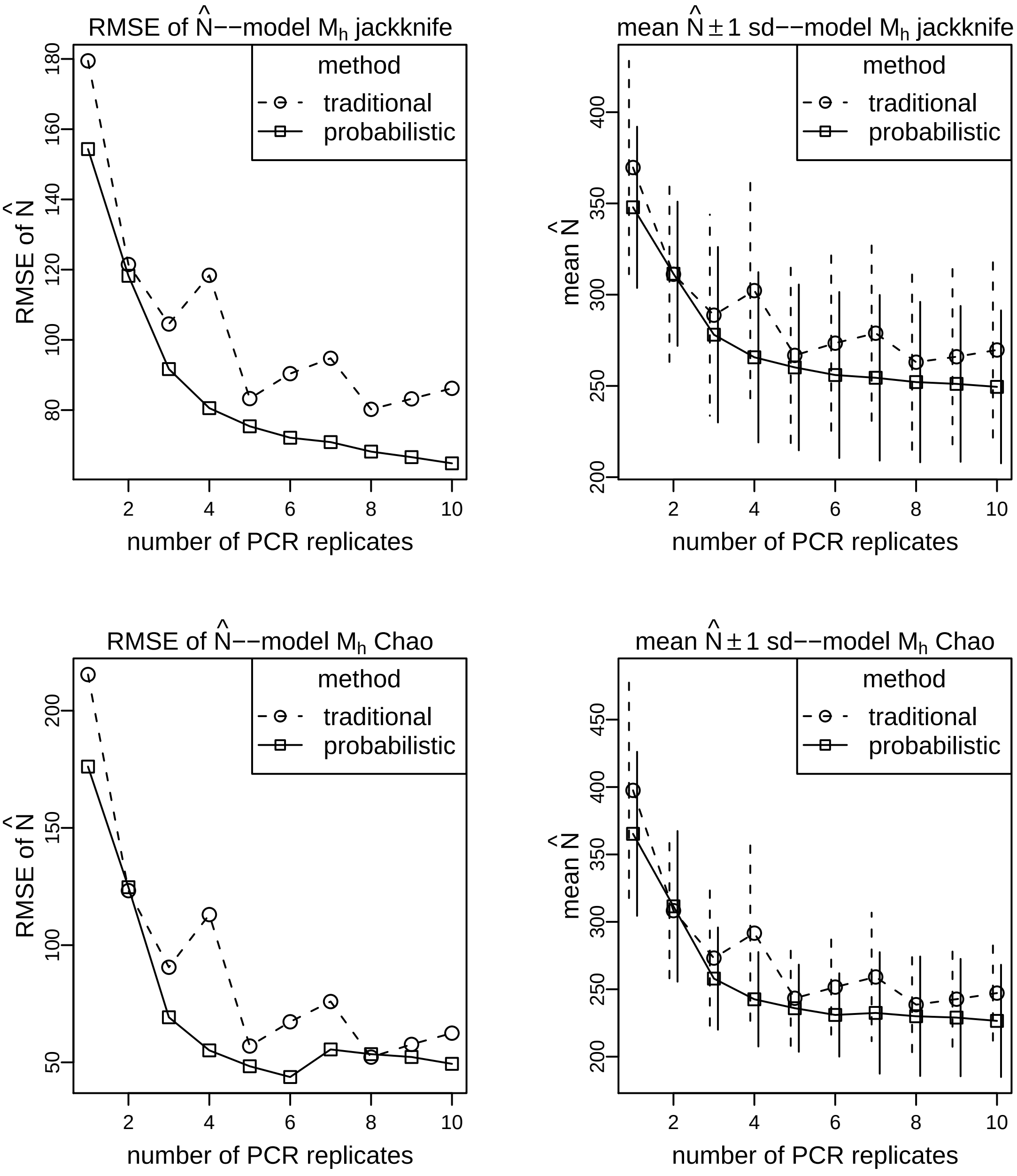


Figure D.8: Results for simulation 4: N=200, capture probability=.2, sampling occasions=10, null alleles present. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_h jackknife and model M_h Chao for two methods—the traditional consensus method ("traditional") and the probabilistic method ("probabilistic").

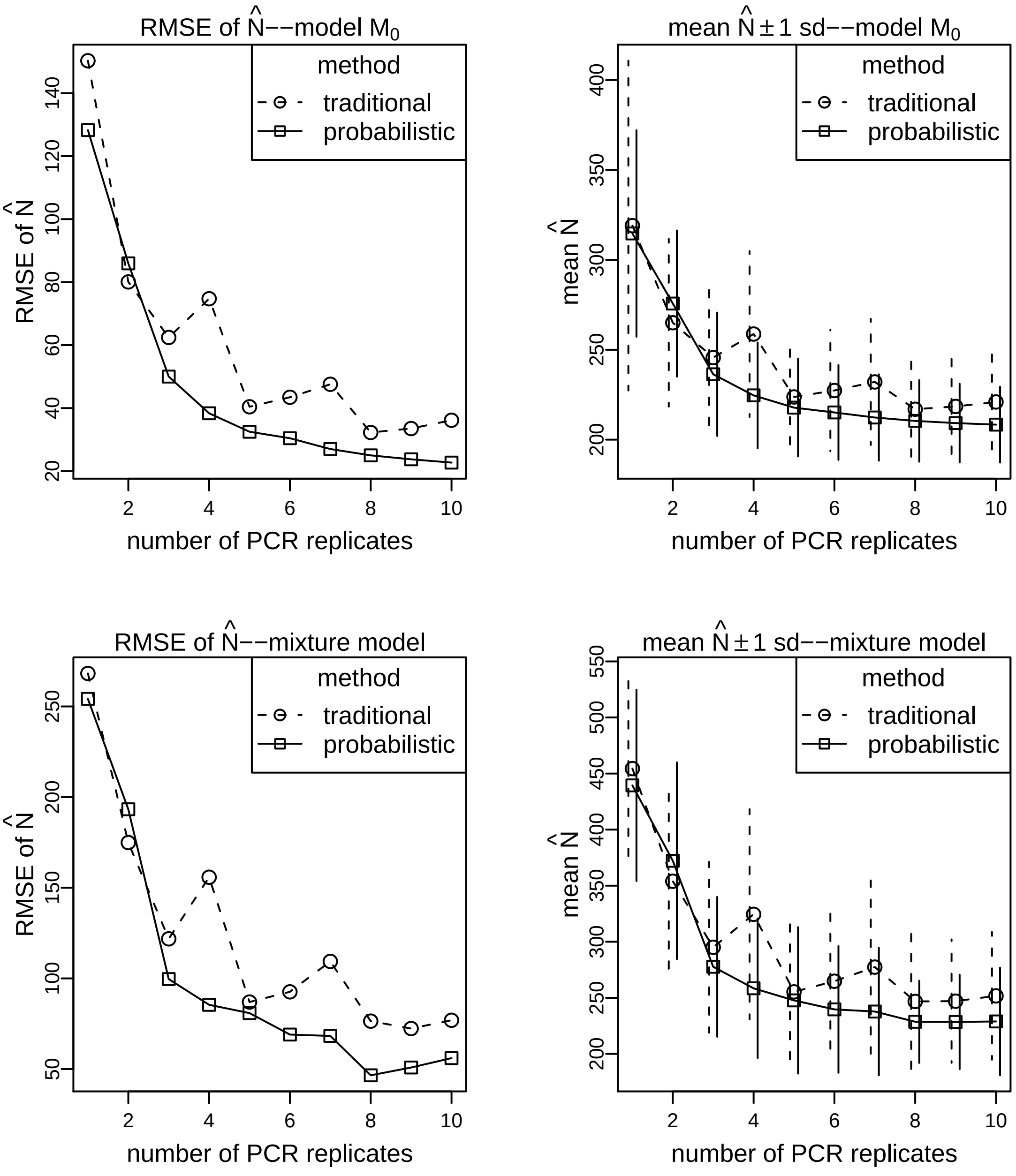


Figure D.9: Results for simulation 5: N=200, capture probability=.5, sampling occasions=5, no null alleles. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_0 and the two point mixture model for two methods—the traditional consensus method ("traditional") and the probabilistic method ("probabilistic").

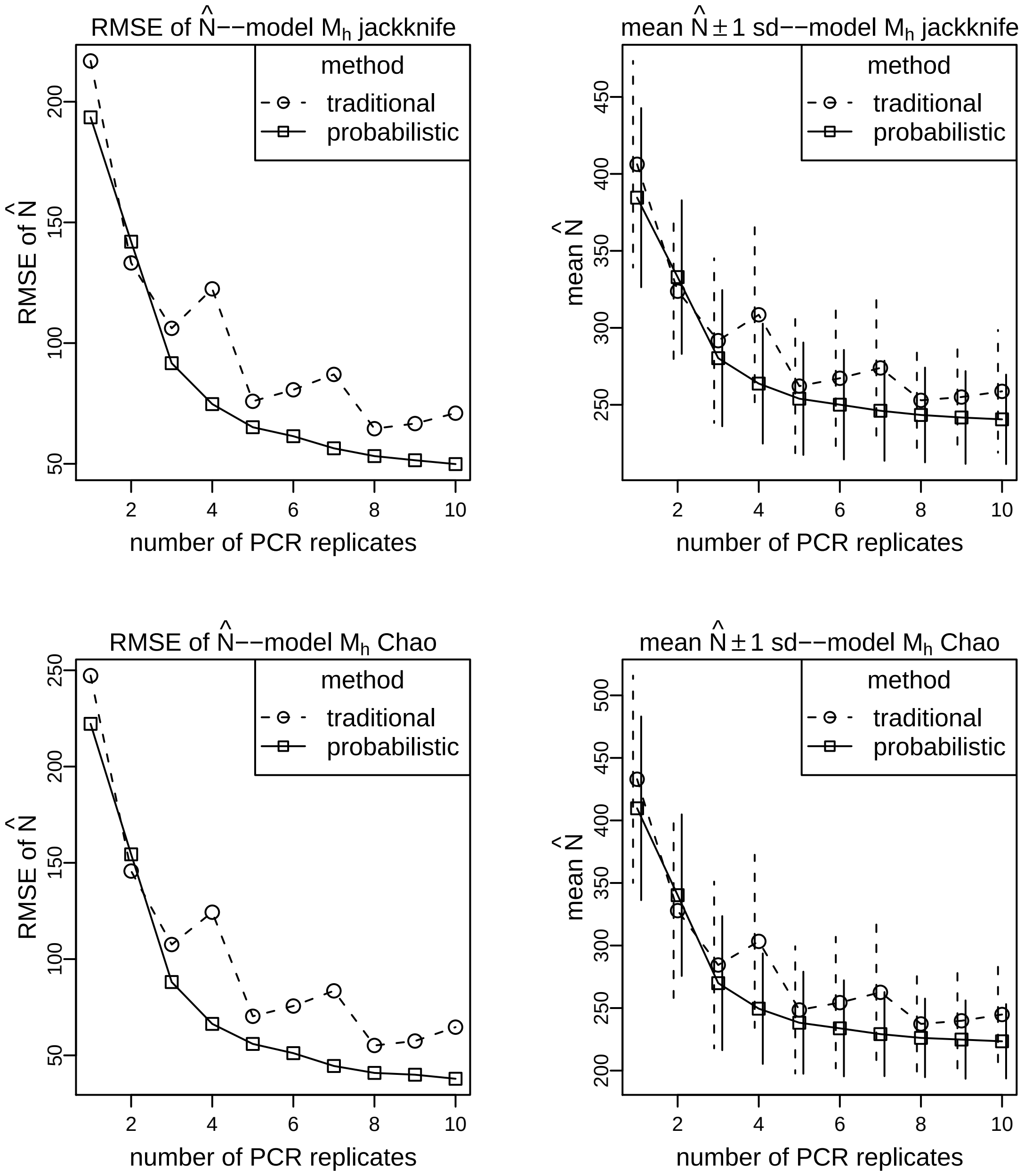


Figure D.10: Results for simulation 5: N=200, capture probability=.5, sampling occasions=5, no null alleles. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_h jackknife and model M_h Chao for two methods—the traditional consensus method (“traditional”) and the probabilistic method (“probabilistic”).

