FINAL REPORT SUBMITTED TO:

Pacific Salmon Commission's Chinook Technical Committee (US Section) for
Funding under the Letter of Agreement (LOA)

12 MARCH 2010

PROJECT TITLE:

Computational algorithms and user-friendly software for parentage-based tagging of Pacific salmonids

PRINCIPAL INVESTIGATOR:

Eric C. Anderson
Fisheries Ecology Division
Southwest Fisheries Science Center
110 Shaffer Road
Santa Cruz, CA 94920-1211

FOR WORK ORIGINALLY PROPOSED FOR THE PERIOD:

1 June 2008 through 31 May 2009

Contents

REITERATION OF OBJECTIVES

The 1985 Pacific Salmon Treaty (PST) mandates that salmon fishery management should provide an appropriate allocation of harvest to the United States and Canada and should prevent overfishing and encourage rebuilding of Pacific salmon stocks. These are difficult management challenges because most salmon harvest occurs in the ocean where fish from different stocks are intermingled. For the last 30 years, the primary tool for distinguishing fish from different stocks, and for estimating stock-, age-, and fishery-specific harvest and mortality rates, has been the coded wire tag (CWT). These tiny tags are implanted in a known fraction of juvenile fish produced in various rivers and hatcheries. When those fish are caught, the code on the tag may be read under a microscope to reveal the age of the fish, the location of its origin, and possibly the group with which it was released from a hatchery. Data from this massive effort are used to parameterize simulation models that guide seasonal fishery management decisions. While CWTs have been instrumental in the implementation of the PST, the CWT program currently faces great challenges, most notably from federal and state laws requiring mass-marking of hatchery-produced salmon. Since mass-marking will make the recovery of CWTs from fisheries and terminal escapements more difficult and costly, the Pacific Salmon Commission (PSC) has been investigating alternatives to CWTs and ways to complement the CWT program.

Genetic methods have been used for salmon management since the 1980's, and emerging genetic technologies hold great promise. Genetic Stock Identification (GSI), first with protein polymorphisms, and now with microsatellites and single nucleotide polymorphisms (SNPs), can yield reliable estimates of the proportion of fish from different stocks or "reporting groups" in mixed stock fisheries. However, GSI cannot provide the data on age and release group that is required for today's fisheries management models. Only a single genetic method has been proposed that could supply age and release group data like CWTs: the method of Parentage Based Tagging (PBT—formerly called Full Parental Genotyping) proposed by Anderson and Garza (see Hankin et al., 2005, pp. 79–90). The PBT method involves genotyping hatchery broodstock with SNPs and recording their genotypes in a data base of parents. Genotypes taken from fishery samples can be compared to this data base, and, if the parents of the fishery sample are found, this provides the age and hatchery of origin of the fishery sample, and can also be used to determine the release group. The feasibility of parentage inference on such a large scale was demonstrated in Anderson & Garza (2006), and there are now numerous federal and state labs pursuing, or proposing, projects to validate the PBT concept and demonstrate its use. However, there is currently no software capable of efficiently analyzing the data from these projects.

In this study, I built on the mathematical methods introduced in Anderson & Garza (2006) to develop a software package capable of the large scale, likelihood-based, parentage inference required to make SNP-based PBT both feasible and economical. This software provides important savings in genotyping costs for all PBT studies. Previously available parentage methods based on likelihood methods were unable to handle such large problems and the simpler methods, based on Mendelian incompatibilities, though possibly applicable to large scale parentage inference could require up to 45% more SNP markers to achieve the same power as a likelihood-based analysis (Anderson & Garza, 2006). Thus, our software implementing likelihood-based parentage could save roughly $40,000 to $150,000 in genotyping costs, *per study*, depending on the number of individuals genotyped.

In this study we accomplished the following:

1. Developed a software program optimized to rapidly search for feasible parent-offspring trios in a large, parent data base, and then developed (and implemented in software) the mathematical machinery to compute the likelihood for those trios, conditional on their being feasible under the terms of the rapid search. These conditional distributions were then used to compute p-values for individuals parent-offspring trio assignments, and these were used in a False Discovery Rate framework that allows control of the Type I error rate, even in the absence of prior knowledge about the fraction of sampled parents.

2. Extended the method and software to handle missing data (a ubiquitous feature of real data that isn't well accounted for in any existing methods) and to simultaneously handle multiple hatchery populations with different allele frequencies at the SNPs under study. This ensures that the software will be scalable to PBT on a very large (*i.e.*, coastwide) scale.

3. Developed methods to sharpen parentage inferences in the face of related individuals occurring in the parent database. This was done by incorporating a prior on the expected fraction of different relationship categories.

4. Rigorously tested the methods and software developed, and applied the software to PBT simulations at a California-coastwide scale using allele frequencies estimated from 96 SNPs developed at the SWFSC lab.

5. Compiled the program for both Mac OS X and Windows operating systems. The software and its source code are freely available from
`http://swfsc.noaa.gov//staff.aspx?Division=FED&id=740`.

OVERVIEW

The following report is divided into three parts. The first is the technical, mathematical description of the method. The second describes the software implementation and documents how to use the software. The third part describes the large set of simulations using the software which verify that it can handle very large data sets and that it provides good results.

Part I

Statistical and Computational Advances

INTRODUCTION

Likelihood-based pedigree reconstruction methods are increasingly used in studies of natural populations (Pemberton, 2008). The scale of these studies is growing as molecular ecologists adopt new and more efficient genotyping technologies from the fields of human and medical genetics. In particular, the recent development of single nucleotide polymorphism (SNP) markers in non-model organisms has enabled the rapid genotyping of many thousands of individuals in commercially important species such as Pacific salmon (Elfstrom et al., 2006), Atlantic salmon (Hayes et al., 2007), beef cattle (Heaton et al., 2002), and pigs (Fahrenkrug et al., 2002). This capacity makes it possible to reconstruct pedigree relationships with genetic data from amongst tens of thousands of candidate parents (Anderson & Garza, 2006) and is revolutionizing the management of livestock operations and hatchery-propagated fish populations. It is only a matter of time before similar genotyping capacity is realized in many other species; however, the likelihood-based methods in use today were not designed for parentage inference on a very large scale, and many are not computationally efficient enough to handle large quantities of data. This paper introduces a number of novel statistical and computational approaches to likelihood-based parentage inference with SNPs that improve upon currently-available methods and that are efficient enough to allow the rapid and accurate calculation of statistical confidence for individual parentage assignments, even when the number of candidate parents is very large.

Thompson (1976b) introduced likelihood methods for pedigree reconstruction in human populations. These methods were first adapted and applied to nonhuman populations in Meagher & Thompson (1987). This approach to pedigree reconstruction employs the log-odds (LOD)—the log of the probability of the offspring and putative-parent genotypes under the hypothesis of parentage divided by the probability under the hypothesis that the offspring is unrelated to the putative parent(s)—and focuses on parentage inference in two steps: 1) the identification of parent-offspring pairs, then 2) the identification of parent-pair + offspring trios from amongst the putative parent-offspring pairs having high LODs. Meagher & Thompson (1986) showed that conducting step 2 using only high-LOD parent-offspring pairs is statistically justifiable, and Thompson & Meagher (1987) investigated approaches to mitigate the fact, noted by Thompson (1976a), that a full sibling of the offspring, when not excluded from parentage on the basis of Mendelian incompatibility, gives a higher LOD score on average than the true parent. This work did not, however, describe a means for estimating the statistical confidence in individual parentage assignments.

Marshall et al. (1998) extended the likelihood-based approaches of Meagher & Thompson (1987) by allowing for genotyping error, and by developing a Monte Carlo scheme to estimate statistical confidence in parentage assignments. The method and its revisions (Kalinowski et al., 2007), implemented in the user-friendly software program CERVUS, have been instrumental in advancing the practice of likelihood-based parentage, and, as the *de facto* standard method of parentage inference amongst molecular biologists, CERVUS has been used in hundreds of natural population studies. Nonetheless, the statistical approaches implemented within CERVUS, and in most other related, likelihood-based and Bayesian methods, (*e.g.*, Neff et al., 2001; Duchesne et al., 2002; Cercueil et al., 2002; Hadfield et al., 2006) could be improved in several ways. Here we develop methods that allow the following four improvements, which we enumerate below to allow referencing them later in the paper.

1. The LOD scores computed by CERVUS are identical whether or not there is any prior probability that the sample includes putative parents that are related in some way to the true parents or to the offspring. It would be preferable to include, as suggested by Thompson & Meagher (1987), terms in the LOD for the possibility of related but nonparental individuals amongst the putative parents.

2. Statistical significance of individual parentage assignments are assessed by comparing the observed LOD of the trio with the simulated distribution of LODs expected *marginally* for a trio drawn randomly from a population with similar allele frequencies, rates of missing data, and rates of occurrence of related individuals; however, the genotype of each individual offspring is held constant when comparing it to multiple possible parents, so that the relevant null distribution is the distribution of trio LODs *conditioned* on the offspring genotype.

3. The significance values computed by CERVUS are posterior predictive values and, as such, require that the user specify, as a prior probability, the fraction of possible parents in the population that are included in the genetic sample. While this fraction may be known in some closed study populations, in others it might not be known at all. For such cases, we explore the adaptive control of the false discovery rate (Benjamini & Hochberg, 1995, 2000) as an alternative.

4. Finally, as the scale of the parentage inference problem grows (more putative parents and more offspring) the simulation procedure in CERVUS requires a very large amount of time, making it unattractive to apply to large scale problems in Pacific salmon fisheries management. Our methods reduce running times by orders of magnitude.

Improvements 1 and 3 require only minor adjustments to existing methods. Improvements 2 and 4 require more novel methods. We develop an approach based upon the simulation of distributions of LODs conditioned on a maximum number of Mendelian incompatibilities in a trio. This is made efficient by adopting methods from the analysis of hidden Markov chains (Baum et al., 1970). Such an approach is computationally feasible for SNP markers, and we focus upon SNPs entirely in this paper.

In the following section we define notation and describe the statistical method. Subsequently we evaluate the method using simulated data.

METHODS

Data, notational conventions, and preliminaries

We assume that individuals in the study are diploids with genetic data at L independently segregating SNP loci. At each locus ℓ there are two alleles: one labeled 0 and having frequency q_ℓ in the population under study and the other labeled 1 and having the frequency $p_\ell = 1 - q_\ell$. At each locus the genotype g of an individual is the number of 1 alleles it carries (*i.e.*, $g = 0$, 1, or 2), or, if the individual was not successfully genotyped at the locus, $g = \bullet$. We have a list $\mathscr{O}$ of offspring individuals whose parents we wish to infer from amongst a collection of possible fathers

($\mathscr{S}$ for "sires") and mothers ($\mathscr{D}$ for "dams"). There may be individuals in $\mathscr{O}$ whose parents are not in $\mathscr{S}$ or $\mathscr{D}$. Our goal is to infer, for every individual i in $\mathscr{O}$ the *pair* of i's parents, if present in both $\mathscr{S}$ and $\mathscr{D}$. In our application, we are not interested, for example, in inferring only the father when the mother is not in $\mathscr{D}$ because in fish hatcheries one can ensure that the mates of every female included in $\mathscr{D}$ are in $\mathscr{S}$, and vice-versa. There might additionally be information about the possible matings ($\mathscr{C}$ for "crosses") between the members of $\mathscr{S}$ and $\mathscr{D}$ and there may be additional data, collectively $\mathscr{H}$, such as age information that can be used to exclude certain individuals from parentage.

Being interested in inferring parent pairs by likelihood, a basic unit whose probability is of interest is the trio of putative youth, father, and mother. At the ℓ^{th} locus the genotype of such a trio is the triplet $(g_\ell^{\mathrm{kid}}, g_\ell^{\mathrm{pa}}, g_\ell^{\mathrm{ma}})$ which we will denote by a_ℓ, the values of which we will write without commas (*e.g.*, $a_\ell = 000$ or $a_\ell = 10\bullet$). Note that the superscript kid, pa, and ma refer to the *putative* youth, *putative* father and *putative* mother, respectively. When all individuals are successfully genotyped the 27 possible states of a_ℓ are the set $\mathscr{A} = \{000, 001, 002, 010, \dots, 221, 222\}$. When as many as three individuals in the trio can have missing data at the locus the 64 possible states are the set $\mathscr{A}^\bullet = \{000, 001, 002, 00\bullet, 010, \dots, \bullet\bullet 2, \bullet\bullet\bullet\}$. For values of $a_\ell \in \mathscr{A}$ the probability of a_ℓ depends on the allele frequency p_ℓ, the genotyping error rate at the locus μ_ℓ, and the true relationship r of the members of the trio. We denote this probability $P(a_\ell|r)$ taking the dependence on p_ℓ and μ_ℓ as always implicit. These probabilities are easily computed for any possible model of genotyping error and any r by simply summing over the genotypes of any relevant but unobserved individuals in the pedigree describing r and over the unobserved true genotypic states underlying the observed, possibly erroneous genotypes. Details can be found in the appendix of Anderson & Garza (2006). We will make use of $P(a_\ell|r, g_{\mathrm{kid}})$, the conditional probability of a_ℓ given r and the genotype of the kid in the trio. This probability is proportional to $P(a_\ell|r)$ for all states a_ℓ consistent with g_{kid} and 0 otherwise, so is also computed easily. Over L independently-segregating loci which are not in linkage disequilibrium in the population, the probability of $\boldsymbol{a} = (a_1, \dots, a_L)$ given r is simply a product, $P(\boldsymbol{a}|r) = \prod_{\ell=1}^{L} P(a_\ell|r)$. Again, the dependence on $\boldsymbol{\mu} = (\mu_1, \dots, \mu_L)$ and $\boldsymbol{p} = (p_1, \dots, p_L)$ may be omitted in the notation, but is implicit.

We assume that the data missing at any member of a trio are missing at random and do not attempt to model the missing data. Instead, the missing data are merely omitted, and hence the probability of the observed data at a locus given missing data at some members of the trio is computed by summing the values of $P(a_\ell|r)$ for $a_\ell \in \mathscr{A}$ over the unobserved members. For example, we define

$$P(a_\ell = 1\bullet 2|r) = \sum_{k=0}^{2} P(g^{\mathrm{kid}} = 1, g^{\mathrm{pa}} = k, g^{\mathrm{ma}} = 2|r)$$

and the extensions to data missing at other members (or at more members) of the trio or to conditioning on g_{kid} are obvious.

A cornerstone of our method involves excluding parent-offspring pairs and trios on the basis of Mendelian incompatibility and then conducting Monte Carlo simulation of LODs conditional on the fact that trios were not excluded by such a screening. We introduce some notation to describe that here. We let $\boldsymbol{v}(a_\ell)$ be a vector of three binary components whose values indicate the manner in which the observed genotypes of a trio are, or are not, compatible with Mendelian inheritance between a mother, father, and offspring. $\boldsymbol{v}(a_\ell)$ always depends on a_ℓ and so the a_ℓ may sometimes

Table 1: Patterns of Mendelian incompatibility $\boldsymbol{v}$ with corresponding trio genotype states a_ℓ.

$\boldsymbol{v}$	$a_\ell \in \mathscr{A}$			Additional $a_\ell \in \mathscr{A}^\bullet$					
$(0,0,0)$	000	001	010	00●	01●	0●0	0●1	0●●	10●
	011	101	102	11●	12●	1●0	1●1	1●2	1●●
	110	111	112	21●	22●	2●1	2●2	2●●	●00
	120	121	211	●01	●02	●0●	●10	●11	●12
	212	221	222	●1●	●20	●21	●22	●2●	●●0
				●●1	●●2	●●●			
$(0,0,1)$	100	122							
$(0,1,1)$	002	012	210	0●2	2●0				
	220								
$(1,0,1)$	020	021	201	02●	20●				
	202								
$(1,1,1)$	022	200							

be dropped from the notation. The first element of $\boldsymbol{v}$ is 1 if pa and kid are Mendelian-incompatible and 0 otherwise; the second element of $\boldsymbol{v}$ is 1 if ma and kid are incompatible and 0 otherwise; the third element of $\boldsymbol{v}$ is 1 if either pa or ma are incompatible, considered alone, with kid, or, when taken together, pa and ma are not compatible as a pair of parents for kid. Values of $\boldsymbol{v}$ are written with commas like $\boldsymbol{v} = (1,0,1)$. An individual with missing data at a locus is deemed to provide no evidence that can be used to declare Mendelian incompatibility. There are 5 possible values of $\boldsymbol{v}$, each one corresponding to a subset of $\mathscr{A}$ and $\mathscr{A}^\bullet$ as summarized in Table 1.

The probability that $\boldsymbol{v}$ at locus ℓ takes a values $\boldsymbol{v}^*$ is computed by a sum over genotype states a_ℓ:

$$P(\boldsymbol{v}(a_\ell) = \boldsymbol{v}^* | r) = \sum_{a' : \boldsymbol{v}(a'_\ell) = \boldsymbol{v}^*} P(a'_\ell | r).$$

The same holds when conditioning on g_{kid}:

$$P(\boldsymbol{v}(a_\ell) = \boldsymbol{v}^* | r, g_{\mathrm{kid}}) = \sum_{a' : \boldsymbol{v}(a'_\ell) = \boldsymbol{v}^*} P(a'_\ell | r, g_{\mathrm{kid}}).$$

We also develop notation to express the cumulative number of Mendelian incompatibilities observed at the first k SNP loci in a trio having genotypes $\boldsymbol{a}$. This is $\boldsymbol{v}^{(k)}(\boldsymbol{a}) = \sum_{\ell=1}^{k} \boldsymbol{v}(a_\ell)$, and may be written simply as $\boldsymbol{v}^{(k)}$. The components of this vector are written as $v_1^{(k)}$, $v_2^{(k)}$, and $v_3^{(k)}$, and if we write $\boldsymbol{v}^{(k)} \leq \boldsymbol{v}^{*(k)}$ it means that $v_1^{(k)} \leq v_1^{*(k)}$, $v_2^{(k)} \leq v_2^{*(k)}$, and $v_3^{(k)} \leq v_3^{*(k)}$. The analogous convention holds if we write $\boldsymbol{v}^{(k)} \geq \boldsymbol{v}^{*(k)}$.