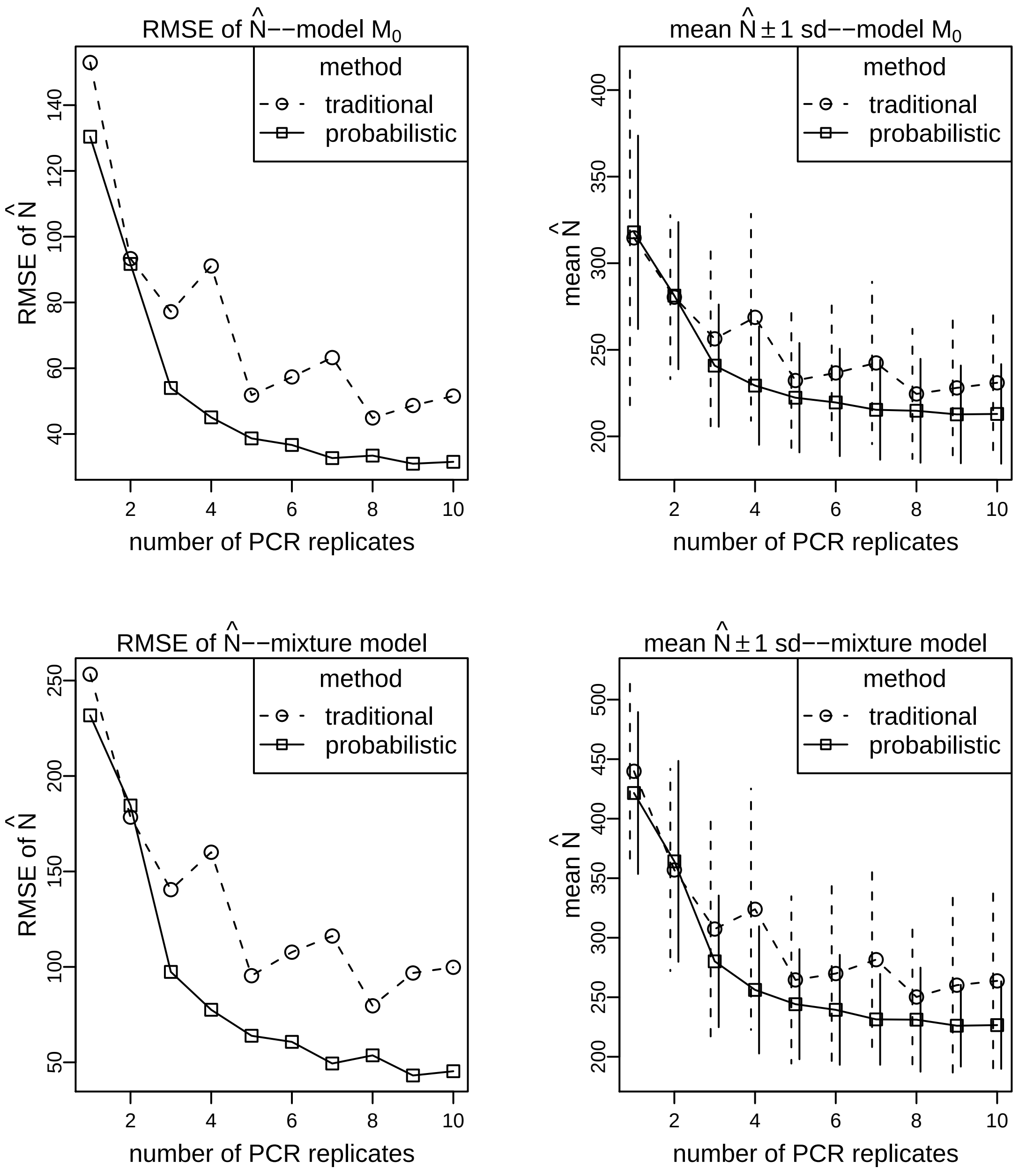


Figure D.11: Results for simulation 6: N=200, capture probability=.5, sampling occasions=5, null alleles present. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_0 and the two point mixture model for two methods—the traditional consensus method ("traditional") and the probabilistic method ("probabilistic").

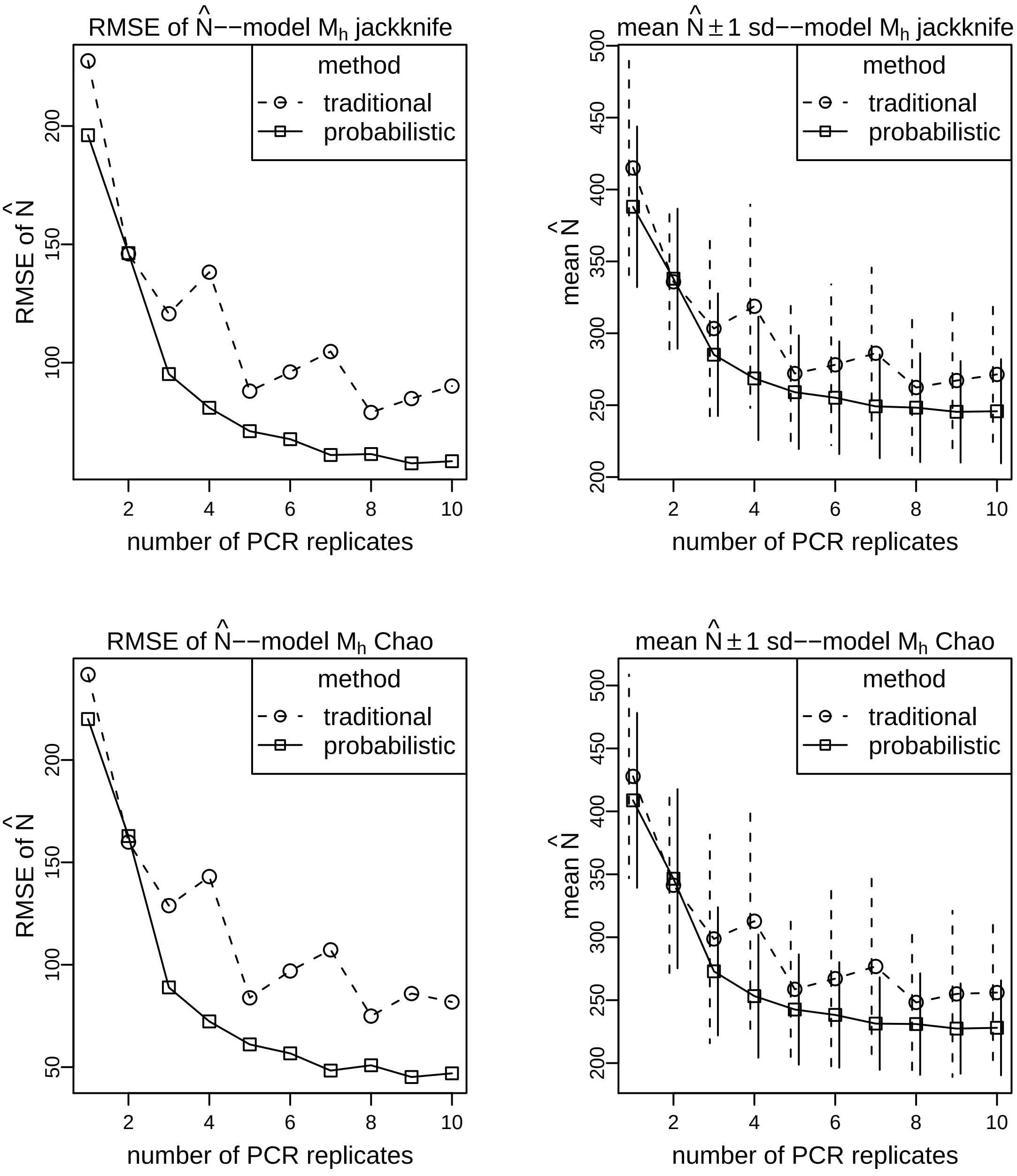


Figure D.12: Results for simulation 6: N=200, capture probability=.5, sampling occasions=5, null alleles present. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_h jackknife and model M_h Chao for two methods—the traditional consensus method ("traditional") and the probabilistic method ("probabilistic").

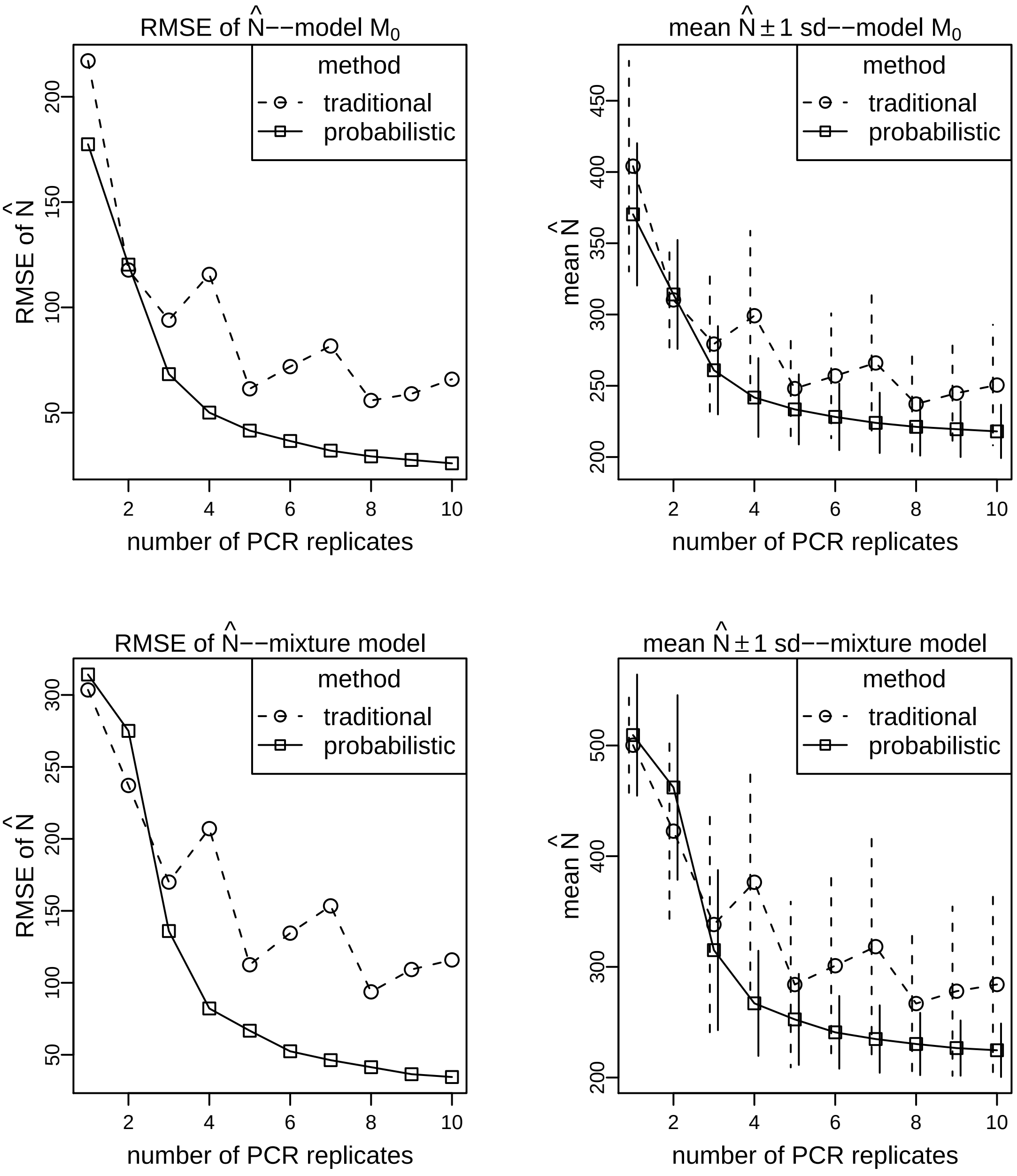


Figure D.13: Results for simulation 7: N=200, capture probability=.5, sampling occasions=10, no null alleles. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_0 and the two point mixture model for two methods—the traditional consensus method ("traditional") and the probabilistic method ("probabilistic").