In many parentage applications, the log likelihood ratio or LOD, $\Lambda(\boldsymbol{a}) = \log[P(\boldsymbol{a}|QQ)/P(\boldsymbol{a}|UU)]$ is employed to compare candidate parents of an individual. Here QQ denotes the hypothesis that pa and ma are the true parents of kid and UU denotes the hypothesis that pa and ma are completely unrelated to kid. This statistic is most appropriate when all trios in the sample are either QQ or UU, which will seldom be the case because many individuals in a finite population will be related to some degree. In such cases a preferable test statistic will be the posterior probability of parentage for a trio, as suggested by Thompson & Meagher (1987). Denoting by $\mathscr{R}$ the set of relationships amongst a trio that will be considered, and assuming that a prior probability π_r is available for all $r \in \mathscr{R}$ and $\sum_{r\in\mathscr{R}} \pi_r = 1$, the posterior probability of parentage is $P(QQ|\boldsymbol{a}, \boldsymbol{\pi}) = \pi_{QQ}P(\boldsymbol{a}|QQ)/[\sum_{r\in\mathscr{R}} \pi_r P(\boldsymbol{a}|r)]$. In fish hatchery applications, there is typically enough information on sizes of spawning populations in the past, that a reasonable estimate of $\boldsymbol{\pi} = (\pi_r)_{r\in\mathscr{R}}$ can be made. A recursive method for doing so in populations semelparous organsims with overlapping age structure, like salmon, was developed, but will not be described here.

In the populations that we study, we determined by simulation that there are 18 trio relationship categories which, given their expected chances of occurrence and their probability of being mistaken for a parental trio, should be included in $\mathscr{R}$. In populations with different demographic structure it may be beneficial to include more or fewer trio relationships in $\mathscr{R}$. In all 18 of these categories, the individuals are assumed to be noninbred, so we do not consider categories in which, for example, a candidate father is both a sibling and the true father of the putative offspring; however, such trio categories could be accommodated without great difficulty. The first nine trio relationship categories involve situations in which ma or pa share a unilineal relationship to a noninbred kid through the true parents. Following Anderson & Garza (2006) these are the C-type relationships all of which may be denoted by $\mathrm{C}^{\mathrm{pa}}_{\mathrm{ma}}$ where pa and ma are placeholders for the relationship (Se for self, Si for full sibling, U for unrelated) between pa and a true parent and ma and the other true parent, respectively. For example, the QQ relationship is $\mathrm{C}^{\mathrm{Se}}_{\mathrm{Se}}$ and the U relationship is $\mathrm{C}^{\mathrm{U}}_{\mathrm{U}}$, and we will refer to them as such for the remainder of the paper. The next eight trio relationship categories that we consider are those in which exactly one of ma or pa is related as a full sibling or as a half sibling with kid and the other candidate parent is related unilineally to the true parents of kid through a relationship (Se, Si, or U) with one of the true parents. We denote these trio relationships by F (for full sibling) or H (for half sibling) adorned with a superscript or subscript Se, Si, or U, if the candidate that does not have the full- or half-sibling relationship with kid is pa or ma, respectively. For example, F_{Si} indicates that pa is the full sibling of kid and ma is the full sibling of the true mother (or the true father), and H^{U} indicates that ma is a half-sibling of kid and pa is unrelated to either of the true parents. The final trio relationship that we consider is FF—both pa and ma are full siblings of kid. Some of these 18 relationship categories may contain up to two underlying pedigree relationships owing to the fact that, in some cases, the candidate parents may be related to the true parents of like or opposite sex. This distinction becomes important when predicting $\boldsymbol{\pi}$ recursively. The 18 categories and the 22 states underlying them are enumerated and described in Table 2.

Using the notation for relationships above, our test statistic for assessing confidence in an assignment of parentage of a kid to a pa and ma having trio genotypes of $\boldsymbol{a}$ is the probability:

$$P(\mathrm{C}^{\mathrm{Se}}_{\mathrm{Se}}|\boldsymbol{a}) = \frac{\pi_{\mathrm{C}^{\mathrm{Se}}_{\mathrm{Se}}} P(\boldsymbol{a}|\mathrm{C}^{\mathrm{Se}}_{\mathrm{Se}})}{\sum_{r\in\mathscr{R}} \pi_r P(\boldsymbol{a}|r)}. \tag{1}$$

Table 2: Relationship categories in $\mathscr{R}$ and the relationships underlying them. In the Type column, the subscript refers to the putative female parent and the subscript refers to the putative male parent.

PFR idx	Type	$\langle \mathrm{pa}, \mathrm{s} \rangle$	$\langle \mathrm{pa}, \mathrm{d} \rangle$	$\langle \mathrm{ma}, \mathrm{d} \rangle$	$\langle \mathrm{ma}, \mathrm{s} \rangle$	$\langle\langle \mathrm{pa}, \mathrm{kid} \rangle\rangle$	$\langle\langle \mathrm{ma}, \mathrm{kid} \rangle\rangle$	Flat Text Name
0	$\mathrm{C}^{\mathrm{Se}}_{\mathrm{Se}}$	Se	U	Se	U	–	–	`C_Se_Se`
1	$\mathrm{C}^{\mathrm{Se}}_{\mathrm{Si}}$	Se	U	Si	U	–	–	`C_Se_Si`
2	$\mathrm{C}^{\mathrm{Si}}_{\mathrm{Se}}$	Si	U	Se	U	–	–	`C_Si_Se`
3	$\mathrm{C}^{\mathrm{Se}}_{\mathrm{U}}$	Se	U	U	U	–	–	`C_Se_U`
4	$\mathrm{C}^{\mathrm{U}}_{\mathrm{Se}}$	U	U	Se	U	–	–	`C_U_Se`
5	$\mathrm{C}^{\mathrm{Si}}_{\mathrm{Si}}$	Si	U	Si	U	–	–	`C_Si_Si`
5	$\mathrm{C}^{\mathrm{Si}}_{\mathrm{Si}}$	U	Si	U	Si	–	–	`C_Si_Si`
6	$\mathrm{C}^{\mathrm{Si}}_{\mathrm{U}}$	Si	U	U	U	–	–	`C_Si_U`
6	$\mathrm{C}^{\mathrm{Si}}_{\mathrm{U}}$	U	Si	U	U	–	–	`C_Si_U`
7	$\mathrm{C}^{\mathrm{U}}_{\mathrm{Si}}$	U	U	Si	U	–	–	`C_U_Si`
7	$\mathrm{C}^{\mathrm{U}}_{\mathrm{Si}}$	U	U	U	Si	–	–	`C_U_Si`
8	$\mathrm{C}^{\mathrm{U}}_{\mathrm{U}}$	U	U	U	U	–	–	`C_U_U`
9	F^{Se}	Se	–	–	–	–	F	`Se_F`
10	F_{Se}	–	–	Se	–	F	–	`F_Se`
11	H_{Se}	–	–	Se	–	H	–	`H_Se`
12	H^{Se}	Se	–	–	–	–	H	`Se_H`
13	F_{Si}	–	–	–	Si	F	–	`F_Si`
13	F_{Si}	–	–	Si	–	F	–	`F_Si`
14	F^{Si}	–	Si	–	–	–	F	`Si_F`
14	F^{Si}	Si	–	–	–	–	F	`Si_F`
15	F_{U}	–	–	U	U	F	–	`F_U`
16	F^{U}	U	U	–	–	–	F	`U_F`
17	FF	–	–	–	–	F	F	`F_F`

This form is convenient and familiar. It is also functionally equivalent to using a LOD or a likelihood ratio criterion for an alternative hypothesis of parental relationship ($\mathrm{C}_{\mathrm{Se}}^{\mathrm{Se}}$) versus a null hypothesis of "non-parental" relationship (*i.e.*, $\{\mathscr{R}\backslash\mathrm{C}_{\mathrm{Se}}^{\mathrm{Se}}\}$—the set $\mathscr{R}$ excluding $\mathrm{C}_{\mathrm{Se}}^{\mathrm{Se}}$). This equivalence follows from the fact that (1) is monotonically increasing with the likelihood ratio

$$\frac{P(\boldsymbol{a}|\mathrm{C}_{\mathrm{Se}}^{\mathrm{Se}})}{\sum_{r\in\{\mathscr{R}\backslash\mathrm{C}_{\mathrm{Se}}^{\mathrm{Se}}\}}(\pi_r/\pi_{\mathrm{C}_{\mathrm{Se}}^{\mathrm{Se}}})P(\boldsymbol{a}|r)}.$$

Overview of the method

Here we give an overview of our method, providing further detail on certain aspects and calculations in subsequent sections. The main steps are as follows:

1. Data are read in and values of parameters are initialized:
    - The genotypes of the individuals in $\mathscr{S}$ and $\mathscr{D}$ are used together to make an estimate of p_ℓ for each locus by the posterior mean given a $\mathrm{Beta}(\frac{1}{2},\frac{1}{2})$ prior and the data in $\mathscr{S}$ and $\mathscr{D}$. This estimate is taken to be the value p_ℓ used in all probability calculations in the preceding and following sections.
    - Values of μ_ℓ, $\ell = 1,\dots,L$, are assumed known from other sources of data, experiments, or prior beliefs.
    - Values of $\boldsymbol{\pi}$ are estimated from demographic data and from assumptions or estimates of variance in reproductive success. These estimates of $\boldsymbol{\pi}$ are used in the method as if known without error.
2. A value of $\boldsymbol{v}^{(L)}$, denoted $\boldsymbol{v}^{(L)\max}$, is chosen such that, given $\boldsymbol{\mu}$ and $\boldsymbol{p}$ there is only a small probability, β^{MI}, that a truly parental trio will have a $\boldsymbol{v}^{(L)}$ with any of its three components exceeding the corresponding component of $\boldsymbol{v}^{(L)\max}$. That is:

$$1-\beta^{\mathrm{MI}} = \sum_{a_\ell\in\mathscr{A},\ \ell=1,\dots,L} \mathcal{I}\{\boldsymbol{v}^{(L)}(\boldsymbol{a}) \le \boldsymbol{v}^{(L)\max}\}P(\boldsymbol{a}|\mathrm{C}_{\mathrm{Se}}^{\mathrm{Se}}) \tag{2}$$

    where $\mathcal{I}\{x\}$ is the indicator function returning 1 if x is true and 0 otherwise. The superscript MI stands for "Mendelian Incompatibility." β^{MI} is the rate at which truly parental trios will not be identified in our parentage inference exercise due to the fact that they have too many Mendelian incompatibilities. In practice, we use values of β^{MI} on the order of 0.001. The sum in (2) is calculated efficiently via a recursion which is the forward step of the forward-backwards algorithm described later.
3. Each individual i in $\mathscr{O}$ is compared against every male in $\mathscr{S}$ that is a potential father of i according to $\mathscr{H}$, and a list $\mathrm{Pas}^{(i)}$ is maintained of those potential fathers j having no more than $v_1^{(L)\max}$ Mendelian incompatibilities with i. Likewise, each i is compared to every female in $\mathscr{D}$ that qualifies as a potential mother of i according to $\mathscr{H}$, and a list $\mathrm{Mas}^{(i)}$ is made of potential mothers k having no more than $v_2^{(L)\max}$ Mendelian incompatibilities with i.

4. The genotype of each i in $\mathscr{O}$ is compared to every pair (j,k) of $j \in \mathrm{Pas}^{(i)}$ and $k \in \mathrm{Mas}^{(i)}$ such that j and k are a possible mated pair according to $\mathscr{C}$, and a list $\mathrm{Pairs}^{(i)}$ is maintained of all parent pairs such that the trio they form with i has $\boldsymbol{v}^{(L)} \leq \boldsymbol{v}^{(L)\max}$. Let $N^{(i)}$ denote the number of elements in $\mathrm{Pairs}^{(i)}$, and take the elements of $\mathrm{Pairs}^{(i)}$ to be sorted by the largest to smallest value of $P(\mathrm{C}^{\mathrm{Se}}_{\mathrm{Se}}|\boldsymbol{a},\boldsymbol{\pi})$. Thus, $\mathrm{Pairs}^{(i)}_1$ is the pair of potential parents with highest posterior probability of parentage to offspring i.

5. The pair $\mathrm{Pairs}^{(i)}_1$ is assigned parentage to i, and the statistical confidence in that assignment is assessed by comparison to a "null" distribution approximated via Monte Carlo by simulating for M replicates $N^{(i)}$ pairs of non-parental genotypes drawn conditional on $\boldsymbol{\pi}$ and the fact that they must have no more than $\boldsymbol{v}^{(L)\max}$ incompatibilities with i, and recording for each of the M replicates the highest values of $P(\mathrm{C}^{\mathrm{Se}}_{\mathrm{Se}}|\boldsymbol{a},\boldsymbol{\pi})$ amongst the $N^{(i)}$ simulated values. This simulation, described fully in Section I, makes extensive use of the forward-backwards algorithm.

6. Interpreting the statistical confidence computed in step 5 as p-values, we then use them to control the false discovery rate (Benjamini & Hochberg, 1995). Even when an estimate of the fraction of sampled parents is not available, use of the adaptive procedure of Benjamini & Hochberg (2000) can provide a reasonable estimate of the fraction of Pairs_i observed that are true parent pairs, and this allows for more powerful control of the rate of incorrect parentage assignments.

Like most methods for inferring parent pairs, we first identify individual males and females with a good chance of being parents, and then we restrict our attention to the pairs formed from that small group of males and females. However, instead of using both Mendelian incompatibility *and* the parent-offspring LOD to initially screen individual males and females (as done in Meagher & Thompson 1987), we screen candidate males and females solely on the basis of the number of loci with Mendelian incompatibilities. At first this may seem disadvantageous compared to using LODs, however, it allows the assessment of statistical significance of individual parentage assignments by performing simulations while conditioning on the fact that only $N^{(i)}$ pairs had sufficiently few Mendelian incompatibilities to be included in $\mathrm{Pairs}^{(i)}$. By contrast, it is not clear how one could efficiently simulate genotypes while conditioning on the LOD exceeding a certain amount.

Typically $N^{(i)}$ is substantially smaller than the number of candidate males or females in the study, so, each Monte Carlo replicate from the null distribution requires simulating the genotypes of only $N^{(i)}$ pairs. In large studies this becomes quite important. For example, if there are 10^4 candidate males and 10^4 candidate females, but $N^{(i)}$ is only 10, then each Monte Carlo replicate requires only 10 realizations of genotype pairs. Contrast this with the standard simulation routine of CERVUS: each Monte Carlo replicate requires simulating 10^4 male and 10^4 female genotypes, each of those genotypes must be compared to a single offspring genotype, then all 10^4 males and females must be sorted, and finally some number of simulated male-female pairs are compared to an individual offspring genotype.

In the following section we show how to simulate a pair of genotypes conditional on r and $\boldsymbol{v}^{(L)}(\boldsymbol{a}) \leq \boldsymbol{v}^{(L)\max}$.

Table 3: Some notation used in the paper

g_ℓ	genotype at locus ℓ: the number of "1" alleles, or $\bullet$ if missing.
p_ℓ	the relative frequency of the "1" allele at locus ℓ in the population.
q_ℓ	relative frequency of the "0" allele at locus ℓ. $q_\ell = 1 - p_\ell$.
$\mathscr{O}$	list of offspring whose parents are to be inferred.
$\mathscr{S}$	list of possible fathers (sires).
$\mathscr{D}$	list of possible mothers (dams).
$\mathscr{C}$	information, if available, detailing which members of $\mathscr{S}$ and $\mathscr{D}$ could have mated (crosses).
$\mathscr{H}$	other information, like age data, which, if available, could be used to exclude some parents from parentage with particular offspring.
kid, pa, pa	an offspring and a *putative* father and mother respectively.
a_ℓ	genotypes at locus ℓ in a kid, pa, and ma: $(g_\ell^{\mathrm{kid}}, g_\ell^{\mathrm{pa}}, g_\ell^{\mathrm{ma}})$.
$\mathscr{A}$	the 27 possible states a_ℓ can take with no missing data.
$\mathscr{A}^\bullet$	the 64 possible states of a_ℓ when missing data is allowed.
μ_ℓ	the rate of genotyping error at locus ℓ.
L	the total number of SNPs in the data set.
$\boldsymbol{p}$, $\boldsymbol{a}$, $\boldsymbol{\mu}$	$(a_1, \ldots, a_L)$, $(p_1, \ldots, p_L)$, and $(\mu_1, \ldots, \mu_L)$, respectively.
r	generically, a relationship between a trio of kid, pa, and ma.
$\mathscr{R}$	the set of relationships r given positive prior probability.
π_r	the prior probability that a kid, pa, and ma drawn at random from the population have relationship r.
$\boldsymbol{v}(a_\ell)$	vector of three binary indicators describing patterns of Mendelian incompatibility in a kid-pa-ma trio at locus ℓ. Sometimes denoted simply by $\boldsymbol{v}$.
$\boldsymbol{v}^{(k)}(\boldsymbol{a})$	cumulative number of Mendelian incompatibilities (of certain types) at loci 1 through k. Also denoted simply by $\boldsymbol{v}^{(k)}$. $\boldsymbol{v}^{(k)}(\boldsymbol{a}) = \boldsymbol{v}^{(k)} = \sum_{\ell=1}^{k} \boldsymbol{v}(a_\ell)$. Has three components: $\boldsymbol{v}^{(k)} = (v_1^{(k)}, v_2^{(k)}, v_3^{(k)})$
$\boldsymbol{v}^{(k)} \leq \boldsymbol{v}^{*(k)}$	shorthand for componentwise equality/inequality: $v_1^{(k)} \leq v_1^{*(k)}$, $v_2^{(k)} \leq v_2^{*(k)}$, and $v_3^{(k)} \leq v_3^{*(k)}$.