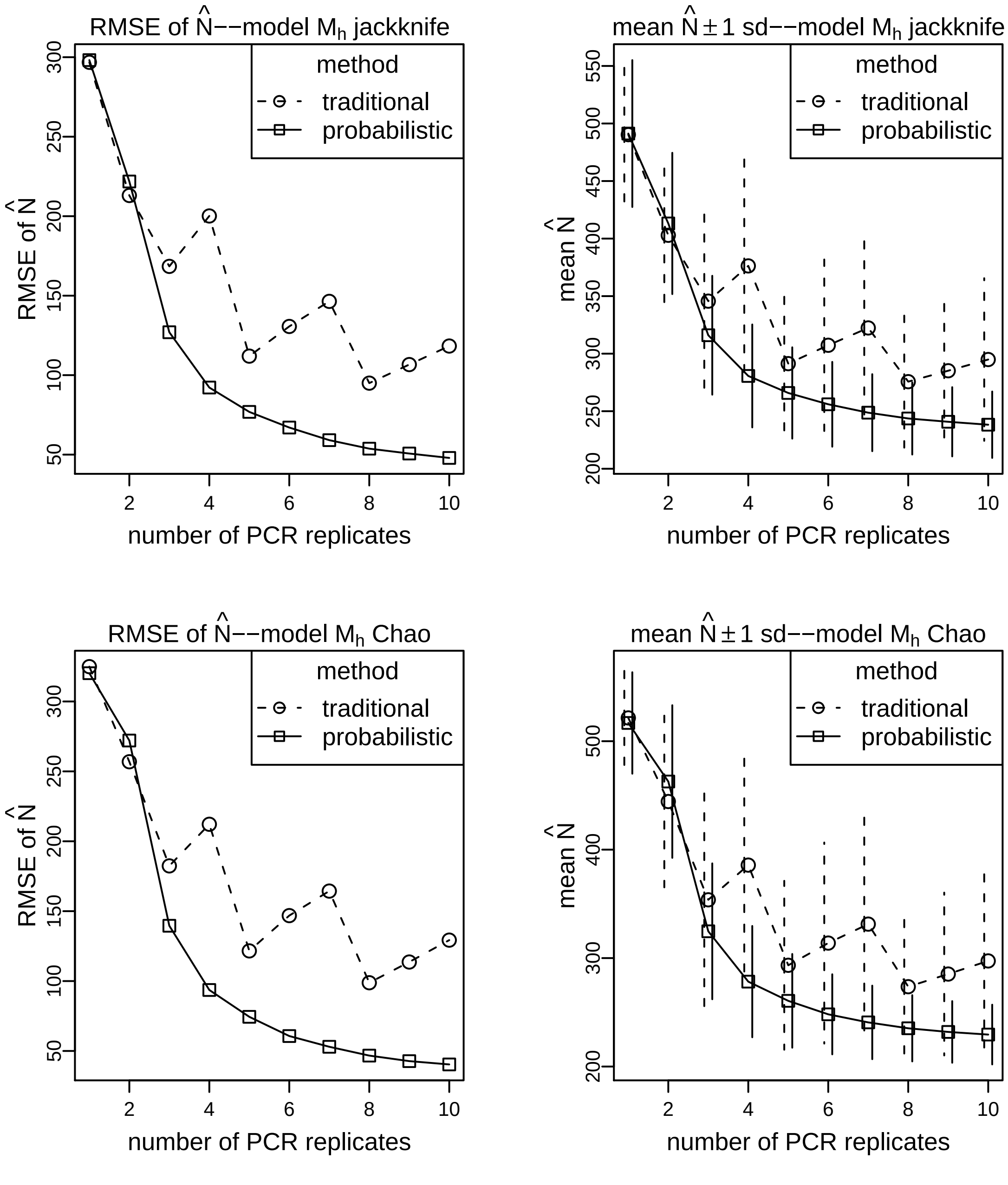


Figure D.14: Results for simulation 7: N=200, capture probability=.5, sampling occasions=10, no null alleles. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_h jackknife and model M_h Chao for two methods—the traditional consensus method ("traditional") and the probabilistic method ("probabilistic").

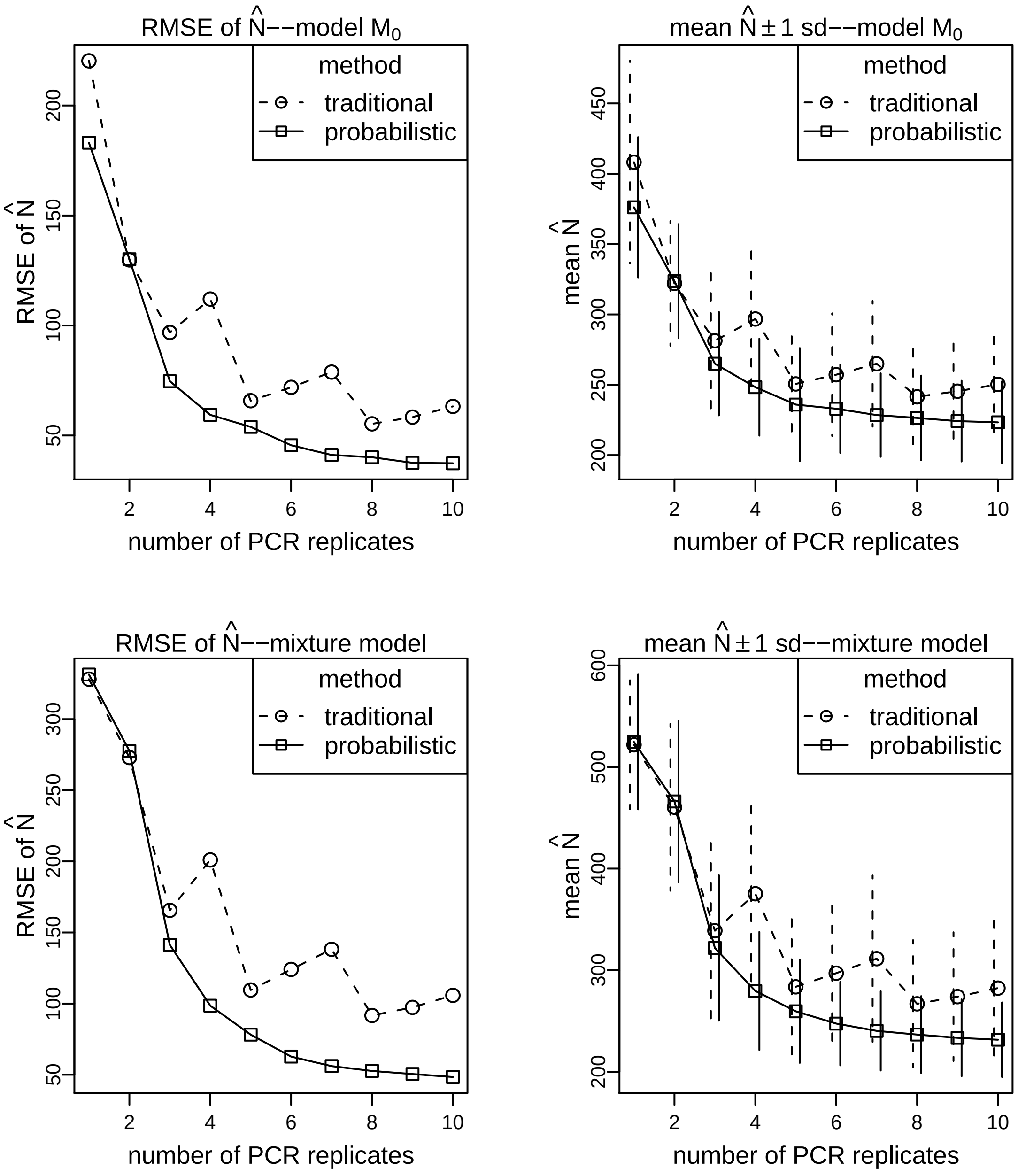


Figure D.15: Results for simulation 8: N=200, capture probability=.5, sampling occasions=10, null alleles present. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_0 and the two point mixture model for two methods—the traditional consensus method ("traditional") and the probabilistic method ("probabilistic").

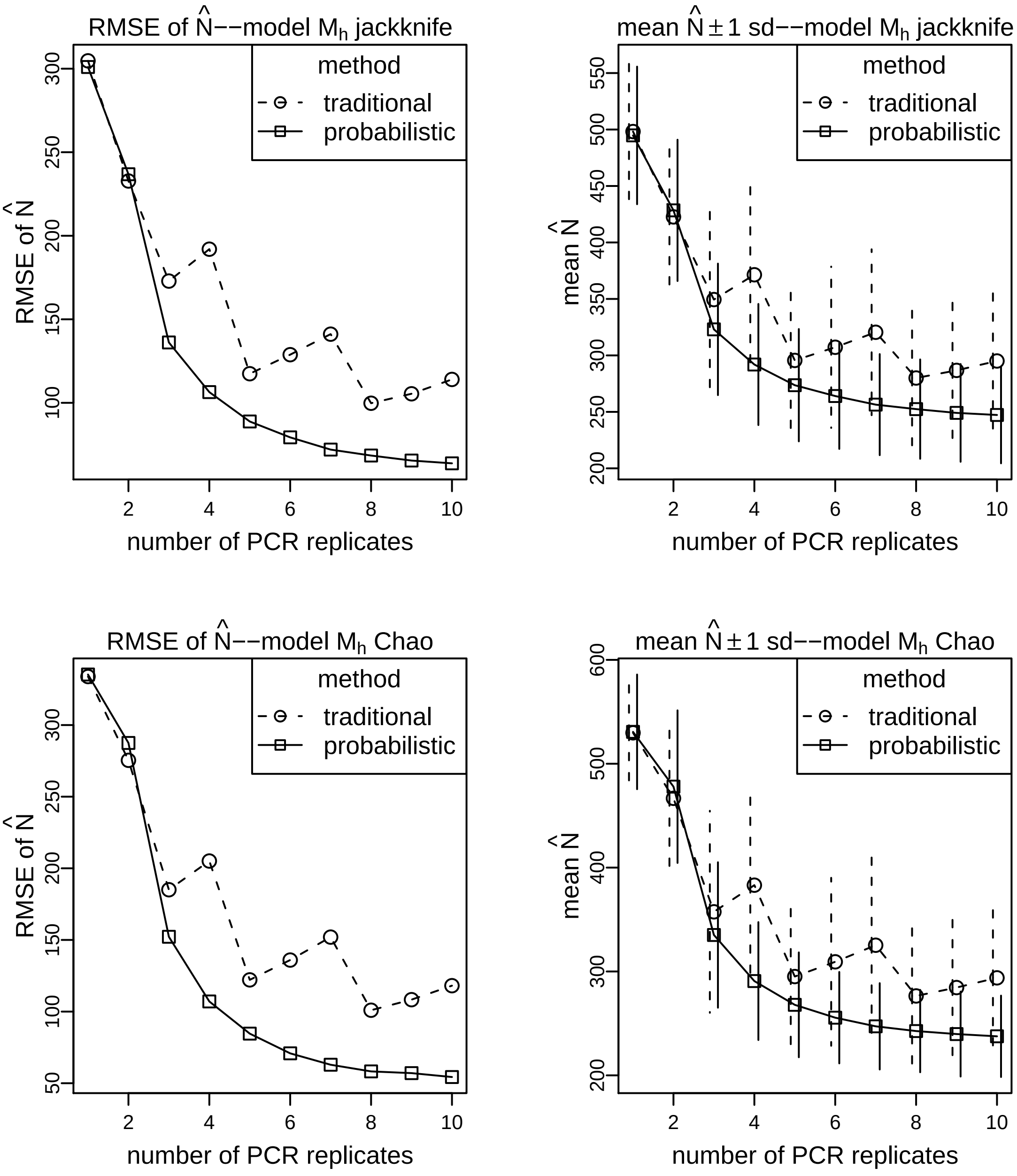


Figure D.16: Results for simulation 8: N=200, capture probability=.5, sampling occasions=10, null alleles present. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_h jackknife and model M_h Chao for two methods—the traditional consensus method ("traditional") and the probabilistic method ("probabilistic").