Forward-backwards algorithm

For convenience we define $\boldsymbol{v}^{(0)} = (0,0,0)$. This definition merely says that, "Before looking at the genetic data at any loci, there are zero Mendelian incompatibilities of any type." It is apparent that, conditional on r, $\boldsymbol{\mu}$, and $\boldsymbol{p}$, the variables $\boldsymbol{v}^{(\ell)}$, $\ell = 1,\dots,L$, form a Markov chain. That is,

$$P(\boldsymbol{v}^{(\ell)}|\boldsymbol{v}^{(0)},\dots,\boldsymbol{v}^{(\ell-1)}) = P(\boldsymbol{v}^{(\ell)}|\boldsymbol{v}^{(\ell-1)}) \ \text{ for } \ \ell = 1,\dots,L.$$

The joint distribution of $\boldsymbol{a}$ and all the $\boldsymbol{v}^{(\ell)}$'s respects the directed graph shown in Figure 1(a). The arrows from each μ_ℓ and p_ℓ into each $\boldsymbol{v}^{(\ell)}$ run in the reverse direction of a typical hidden Markov chain, but the moralized undirected graph [Figure 1(b)] is easily recognized as having the same undirected graphical structure as a simple hidden Markov chain, especially when collapsing each $\boldsymbol{v}^{(\ell)}$ and a_ℓ, and each μ_ℓ and p_ℓ into single (composite) variables, as in Figure 1(c). Therefore, we can employ the familiar forward-backwards family of algorithms (Baum et al., 1970) to efficiently compute the marginal probability of $\boldsymbol{v}^{(L)}$ and to simulate values of $\boldsymbol{a}$ conditional on $\boldsymbol{v}^{(L)}(\boldsymbol{a}) \le \boldsymbol{v}^{(L)\max}$.

Let $\mathscr{V}^{(\ell)\downarrow}$ denote the set of all vectors $\boldsymbol{v}^{(\ell)} \le \boldsymbol{v}^{(L)\max}$ that can be reached with non-zero probability. Likewise, let $\mathscr{V}^{(\ell)\uparrow}$ be the set of all vectors $\boldsymbol{v}^{(\ell)}$ such that $\boldsymbol{v}^{(\ell)} > \boldsymbol{v}^{(L)\max}$. We will use $\mathscr{V}$ to refer to the five possible values of $\boldsymbol{v}$ (see Table 1). For the current discussion, we will assume that data are not missing at any loci at any of the trio members (*i.e.*, $a_\ell \in \mathscr{A}$). We discuss treatment of missing data in Section I. The probability that $\boldsymbol{v}^{(L)}$ takes a certain value in $\mathscr{V}^{(L)\downarrow}$ can be computed by the forward step recursion:

$$P(\boldsymbol{v}^{(\ell)} = \boldsymbol{v}^*|r) = \sum_{\boldsymbol{v}^{(\ell-1)} \in \mathscr{V}^{(\ell-1)\downarrow}} \sum_{\boldsymbol{v}' \in \mathscr{V}} P(\boldsymbol{v}^{(\ell-1)}|r) P(\boldsymbol{v}(a_\ell) = \boldsymbol{v}'|r) \mathcal{I}\{\boldsymbol{v}^{(\ell-1)} + \boldsymbol{v}' = \boldsymbol{v}^{(\ell)}\} \quad (3)$$

for any value $\boldsymbol{v}^* \in \mathscr{V}^{(\ell)\downarrow}$, for $\ell = 1,\dots,L$. In practice this sum can be calculated for all values of $\boldsymbol{v}^* \in \mathscr{V}^{(\ell)\downarrow}$ by iterating over all the terms in the sum only once. Additionally, it is important to note that, since $\boldsymbol{v}^{(\ell)} \ge \boldsymbol{v}^{(\ell-1)}$ for all ℓ, there is zero probability of reaching any state in $\mathscr{V}^{(t)\downarrow}$ from any state in $\mathscr{V}^{(\ell)\uparrow}$ for any t, ℓ. Hence, so long as the elements of $\boldsymbol{v}^{(L)\max}$ are not large, the sums in (3) can be evaluated quite rapidly.

This recursion is evaluated from $\ell = 1$ to L, and the values of $P(\boldsymbol{v}^{(\ell)} = \boldsymbol{v}^*|r)$ are stored for later use in the backward step. At the end of the forward step, one has obtained $P(\boldsymbol{v}^{(L)} = \boldsymbol{v}^*|r)$ for $\boldsymbol{v}^* \in \mathscr{V}^{(L)\downarrow}$. Summing these values yields the probability that a trio of relationship r, given allele frequencies $\boldsymbol{p}$ and genotyping error rates $\boldsymbol{\mu}$, will have no more than $\boldsymbol{v}^{(L)\max}$ Mendelian incompatbilities:

$$P(\boldsymbol{v}^{(L)} \le \boldsymbol{v}^{(L)\max}|r) = \sum_{\boldsymbol{v}^* \in \mathscr{V}^{(L)\downarrow}} P(\boldsymbol{v}^{(L)} = \boldsymbol{v}^*|r). \quad (4)$$

The conditional probability of each $\boldsymbol{v}^{(L)}$, given that it is in $\mathscr{V}^{(L)\downarrow}$ is found as follows:

$$P(\boldsymbol{v}^{(L)} = \boldsymbol{v}^*|r, \boldsymbol{v}^* \in \mathscr{V}^{(L)\downarrow}) = \frac{P(\boldsymbol{v}^{(L)} = \boldsymbol{v}^*|r)}{P(\boldsymbol{v}^{(L)} \le \boldsymbol{v}^{(L)\max}|r)}. \quad (5)$$

With (5) specified, we now proceed to the backward step.

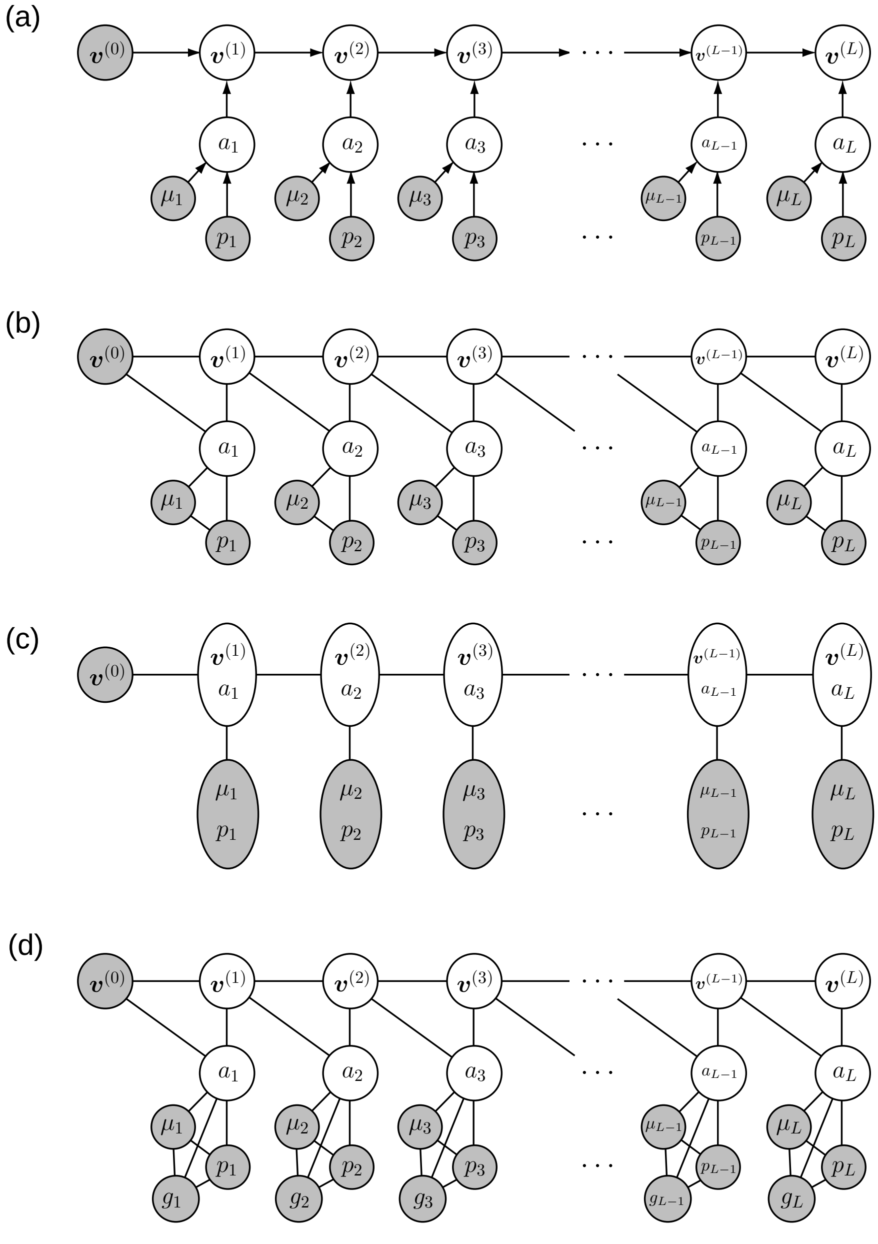

Figure 1: Graphical depictions of the dependence between trio genotypic states, $(a_1, \ldots, a_L)$, and the vectors, $\boldsymbol{v}^{(\ell)}$. Shaded nodes (μ_ℓ = mutation rate, p_ℓ = allele frequency) represent known or fixed quantities to be conditioned upon; unshaded nodes represent variables that we wish to do inference for or that we shall integrate over. The dependence on some trio relationship category r is implicit. (a) The directed graph. (b) Moralized undirected graph. (c) With variables merged into nodes representing several variables together, this more obviously has a hidden Markov chain structure. (d) Conditioning on the offspring genotype is straightforward with the addition of nodes g_ℓ for the offspring genotype.