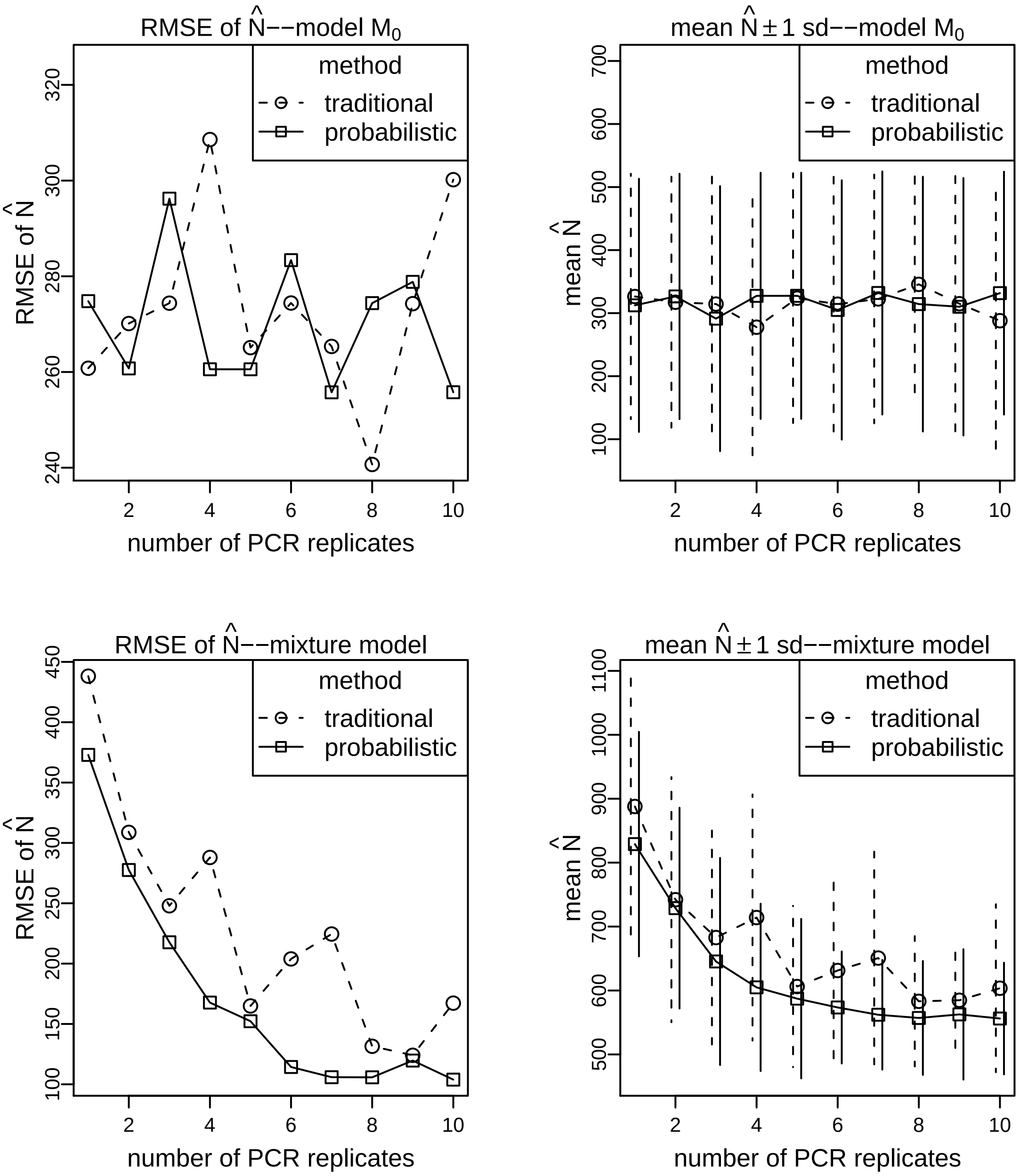


Figure D.17: Results for simulation 9: N=500, capture probability=.2, sampling occasions=5, no null alleles. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_0 and the two point mixture model for two methods—the traditional consensus method ("traditional") and the probabilistic method ("probabilistic").

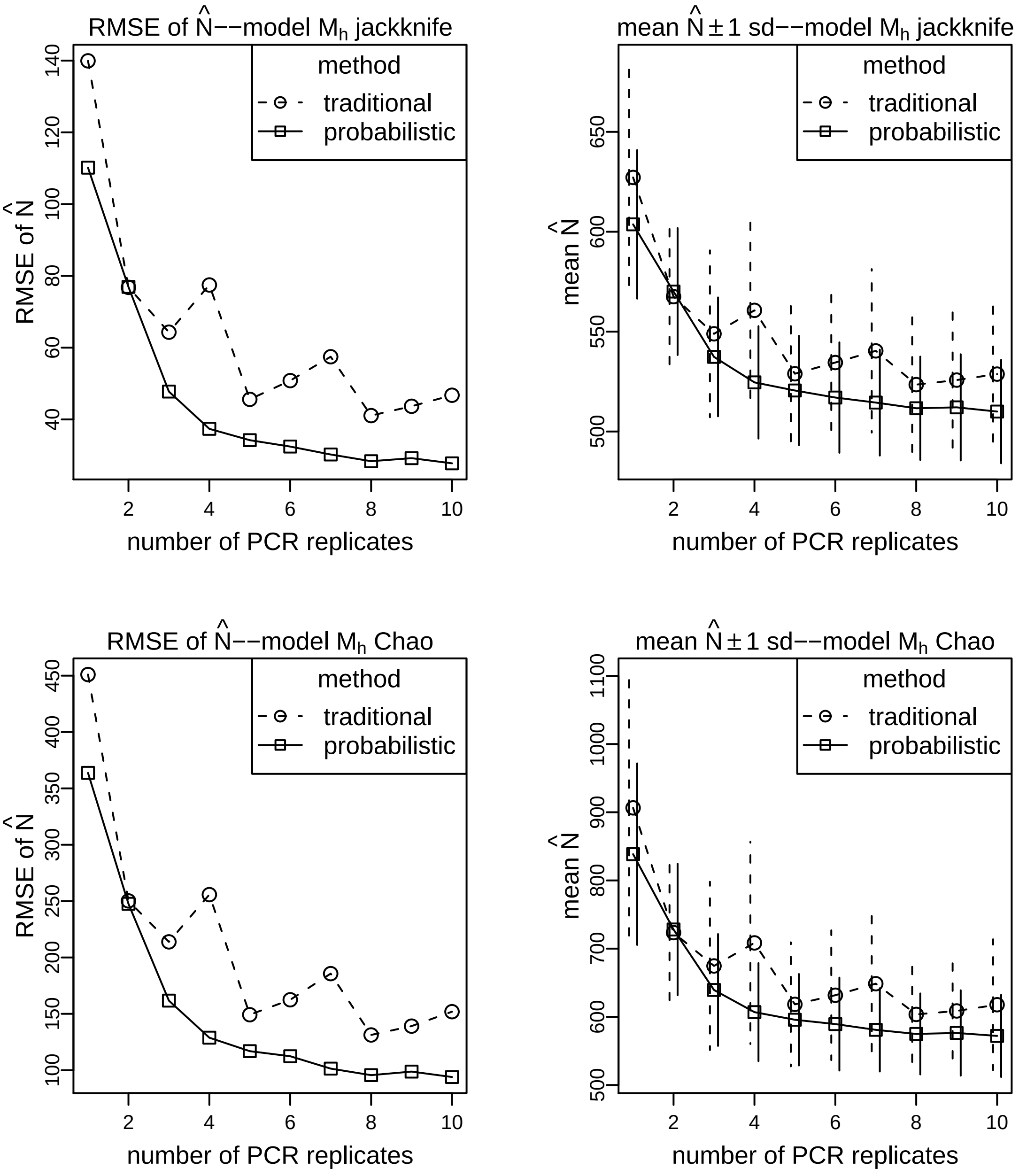


Figure D.18: Results for simulation 9: N=500, capture probability=.2, sampling occasions=5, no null alleles. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_h jackknife and model M_h Chao for two methods—the traditional consensus method ("traditional") and the probabilistic method ("probabilistic").

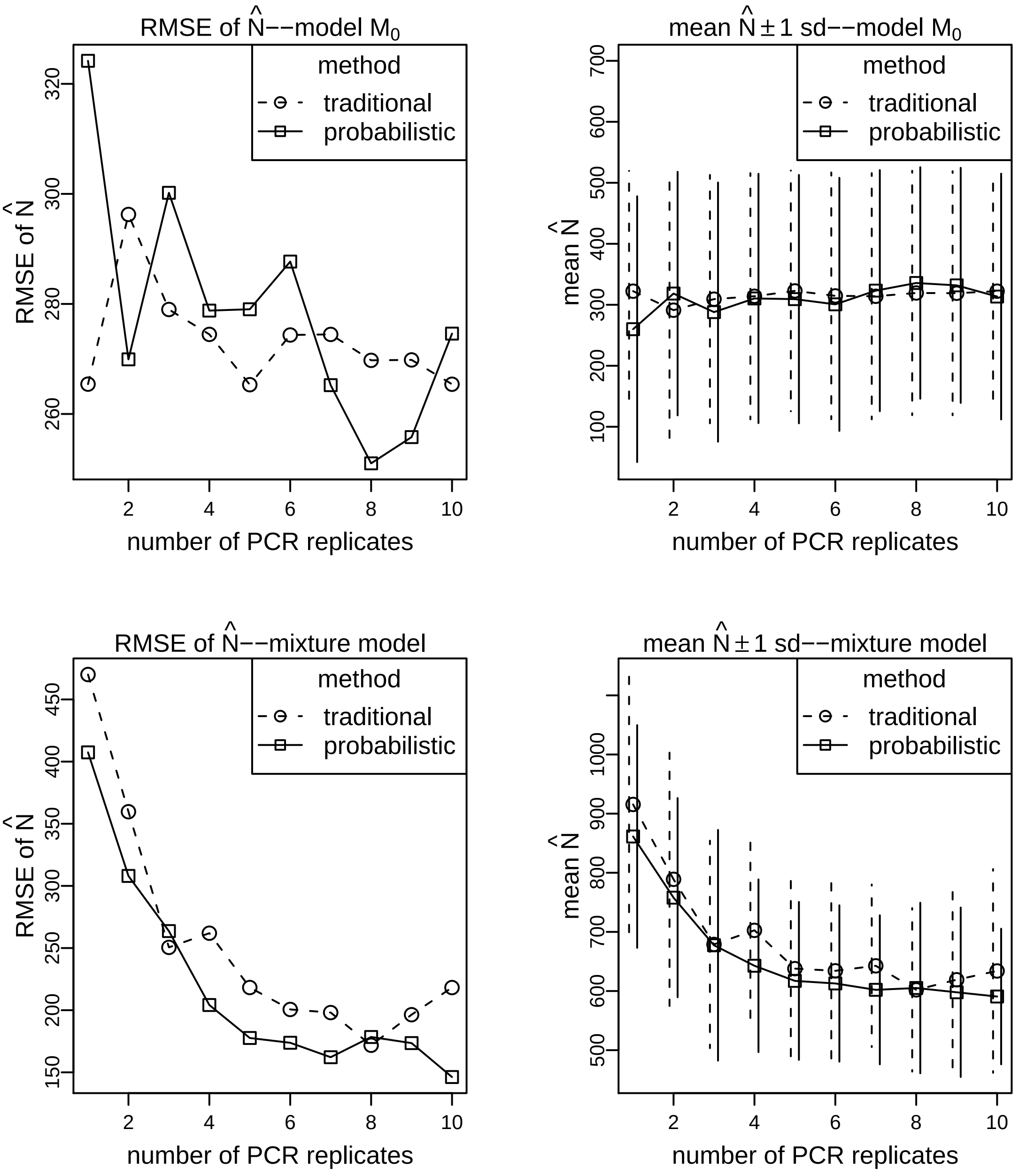


Figure D.19: Results for simulation 10: N=500, capture probability=.2, sampling occasions=5, null alleles present. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_0 and the two point mixture model for two methods—the traditional consensus method ("traditional") and the probabilistic method ("probabilistic").

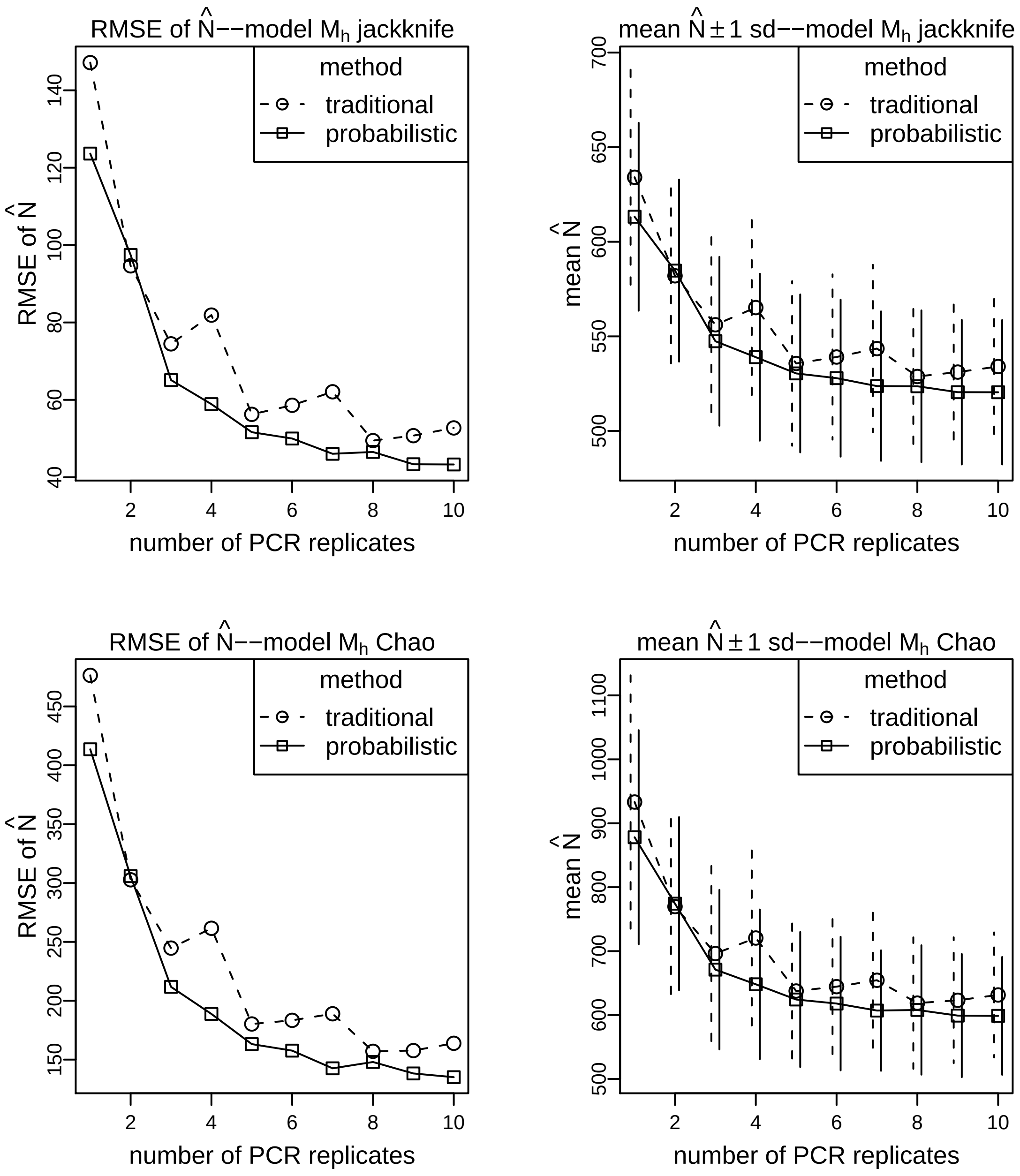


Figure D.20: Results for simulation 10: N=500, capture probability=.2, sampling occasions=5, null alleles present. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_h jackknife and model M_h Chao for two methods—the traditional consensus method (“traditional”) and the probabilistic method (“probabilistic”).

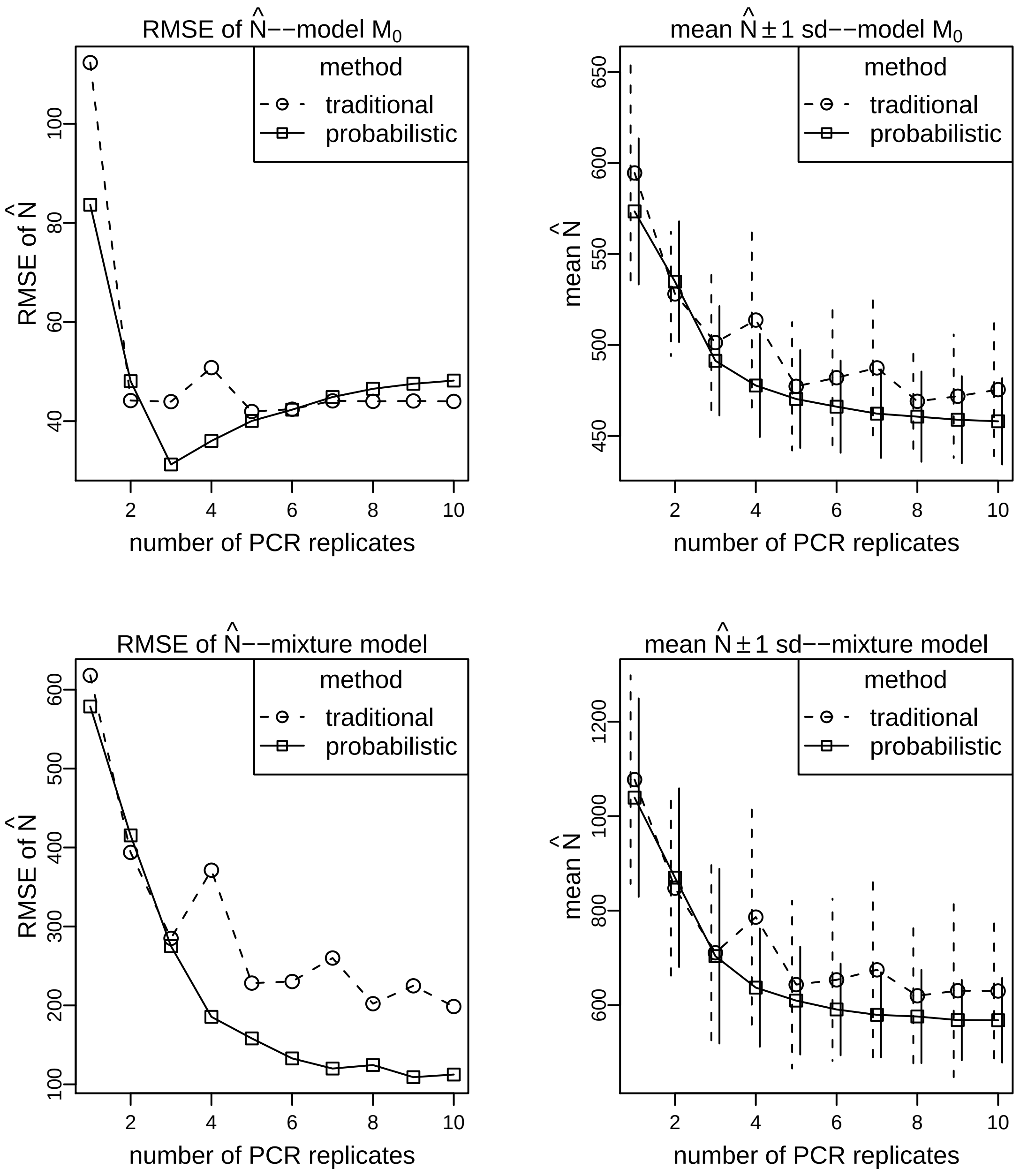


Figure D.21: Results for simulation 11: N=500, capture probability=.2, sampling occasions=10, no null alleles. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_0 and the two point mixture model for two methods—the traditional consensus method (“traditional”) and the probabilistic method (“probabilistic”).

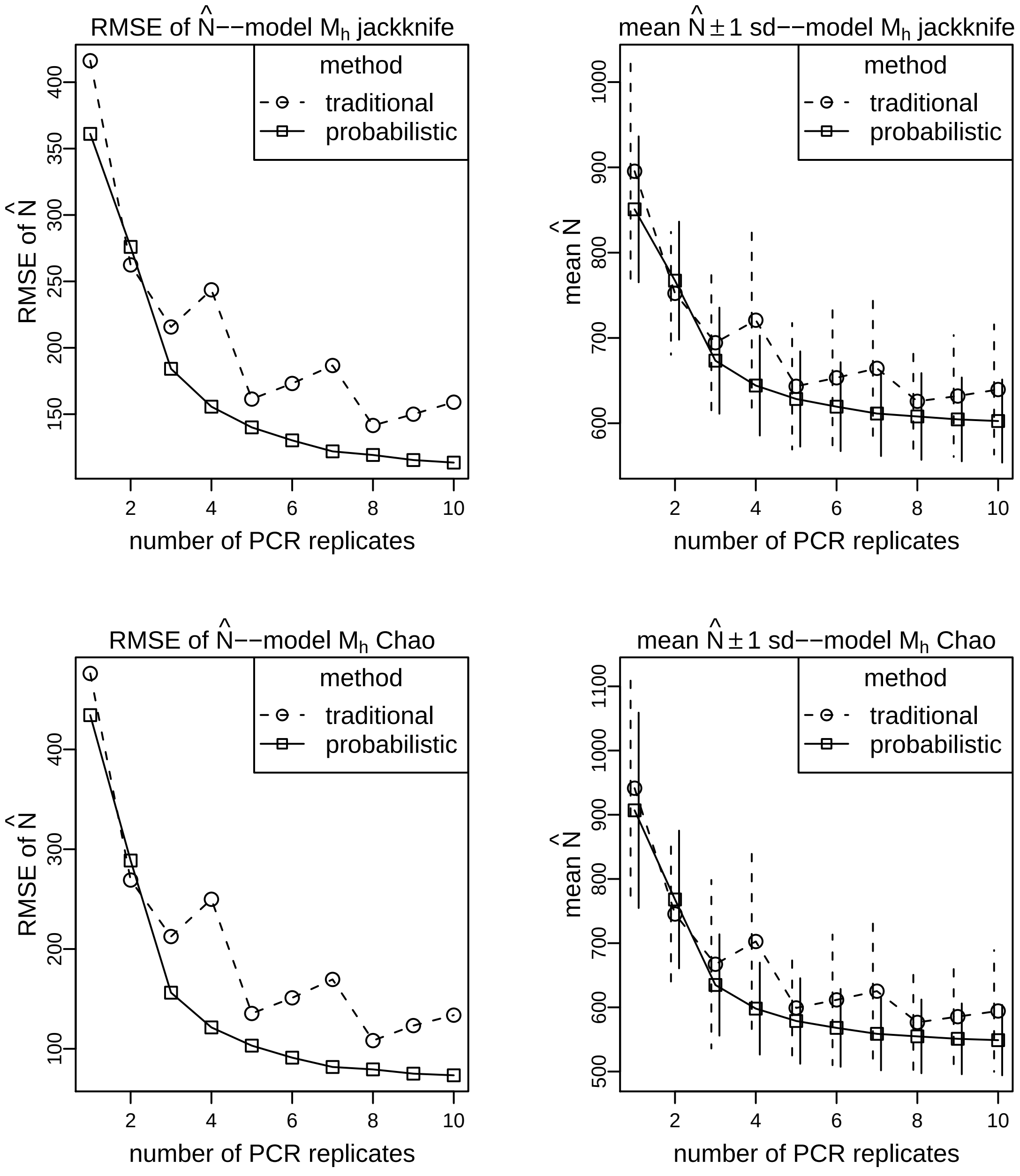


Figure D.22: Results for simulation 11: N=500, capture probability=.2, sampling occasions=10, no null alleles. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_h jackknife and model M_h Chao for two methods—the traditional consensus method ("traditional") and the probabilistic method ("probabilistic").

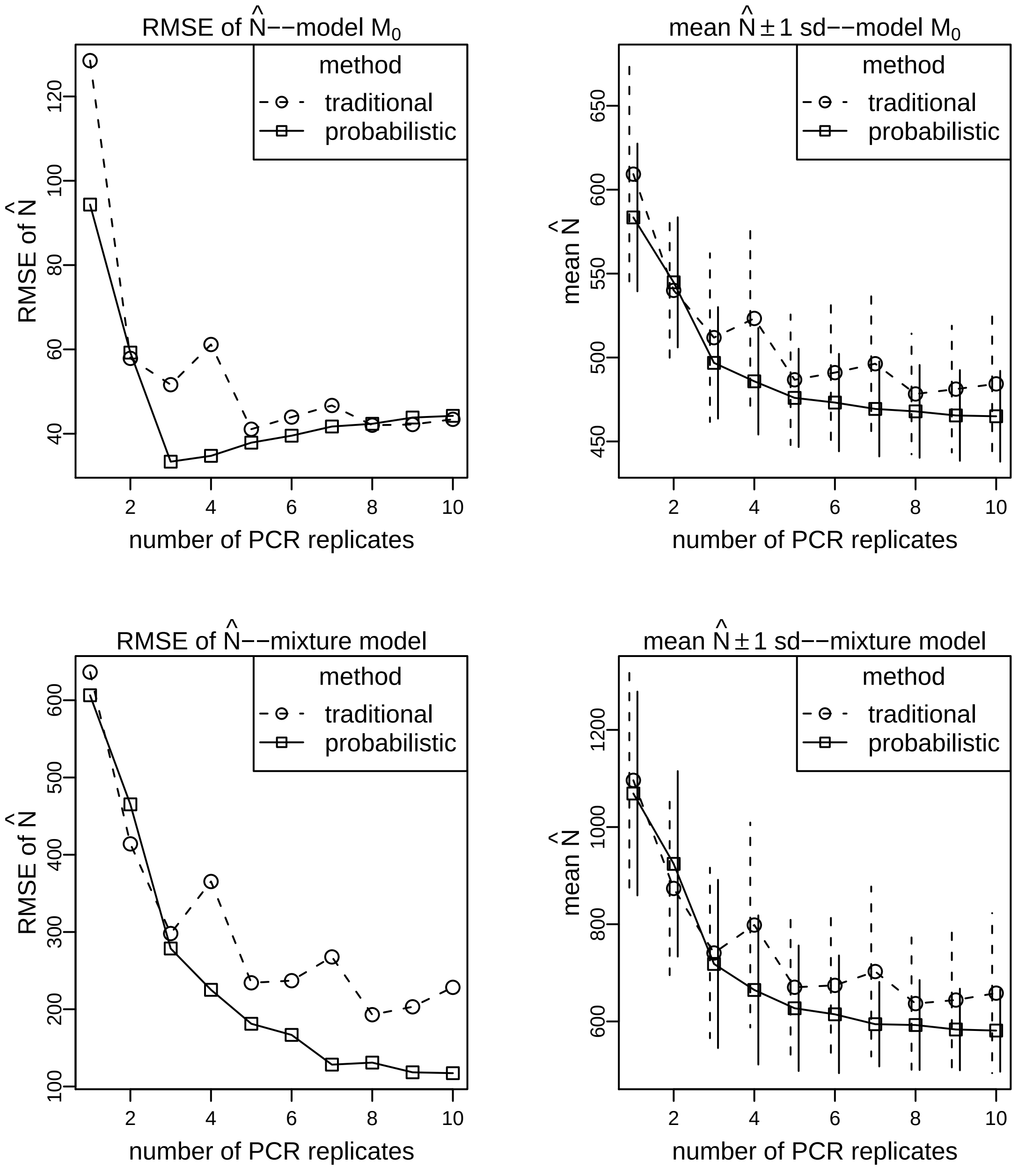


Figure D.23: Results for simulation 12: N=500, capture probability=.2, sampling occasions=10, null alleles present. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_0 and the two point mixture model for two methods—the traditional consensus method ("traditional") and the probabilistic method ("probabilistic").