The goal of the backward step is to simulate a realization of $\boldsymbol{a}$ from its distribution given r, $\boldsymbol{p}$, $\boldsymbol{\mu}$, and conditional on $\boldsymbol{v}^{(L)} \leq \boldsymbol{v}^{(L)\max}$. This is done iteratively starting from $\ell = L$ and working back to $\ell = 1$. Before proceeding, we note that the conditional probability of the observed trio genotypes given the pattern of Mendelian incompatibility is simple to calculate:

$$P(a_\ell = a^*|r, \boldsymbol{v}(a_\ell) = \boldsymbol{v}^*) = \frac{P(a_\ell = a^*|r)}{\sum_{a':\boldsymbol{v}(a')=\boldsymbol{v}^*} P(a_\ell = a'|r)}. \tag{6}$$

The backward step commences by simulating a value of $\boldsymbol{v}^{*(L)}$ from $P(\boldsymbol{v}^{(L)}|r, \boldsymbol{v}^* \in \mathscr{V}^{(L)\downarrow})$ and then simulating the value a_L^* from $P(a_L|r, \boldsymbol{v}(a_L) = \boldsymbol{v}^{*(L)})$. It proceeds by conducting two analogous operations for each ℓ, iteratively, from $L-1$ to 1:

1. Simulate $\boldsymbol{v}^{*(\ell)}$ from a distribution proportional to:

   $$P(\boldsymbol{v}^{(\ell)} = \boldsymbol{v}^{*(\ell)}|r)P(\boldsymbol{v}(a_{\ell+1}) = \boldsymbol{v}^{*(\ell+1)} - \boldsymbol{v}^{*(\ell)}|r)$$

   which is defined over all values of $\boldsymbol{v}^{*(\ell)} \geq \boldsymbol{v}^{(0)}$ that may be obtained by subtracting some $\boldsymbol{v}(a_{\ell+1}) \in \mathscr{V}$ from $\boldsymbol{v}^{*(\ell+1)}$.

2. Simulate a_ℓ^* from $P(a_\ell|r, \boldsymbol{v}(a_\ell) = \boldsymbol{v}^{*(\ell)})$.

At the end of this, $\boldsymbol{a}^* = (a_1^*, \ldots, a_L^*)$ is a realization from $P(\boldsymbol{a}|r, \boldsymbol{v}^{(L)}(\boldsymbol{a}^*) \leq \boldsymbol{v}^{(L)\max})$—the distribution of $\boldsymbol{a}$ conditional on having no more than $\boldsymbol{v}^{(L)\max}$ Mendelian incompatibilities.

As is evident in the graphical structure of Figure 1*d*, the forward-backward algorithm above extends immediately to the case of conditioning both on r and g_{kid}, rather than simply conditioning on r alone. Thus, the meaning of expressions like $P(\boldsymbol{v}^{(L)} \leq \boldsymbol{v}^{(L)\max}|r, g_{\mathrm{kid}})$, and $P(\boldsymbol{a}|r, g_{\mathrm{kid}}, \boldsymbol{v}^{(L)}(\boldsymbol{a}^*) \leq \boldsymbol{v}^{(L)\max})$ should be clear.

Simulation assessment of p-value for a single kid i

For a given kid, i, the ma and pa in $\mathrm{Pairs}_1^{(i)}$ are designated as the best candidates to be the true parents. Let the $P(\mathrm{C}_{\mathrm{Se}}^{\mathrm{Se}}|\boldsymbol{a})$ of this pair have the value $P^{(1)}$. If we declare ma and pa in $\mathrm{Pairs}_1^{(i)}$ the true parents, we risk making the (Type I) error of incorrectly rejecting the null hypothesis of non-parentage, when, in fact $\mathrm{Pairs}_1^{(i)}$ are not the true parents of i. To assess this possibility, we compute a Type I error rate or "p-value" associated with assigning parentage of each kid, i, to $\mathrm{Pairs}_1^{(i)}$. This p-value is the probability that a *at least* one non-parental pair of potential parents has a value of $P(\mathrm{C}_{\mathrm{Se}}^{\mathrm{Se}}|\boldsymbol{a})$ with kid i that exceeds $P^{(1)}$. In typical implementations of likelihood based parentage (e.g. CERVUS) this probability is computed by repeatedly simulating genotypes non-parental to kid i of all possible parent pairs in a simulated parent data base which is the same size as the actual data base and recording whether $P(\mathrm{C}_{\mathrm{Se}}^{\mathrm{Se}}|\boldsymbol{a})$ exceeds $P^{(1)}$ for any (simulated non-parental) pair. This is computationally very demanding for large scale problems. In our approach, because we can simulate genotypes for trios conditional on $\boldsymbol{v}^{(L)}(\boldsymbol{a}^*) \leq \boldsymbol{v}^{(L)\max}$, we need only focus on simulating a number of parent pairs equal to the number of pairs non excluded by Mendelian incompatibility with kid i. The procedure for doing so is as follows

1. Initialize variables.
    - Initialize EXCEED = 0.
    - Recall that $N^{(i)}$ is the number of possible parent pairs having fewer than $\boldsymbol{v}^{(L)\max}$ Mendelian incompatibilities with kid.
2. Perform the forward step calculations.
    - For each $r \in \mathscr{R}$ compute $P(\boldsymbol{v}^{(L)} \leq \boldsymbol{v}^{(L)\max}|r, g_{\text{kid }i})$
    - Note that this is done conditioning on the offspring genotype.
3. Compute the expected fraction of different trio types conditional on them having fewer than $\boldsymbol{v}^{(L)\max}$ Mendelian incompatibilities with kid i:
    - This involves a simple reweighting of π. Writing these expected fractions as π^*_r we have:

      $$\pi^*_r = \frac{\pi_r P(\boldsymbol{v}^{(L)} \leq \boldsymbol{v}^{(L)\max}|r, g_{\text{kid }i})}{\sum_{k\in\mathscr{R}} \pi_k P(\boldsymbol{v}^{(L)} \leq \boldsymbol{v}^{(L)\max}|r = k, g_{\text{kid }i})}$$

      for all r in $\mathscr{R}$.
4. Repeat the following steps REPS times:
    - Repeat the following $N^{(i)}$ times:
        - Simulate a relationship r^* from π^*
        - Using the backward algorithm, simulate the genotypes $\boldsymbol{a}^*$ of a trio from the distribution $P(\boldsymbol{a}|r^*, g_{\text{kid }i}, \boldsymbol{v}^{(L)}(\boldsymbol{a}^*) \leq \boldsymbol{v}^{(L)\max})$
        - Compute $P(\mathrm{C}^{\mathrm{Se}}_{\mathrm{Se}}|\boldsymbol{a}^*)$ (using π, not π^*) for this simulated genotype.
    - If any of the $N^{(i)}$ values of $P(\mathrm{C}^{\mathrm{Se}}_{\mathrm{Se}}|\boldsymbol{a}^*)$ exceeded $P^{(i)}$, add 1 to EXCEED.
5. At the end, EXCEED/REPS is a Monte Carlo estimate of the Type I error for assigning kid i to the ma and pa of $\text{Pairs}^{(i)}$.

Using p-values in the False Discovery Rate procedure

After computing p values as described above for every fish i in $\mathscr{O}$, we use the False Discovery Rate procedure (FDR) to control our rate of False Discoveries (*i.e.*, the fraction of offspring assigned to parent pairs that are assigned to nonparental pairs). Let m be the total number of offspring with $\boldsymbol{v}^{(L)} \leq \boldsymbol{v}^{(L)\max}$ for at least one pair of putative parents, and let $m_0 \leq m$ be the number of those offspring for whom pa and ma in $\text{Pairs}^{(1)}$ are not the true parental pair. Then, order these m offspring from smallest to largest p-value, letting (i) denote the offspring with the i smallest p-value, $p^{(i)}$. In their seminal paper, Benjamini & Hochberg (1995) showed that, in expectation, a false discovery rate less than α_{fdr} can be achieved by declaring parentage to offspring $(1), \dots, (k)$, where (k) is the largest value such that

$$p^{(i)} < \frac{i}{m}\alpha_{\text{fdr}}.$$

A more powerful approach is possible if the number m_0 is known or can be estimated. Benjamini & Hochberg (2000) provide an *ad hoc*, but general, graphically-inspired method for estimating m_0 which is employed in our software. Armed with an estimate of m_0 the FDR can be controlled by assigning parentage to offspring $(1), \dots, (k)$ where (k) is the largest value such that:

$$p^{(i)} < \frac{i}{m_0} \alpha_{\text{fdr}}.$$

The above expressions can be easily inverted to express the FDR as a function of $p^{(i)}$:

$$\alpha_{\text{fdr}} = p^{(i)} \frac{m_0}{i}.$$

This quantity is reported in the output of the software, so users can choose their own FDR.