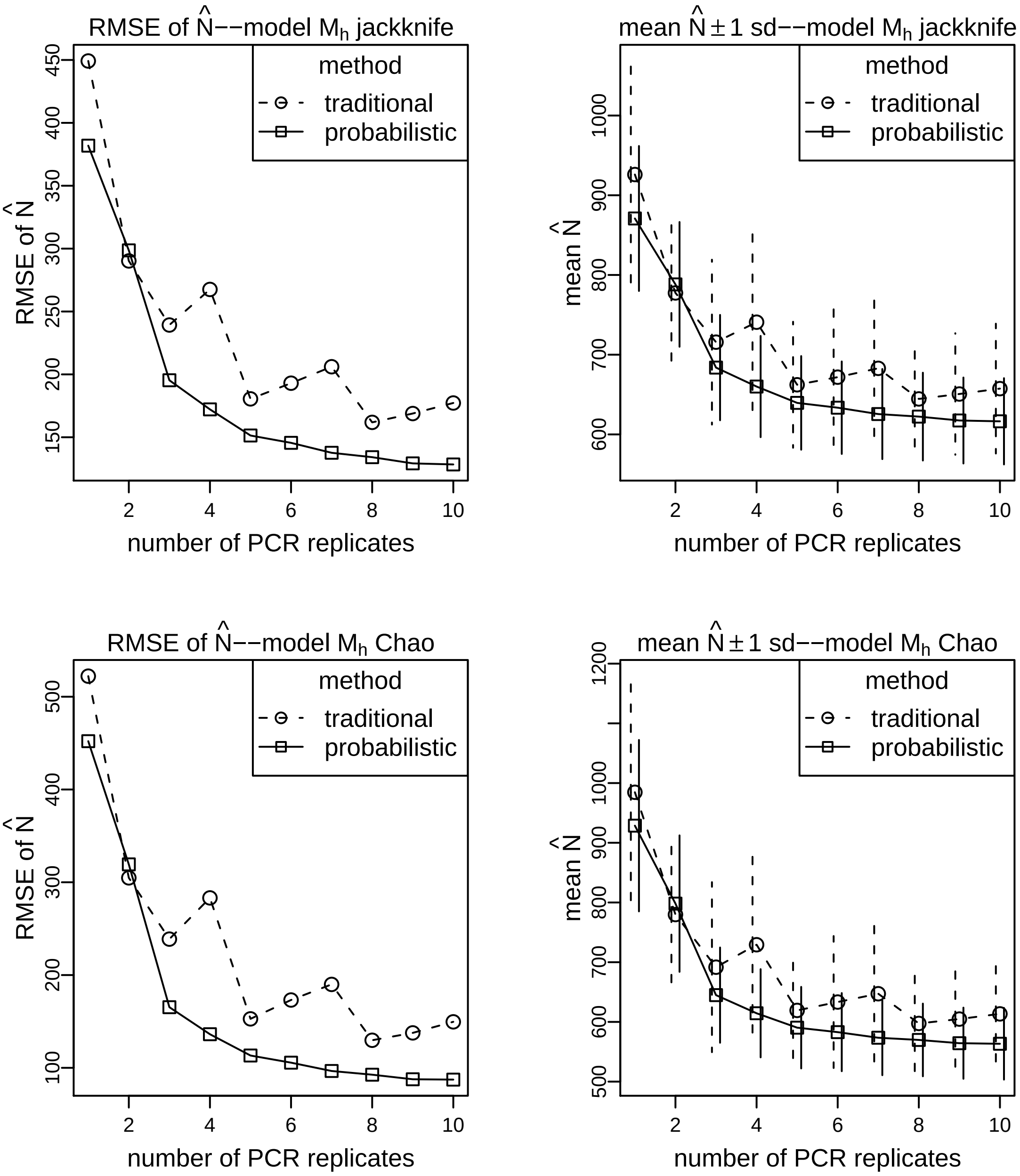


Figure D.24: Results for simulation 12: N=500, capture probability=.2, sampling occasions=10, null alleles present. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_h jackknife and model M_h Chao for two methods—the traditional consensus method ("traditional") and the probabilistic method ("probabilistic").

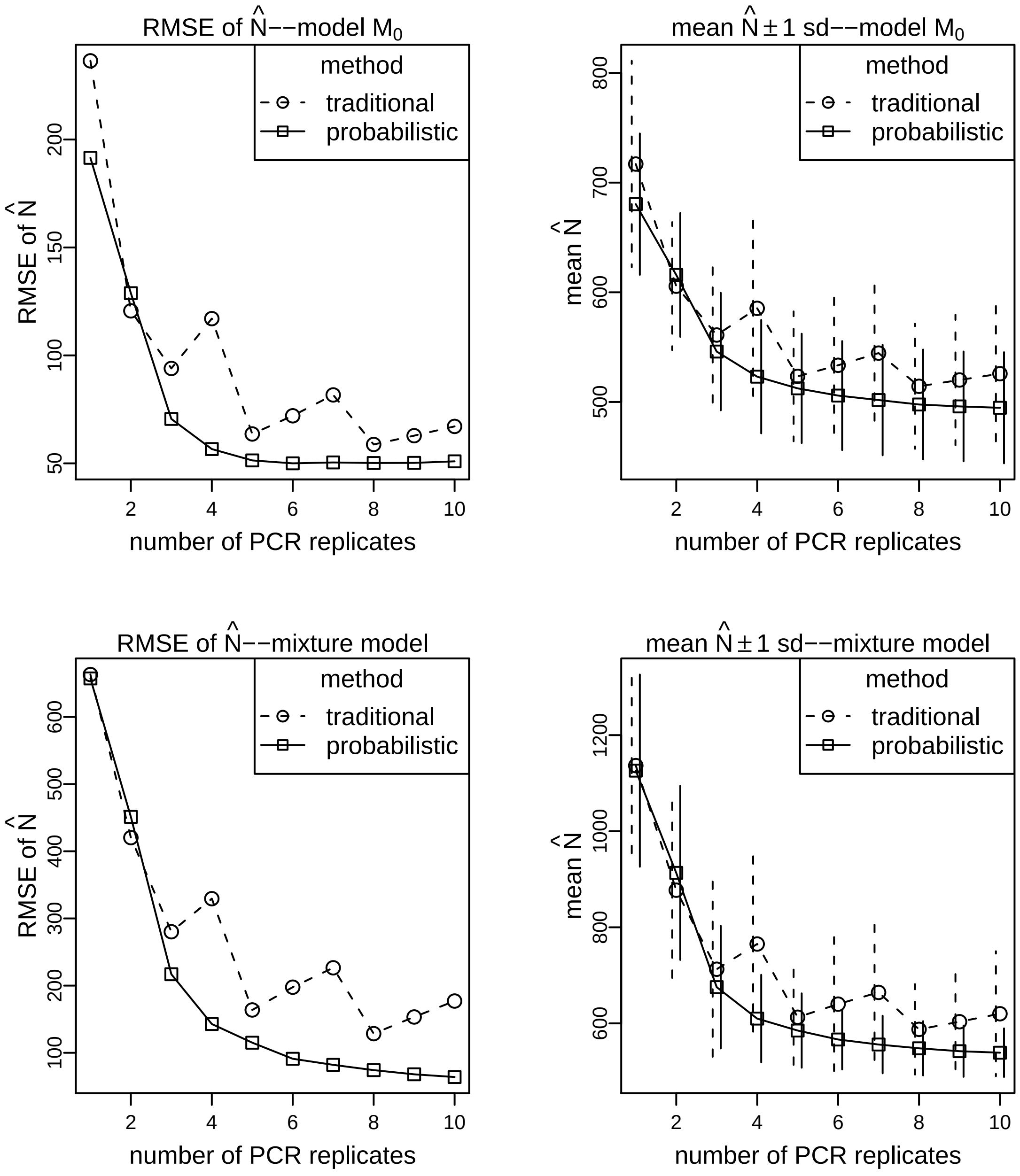


Figure D.25: Results for simulation 13: N=500, capture probability=.5, sampling occasions=5, no null alleles. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_0 and the two point mixture model for two methods—the traditional consensus method (“traditional”) and the probabilistic method (“probabilistic”).

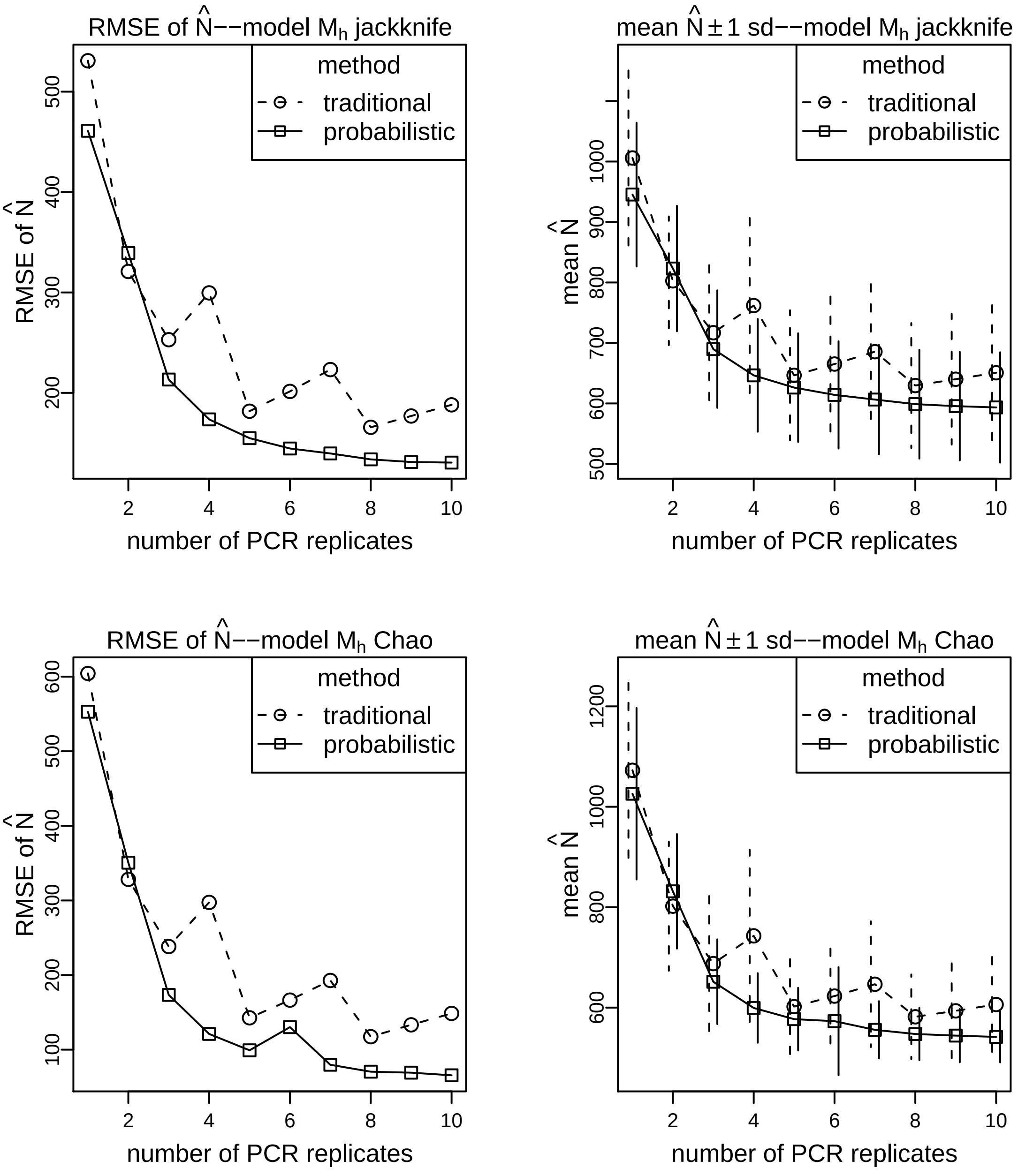


Figure D.26: Results for simulation 13: N=500, capture probability=.5, sampling occasions=5, no null alleles. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_h jackknife and model M_h Chao for two methods—the traditional consensus method (“traditional”) and the probabilistic method (“probabilistic”).

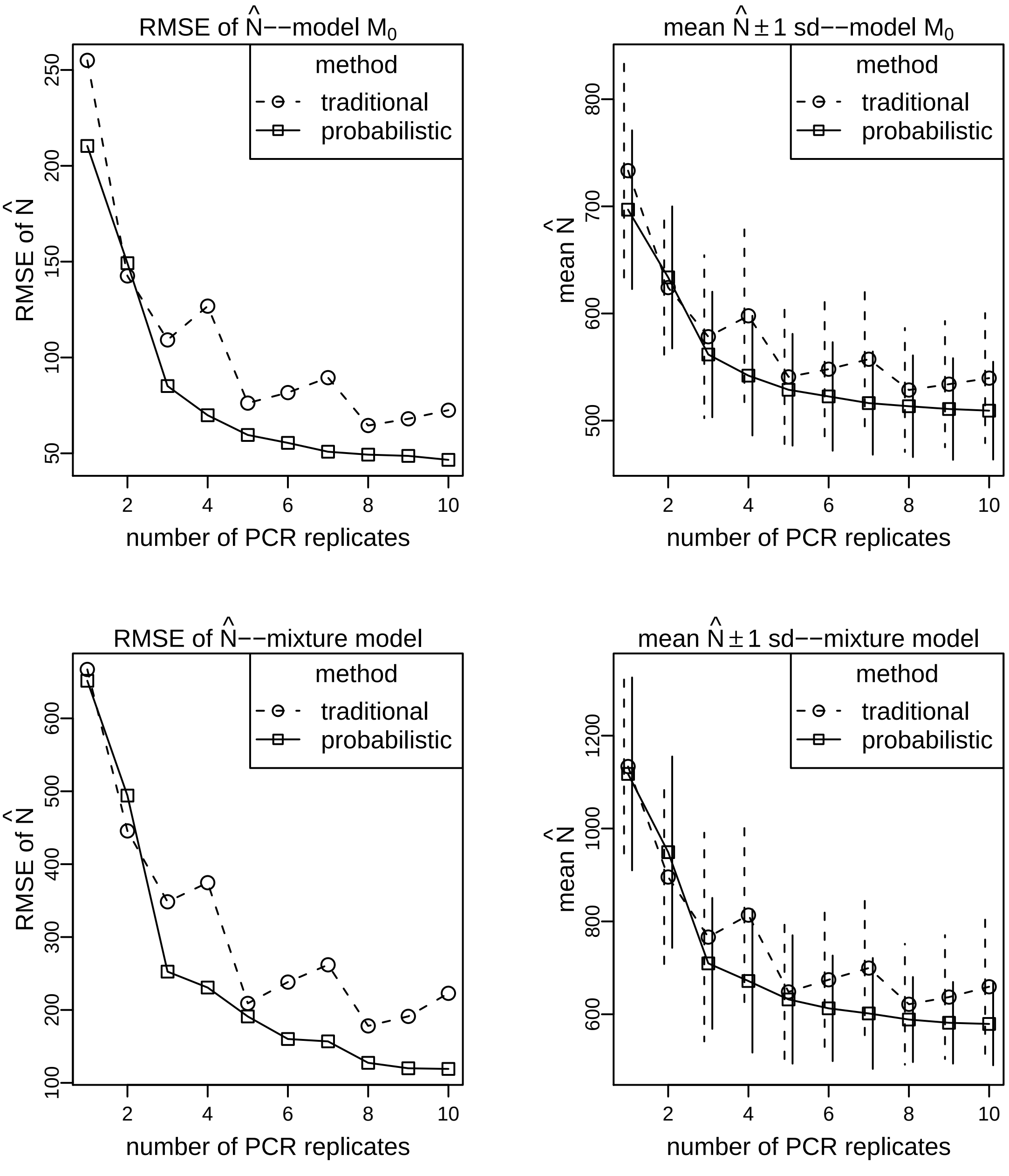


Figure D.27: Results for simulation 14: N=500, capture probability=.5, sampling occasions=5, null alleles present. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_0 and the two point mixture model for two methods—the traditional consensus method ("traditional") and the probabilistic method ("probabilistic").

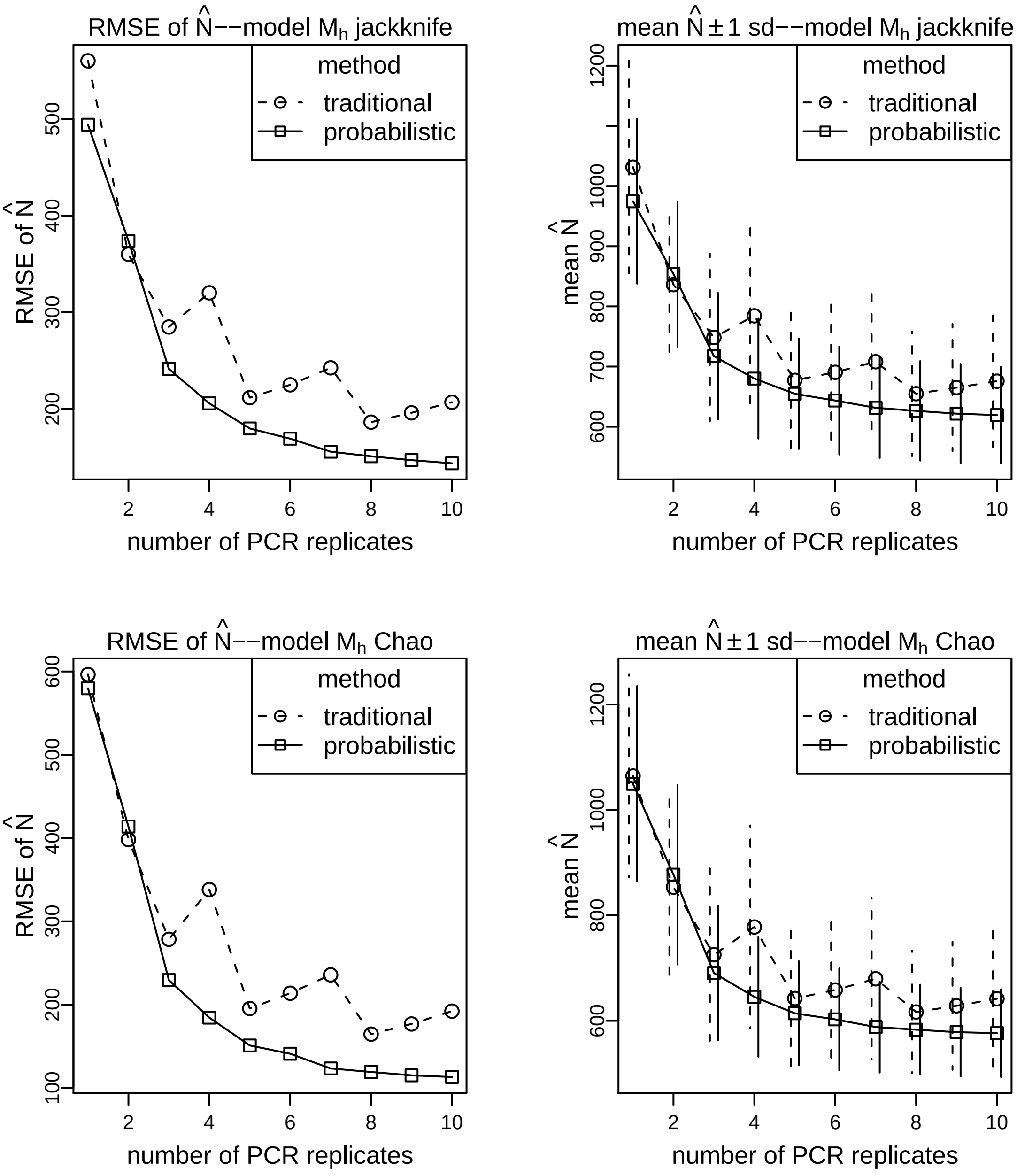


Figure D.28: Results for simulation 14: N=500, capture probability=.5, sampling occasions=5, null alleles present. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_h jackknife and model M_h Chao for two methods—the traditional consensus method ("traditional") and the probabilistic method ("probabilistic").

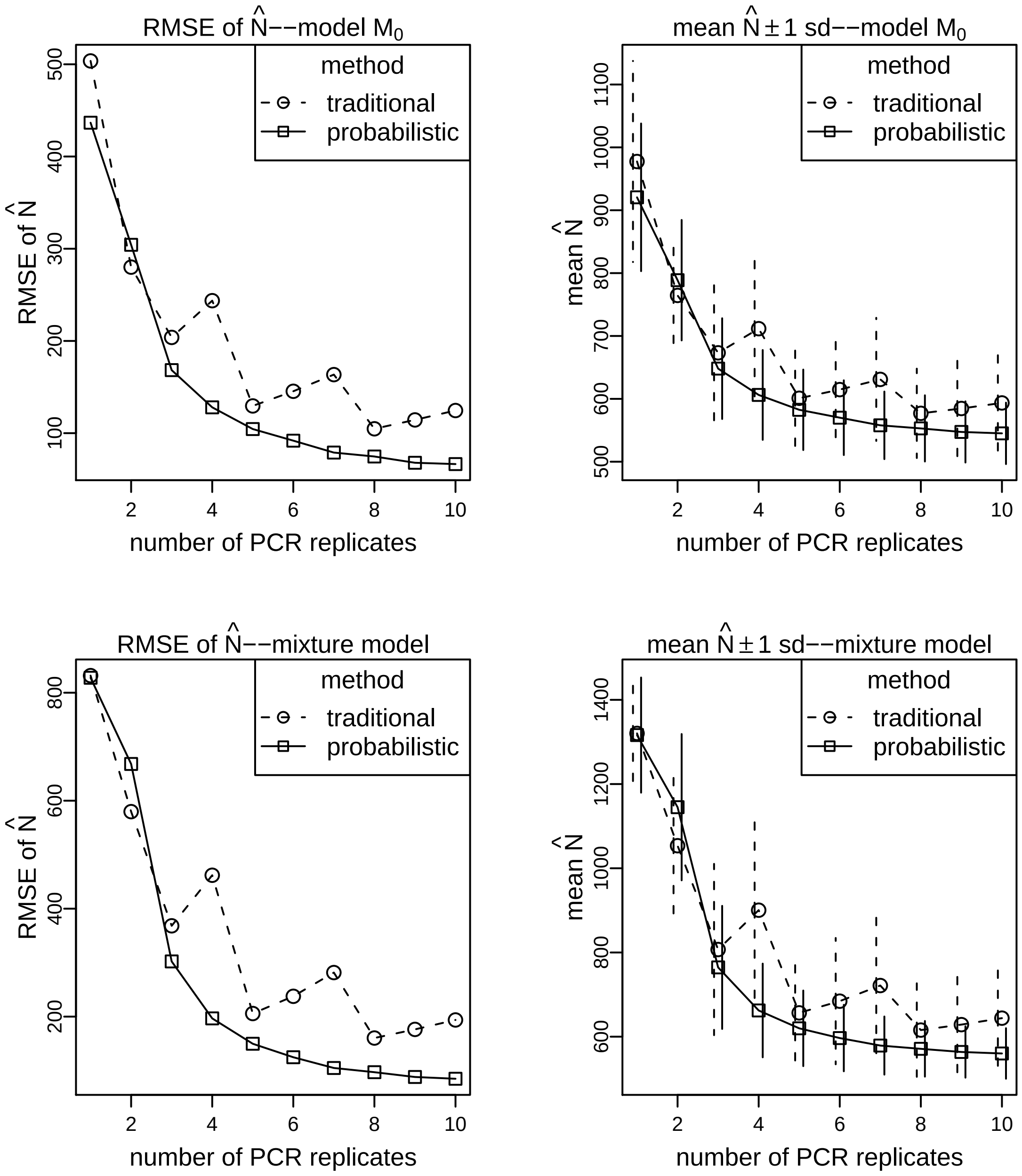


Figure D.29: Results for simulation 15: N=500, capture probability=.5, sampling occasions=10, no null alleles. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_0 and the two point mixture model for two methods—the traditional consensus method ("traditional") and the probabilistic method ("probabilistic").