Treatment of missing data and extension to multiple populations

There are many ways in which missing data might be handled in the above procedures. We have chosen a way that seems to give reasonable results without creating too much computational overhead. First, as described above, we can easily compute the probability of a trio while taking into account the missing data. For the forward step while assesssing p-values, however, we condition only on the missing data in the offspring. This is done via a straightforward side effect of the fact that we condition on the offspring genotype when doing the forward step for assessing p-values. In the backward step, we incorporate the occurrence of missing data in the members of the Pairs_i list by taking the trio genotypes simulated by the j^{th} iteration of the backward step for a particular replicate and masking the simulated genotypes by the pattern of missing data in $\text{Pairs}_i^{(j)}$. If there is a lot of missing data in any individual, we find that the genotype calls in that individual are often suspect so we have a missing data threshhold that the user may set in our software. If an individual has more missing data than this threshold (set by default to be 10 SNPs) then it is discarded from further consideration.

The extension to multiple populations of parents in the parent data base is also quite simple. If it is unknown *a priori* which population a collection of offspring came from then each offspring in that collection is compared to every parent from every population in the parent data base. Each individual i is assigned to the population that $\text{Pairs}_i^{(1)}$ belongs to, and then the analysis proceeds as before assuming that all $N^{(i)}$ pairs in Pairs_i are from the same population as $\text{Pairs}_i^{(1)}$, even if they were not. The false discovery rates are then accordingly computed as FDRs for the individuals non-excluded from a particular population.

Part II

Software Description and Documentation

INTRODUCTION

The method described in the previous part is implemented in the program SNPPIT which stands for "SNP Program for Intergenerational Tagging. This software can be downloaded from:
`http://swfsc.noaa.gov//staff.aspx?Division=FED&id=740`
The distribution of the software includes executable programs for PC and MacOS as well as the source code and compiling instructions. The distribution also includes this report.

This section of the report describes the use of the program, starting with a description of the input file format, then documenting how to run the program and the execution options, finally concluding with a description of the output files.

INPUT FILE FORMAT

SNPPIT takes a single text-based file as input. This file includes the genotypes of the parents in the data base and the offspring whose parentage is to be determined. It may also include several keywords which describe additional columns of information which can be included in the file. There may be multiple sets of possible parents (from different hatcheries, for example) and different sets of offspring (samples from different fisheries, for example). SNPPIT allows all those different sets of parents and offspring to be read in from single file.

Basic genotype data file format

The basic format of the data file is that of genetics "two-column" format with a list of locus names with estimated (or assumed) per-allele genotyping error rates at the top and some additional keywords like a GENEPOP file. The following is a simple example with only two individuals in each POP or OFFSPRING set.

```
NUMLOCI 4
MISSING_ALLELE -9
Locus1   0.003
Locus2   0.007
Locus3   0.005
Locus4   0.002
POP PopName1
FishA   101 102     104 105     102 102    101 101
FishB   102 102     104 104     102 103    101 100
POP PopName2
FishC   101 101     105 105     103 103    100 100
FishD   102 101     104 105     103 102    100 100
POP PopName3
FishE   101 102     104 104     102 103    100 101
FishF   102 102     104 105     103 103    101 101
```

```
OFFSPRING OffSpringCollectionName1  ?
FishG   101 102     104 105     102 103    101 100
FishH   102 102     105 105     102 102    101 100
OFFSPRING OffSpringCollectionName2  PopName1,PopName2
FishI   101 102     104 104     102 103    101 101
FishJ   102 102     105 105     102 102    101 100
OFFSPRING OffSpringCollectionName3  PopName3
FishK   101 102      -9  -9     102 103    101 100
FishL   101 102     105 104     103 103    101 101
```

The basic required keywords are:

NUMLOCI This must be the first string in the file and must be followed by the number of SNPs in the data set.

MISSING_ALLELE This must be given before the names of the loci and has to be followed by the string which is used to denote missing data. In this case that string is "-9".

POP This signifies the beginning of a collection of individuals who should all be treated as coming from a population with common allele frequencies. This might include individuals from the same population in different years if there is little allele frequency change expected between years. Following the POP signifier must be a single string (that includes no white space!) that is the name given to this population of individuals. This name must be unique (*i.e.*, it cannot be shared with another population). Note that all of the POP's must be given before any OFFSPRING.

OFFSPRING This signifies the beginning of a collection of individuals who are all possible offspring that are considered together as a group for some reason (perhaps they were all sampled in the same location, or it is known that their parents must all be from a particular population) and should be analyzed together as a single group of offspring. Following the OFFSPRING keyword is the name of this collection of offspring. This name should be unique. Following that there must be a single string with no white space in it that indicates the population from which potential parents of the individuals in this offspring collection might have descended. If these individuals could be the offspring of parents from any of the populations, then a "?" is appropriate (as used for `OffSpringCollectionName1`). Otherwise, the names of the populations that their parents could belong to should be given in a comma-separated list (with no white space!). The given pop names must match one of the population names given with a previously issued POP keyword.

Each row beneath a POP or an OFFSPRING keyword should be the genotype of an individual. This row must start with an individual identifier which can be any string of (non-whitespace) letters or numbers or text characters less than 100 characters in length. I recommend a word that includes some non-numerical characters as this makes it easy for the program to test to make sure it is scanning an individual identifier when it expects to be doing so. The identifier for any individual should be unique.

Following the individual identifier there may be a variable number of optional columns (described in "Optional Columns" below). Following those extra columns there must be 2 × NUMLOCI columns (delimited by any amount of white space) that give the genotypes. The type of each allele must be described by an integer between 0 and 999. A useful convention has the nucleotide bases assigned numbers in alphabetical order (A=1, C=2, G=3, T=4), but any integers between 0 and 999 inclusive will work. Missing data can be represented by any string or number (other than keywords like POP). If there is one allele missing at a locus the other allele must also be considered missing. If not, then the program will force the remaining allele to be missing and will issue a warning to the user, but will not issue an error. To change the string that is used to describe missing data, the MISSING_ALLELE keyword can be used anywhere after the NUMLOCI X line and the first locus name. For example, if the data file started like:

```
NUMLOCI 4
MISSING_ALLELE *
Locus1      0.002
Locus2      0.004
...
```

Then the string * would denote missing data at an allele. Note that regardless of what string you use to denote missing data, that string must be given twice per locus, because there are two alleles at each locus to denote as missing. Other good choices for MISSING_ALLELE might be ? or NA, etc. Whatever string is used to denote missing data must be used for missing data throughout the data set.

The locus names can include any characters except for white space (but don't be goofball and use commas in the locus name, since that will make it difficult to parse some of the output!). Therefore, please remove any spaces or tabs from your locus names before including them in the data set. The genotyping error rates that follow the locus names are required. Often these genotyping error rates will not be known with great certainty. However the experience in several labs suggests that with SNPs in salmon the genotyping error rates are less than 1% per locus. This is a reasonable value to use. SNPPIT, however, expects the genotyping error rate to be entered in terms of a rate *per gene copy* (or *per allele*). This is $\frac{1}{2}$ the per-locus rate. Therefore a 1% per-locus rate is entered as 0.005 to SNPPIT.

Optional columns in the data set

It will typically be useful to include additional data (like age data or spawning date data) for each individual. For this we have a series of different keywords which may be issued anywhere between the NUMLOCI X line and the first locus name which indicate the additional number of optional columns that are expected between the individual identifier and the first allele column of an individual. Since individuals in the OFFSPRING category will likely be endowed with different types of data than individuals in the POP category, optional columns are specified separately for each type of individual using the obviously named keywords below. The keywords which add optional columns to the POP individual rows are:

POPCOLUMN_SEX This specifies a column for POP individuals that identifies the sex of the individual. The possible values in the column are `M` for male or `F` for female (both uppercase), or `?` for individuals of unknown sex.

POPCOLUMN_REPRO_YEARS This specifies a column which gives the years in which an individual produced offspring (or spawned or mated or could have produced offspring, etc.). Values for this column are year ranges given in a format similar to that of the `cut` utility in UNIX. For example, `1947,1953,1955-1960,1965` (note that there is no white space anywhere in that sequence of numbers) indicates that the individual produced offspring (or spawned or mated or could have produced offspring, etc.) in the years 1947, 1953, 1955 through 1960, inclusive, and 1965. Other examples include: `1946` or `1946-1953` or `1965,1966` or even `1965-1966` (the latter two signify the same years). When dealing with hatchery-spawned chinook, typically there will only be a single year, of course; but the functionality for multiple year spawners is included for steelhead and other organisms. Note that unlike the `cut` utility, any numbers appearing in the above date range *must* be in ascending order. Otherwise the program will exit with an error. If reproduction-year data are unknown for an individual, then a `?` can be put in this column and the individual will be assumed to be a possible parent for an offspring of any age.

POPCOLUMN_SPAWN_GROUP This column is particularly useful for organisms like hatchery salmon which might be spawned together in groups. The value of this column may be any string up to 500 characters in length. It might, for example, indicate the date upon which an individual was spawned. Examples of possible values are: "1", or "1/27/04", or "1/27/04-Batch1", etc. The only restriction is that it must be less than 500 characters in length and *it cannot include any white space.* Any individuals within the same spawning group, in the same population, and spawning in the same years, are considered to be possible mates of any other individuals in the spawning group. If the spawning group of an individual is not known, then a `?` must be used. In this case, the individual will be considered to be a possible mate of any other individual in the population who spawned in the same year as the individual with a `?` for his/her spawner group.