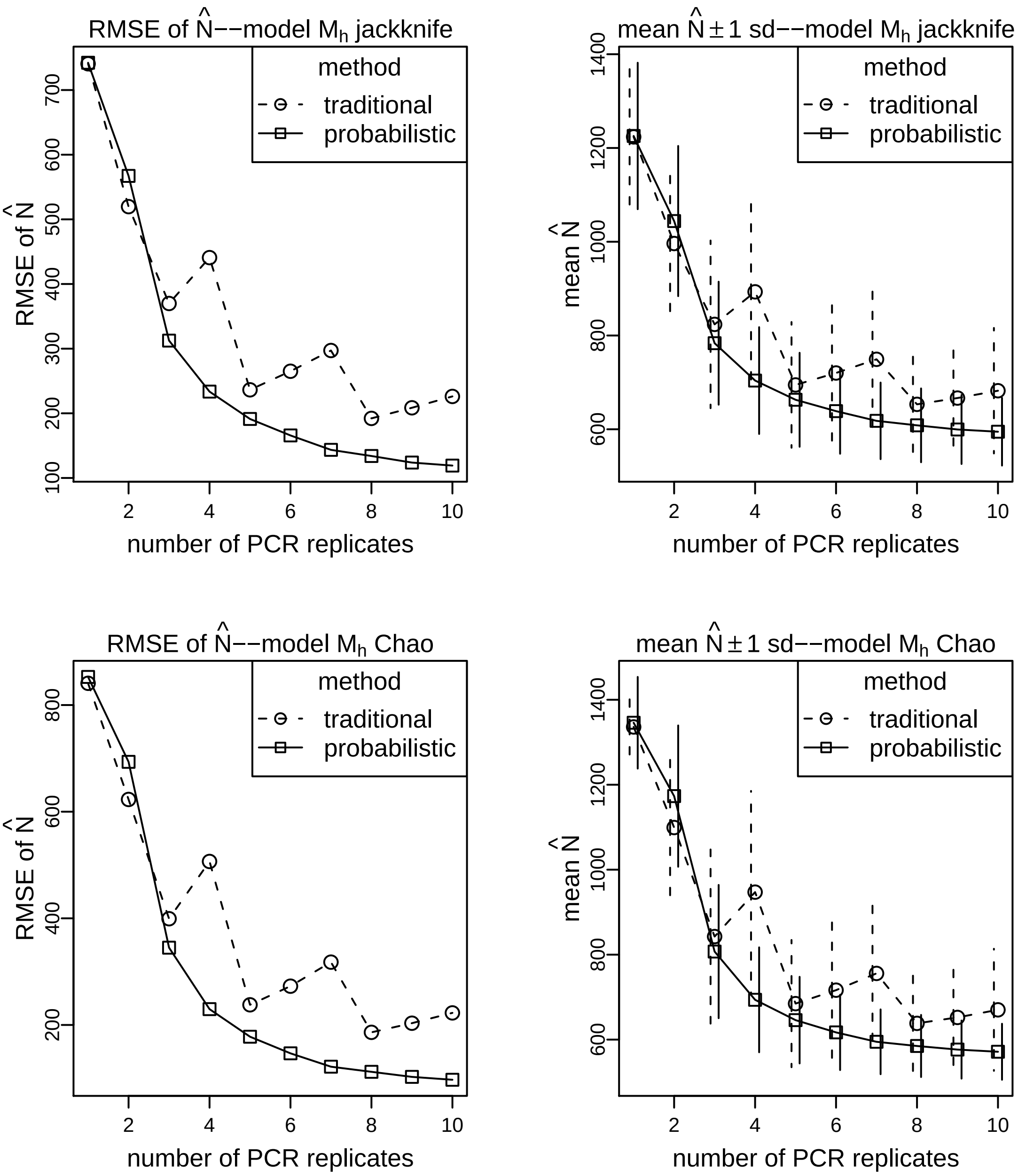


Figure D.30: Results for simulation 15: N=500, capture probability=.5, sampling occasions=10, no null alleles. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_h jackknife and model M_h Chao for two methods—the traditional consensus method (“traditional”) and the probabilistic method (“probabilistic”).

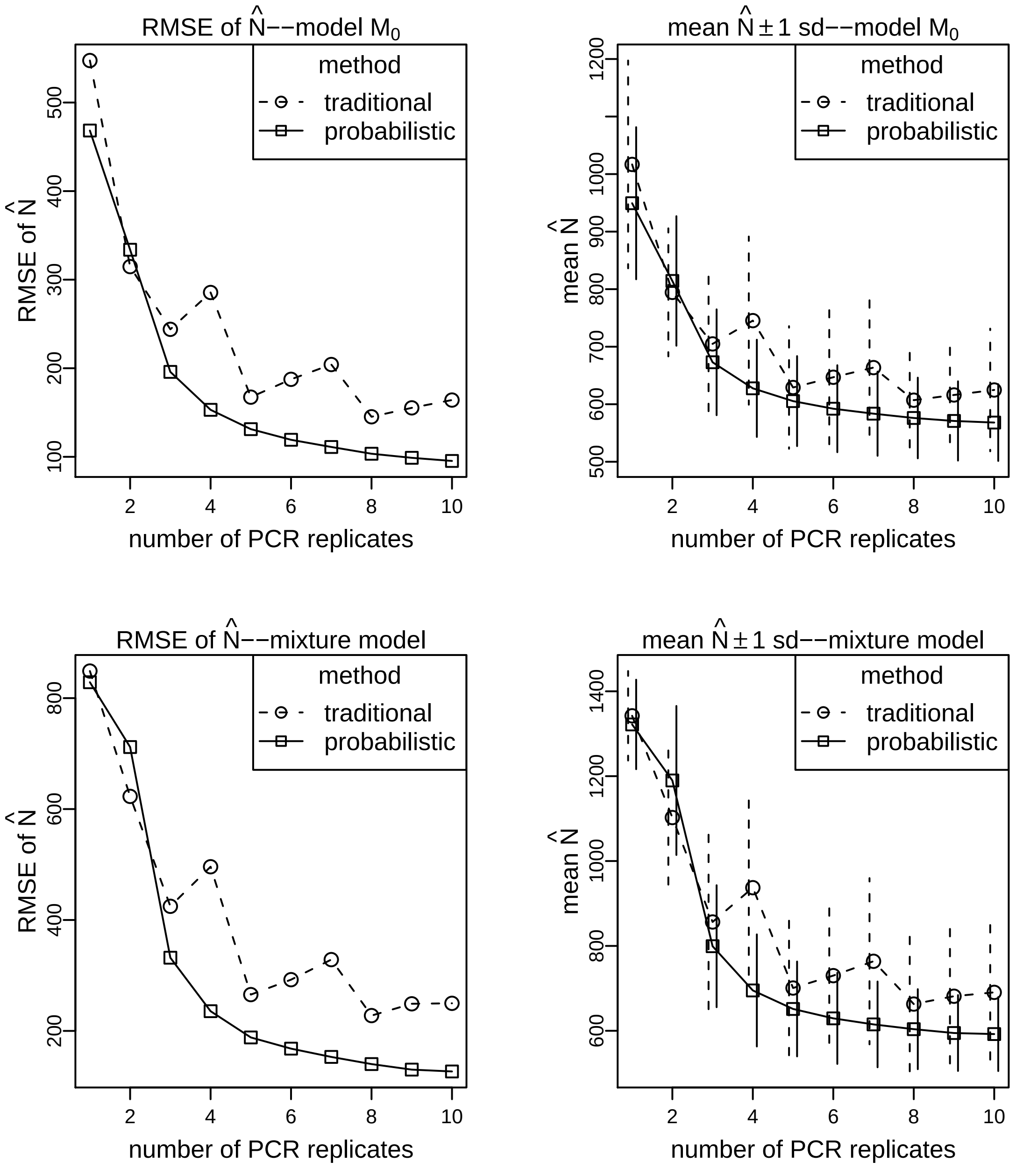


Figure D.31: Results for simulation 16: N=500, capture probability=.5, sampling occasions=10, null alleles present. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_0 and the two point mixture model for two methods—the traditional consensus method ("traditional") and the probabilistic method ("probabilistic").

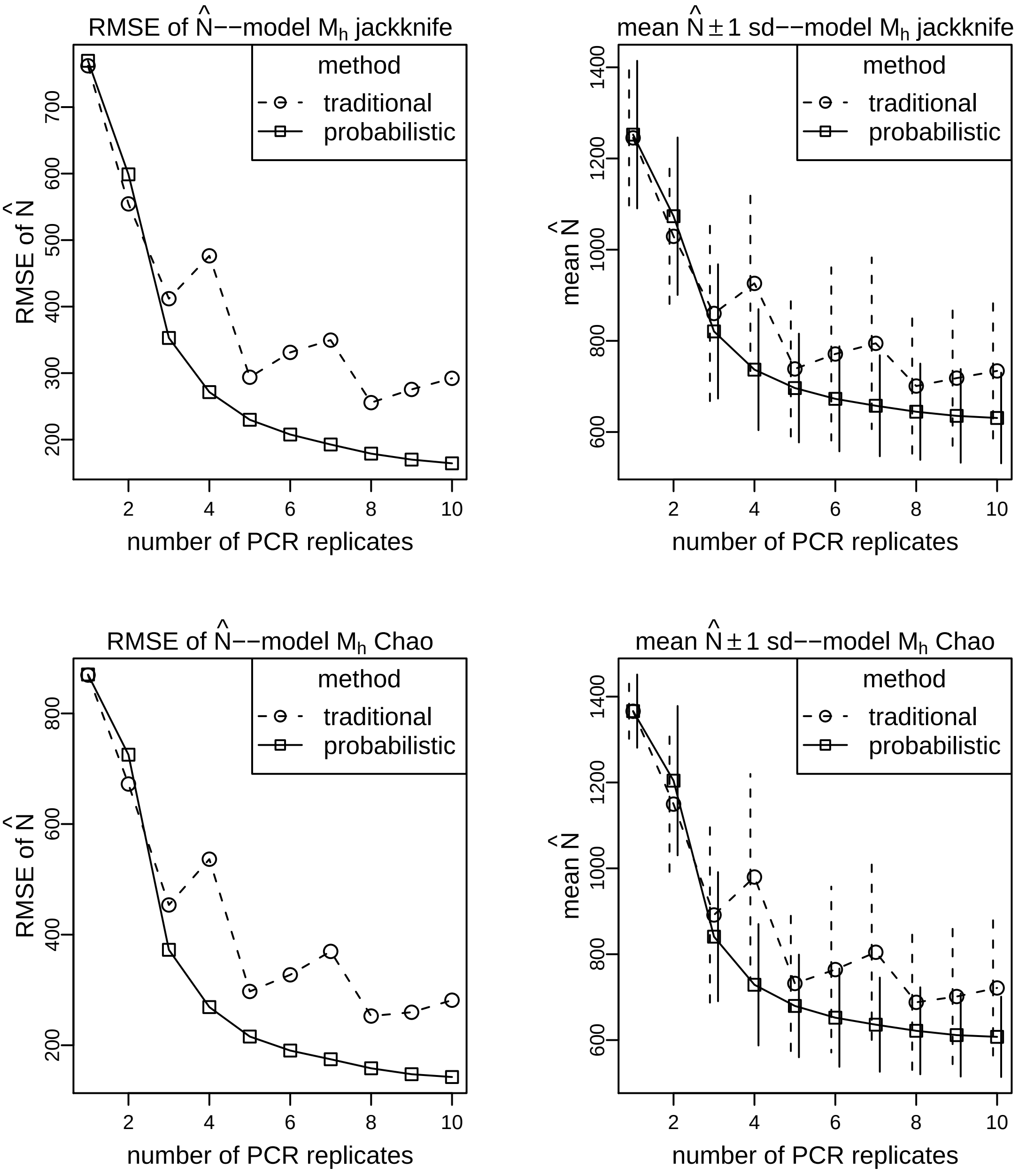


Figure D.32: Results for simulation 16: N=500, capture probability=.5, sampling occasions=10, null alleles present. Root mean squared error of the population size estimator (RMSE of $\hat{N}$), or mean population size estimate (mean $\hat{N}$), versus the number of PCR replicates. Vertical lines on mean estimate graphs indicate ±1 standard deviation of the estimates. Results for model M_h jackknife and model M_h Chao for two methods—the traditional consensus method ("traditional") and the probabilistic method ("probabilistic").