The optional extra columns for the OFFSPRING individuals are specified with these keywords:

OFFSPRINGCOLUMN_BORN_YEAR The value here is the year the individual was born if known. This should be adjusted for season and gestation period in such a way that if a child was born in 1955, then it could have been the offspring of a parent with a REPRO_YEAR of 1955. So, it is called BORN_YEAR, but it really means "year in which it was conceived" so that it corresponds directly to years given in the POPCOLUMN_REPRO_YEARS column. Note that if there is some uncertainty about the year an individual was born, this can be captured by giving a range of years, in exactly the format described for POPCOLUMN_REPRO_YEARS.

OFFSPRINGCOLUMN_SAMPLE_YEAR The year the offspring was sampled. Values for this column are just integers giving the year, or `?`

OFFSPRINGCOLUMN_AGE_AT_SAMPLING This is just the range of possible ages for an individual given as for POPCOLUMN_REPRO_YEARS. Note that OFFSPRINGCOL-

UMN_SAMPLE_YEAR and OFFSPRINGCOLUMN_AGE_AT_SAMPLING only really make sense when used together, and are just offered as a possibly more convenient way of expressing the possible birth-year information of an offspring individual. If there is an OFFSPRINGCOLUMN_BORN_YEAR, then that column will be used to garner the data about the birth year of the individual. Otherwise, the birth year of the individual will be computed from the information in the OFFSPRINGCOLUMN_AGE_AT_SAMPLING and OFFSPRINGCOLUMN_SAMPLE_YEAR columns. If only one of the OFFSPRINGCOLUMN_AGE_AT_SAMPLING and OFFSPRINGCOLUMN_SAMPLE_YEAR columns appear in the data set and there is no OFFSPRINGCOLUMN_BORN_YEAR the program will print a warning that all offspring individuals are assumed to have unknown birth years.

Finally, the order of the columns in the POP individual rows or the OFFSPRING individual rows is given by the order in which the keywords are given between the NUMLOCI X line and the first POP keyword.

Now, we could make our short sample data set look like

```
NUMLOCI 4
MISSING_ALLELE #
POPCOLUMN_SEX
POPCOLUMN_REPRO_YEARS
OFFSPRINGCOLUMN_BORN_YEAR
Locus1   0.003
Locus2   0.007
Locus3   0.005
Locus4   0.002
POP PopName1
FishA   M  1950  101 102     104 105     102 102    101 101
FishB   ?  1950  102 102     104 104     102 103    101 100
POP PopName2
FishC   F    ?        101 101     105 105     103 103    100 100
FishD   M    ?        102 101     104 105     103 102    100 100
FishD   M    ?        102 101     104 105     103 102    100 101
POP PopName3
FishE   ?    ?        101 102     104 104     102 103    100 101
FishF   ?    ?        102 102     104 105     103 103    101 101
OFFSPRING OffSpringCollectionName1  ?
FishG   1950     101 102     104 105     102 103    101 100
FishH   1950     102 102     105 105     102 102    101 100
OFFSPRING OffSpringCollectionName2  PopName1,PopName2
FishI   1950     101 102     104 104     102 103    101 101
FishJ   1950     102 102     105 105     102 102    101 100
OFFSPRING OffSpringCollectionName3  PopName3
FishK   1950     101 102       #   #     102 103    101 100
FishL   1950     101 102     105 104     103 103    101 101
```

Specifying $\boldsymbol{\pi}$ via spawner population sizes in the data set

The expected fraction of different trio relationship categories amongst randomly drawn trios from a population is not easily computed. I have developed a recursive algorithm that does this for a set of inputs that includes past spawning population sizes, the ratio of effective number of breeders to the census number, the age distribution of spawners in each year, *etc.* This is a rather difficult program to use, and, it appears that it is only important to get a reasonable "ballpark" estimate of $\boldsymbol{\pi}$. Therefore, for those conducting parentage in chinook salmon hatchery populations I have developed a simpler method of obtaining the $\boldsymbol{\pi}$ parameter for the program. This is done by specifying the "average" number of spawners per year in the hatchery. This average should be computed as the average over any years that are between 2 and 5 years before any of the years for which parents are included from the hatchery in the data set.

To use this option, you first have to include the keyword CHINOOK_AVE_POP_SIZE in the preamble of the data file, after NUMLOCI but before the first locus name. For example the preamble might then look like:

```
NUMLOCI 4
MISSING_ALLELE #
POPCOLUMN_SEX
POPCOLUMN_REPRO_YEARS
OFFSPRINGCOLUMN_BORN_YEAR
CHINOOK_AVE_POP_SIZE
Locus1   0.003
Locus2   0.007
...
```

If you have included the CHINOOK_AVE_POP_SIZE keyword, then for every population you *must* give an average number of spawners immediately following the population name and in exactly this format: "`ave_sz1300`". This would, for example, mean that the average number of spawners has been 1300. So, this just necessitates a simple change to the POP lines from something that looks like:

```
POP PopName2
```

to something that looks like:

```
POP PopName2  ave_sz450
```

If these population sizes are not give in the data file, then by default the program assumes that the average number of spawners is 1000. It is also possible to specify $\boldsymbol{\pi}$ directly on the command line, but this is an advanced option and is not documented here. If you would like to do so, please send email to eric.anderson@noaa.gov.

RUNNING SNPPIT

SNPPIT has been optimized to be run as a command line program in a terminal window (use the "Command Prompt" application in Windows). The easiest way to do so is to copy the executable program (`snppit` for MacOS and `snppit.exe` for Windows) to a directory, copy the input file (assume it is named `MyDataFile.txt`) to that directory, then navigate to that directory on the command line, and issue this command:

```
./snppit -f MyDataFile.txt
```

on MacOS, or

```
snppit.exe -f MyDataFile.txt
```

on Windows.

That is really all that needs to be done to run the program, but there are a few other options available to the program that we will cover here.

First, there is a `--dry-run` option that reads all of the data in but does not attempt any analysis. Especially with large data sets, it is worth running the data with the `--dry-run` option just to check to make sure that the data are all getting read correctly. After running with the `--dry-run` option it is good to check the data summaries that get written to the file `snppit_output_BasicDataSummary.txt` to make sure that the numbers of individuals read in all look correct, *etc.*.

Another option, `--max-par-miss` controls the number of missing (*i.e.*, ungenotyped) loci a putative parent can have before it is dropped from the analysis. For example, if you issue the option, `--max-par-miss 7`, then any putative parent with more than 7 missing loci will be dropped from consideration as a parent. The default value is 10 which is probably a little on the high side. In our experience, fish with more than about 5 missing loci tend to also have relatively unreliable genotypes, so it is doubtful whether their offspring could be assigned to them anyway. On the other hand, a putative parent with many missing data points has fewer loci remaining for good discrimination and exclusion of its *non*-offspring—so, you don't want to consider parents with a lot of missing data!

Another option can be used to override the population sizes given using the CHINOOK_AVE_POP_SIZE method: issuing the command `--psz-for-all J` will set the average pop size for all populations to `J` (`J` should be an integer, like 1300) regardless of the values set in the data file for them.

The `--mi-fnr` sets the desired false negative rate due to Mendelian incompatibility. Thus, if you issue `--mi-fnr 0.001` then, the program will choose a maximum number of allowable Mendelian incompatibilities $\boldsymbol{v}^{(L)\max}$ (see Part I) so that in none of the populations do you expect to exclude more than 1 out of 1000 correct parent pairs on the basis of Mendelian incompatibility. The default value is 0.005 , *i.e.*, one out of 200 parent pairs are expected to be incorrectly excluded on the basis of Mendelian incompatibility. If you set this parameter too low, then the Mendelian exclusion step will be less stringent and the number of non-excluded possible parent pairs may increase, which will increase running times.

There are a few other options to SNPPIT but they are not documented here. To find out what they are, run the program while issuing the `--help-full` command.

Just to be explicit, a suitable command line using the various options could look like:

RUNNING TIMES

To give a general idea of how long this should take: on simulated data with roughly 250,000 parents from 10 populations in the parent data base, with spawner groups of size 100, and inferring parentage for about 20,000 fish from a fishery, it requires about 1.5 or 2 hours to complete the run on a Mac with a 3 GHz processor.

EXAMPLE DATA SET

The distribution comes with an example data set named `ExampleDataFile1.txt` that includes 10 different parental populations named `ParentPool_0`...`ParentPool_9` and two different offspring collections. One is named `FisherySample_19`. It is a sample from a mixed fishery taken in year 19 (years here are from years of a simulation—you can use actual years, like 2004, for your own values!), consequently, the fish in it could have come from any of the 10 parental populations; therefore, immediately following its name is a `?`. The other offspring sample is named `InRiverSampleYear19_Pop_0or1`. It is a sample in which you can be reasonably sure that all the members of it are either from `ParentPool_0` or `ParentPool_1`. Thus, immediately following it name in the OFFSPRING-tagged line is the list of populations: `ParentPool_0,ParentPool_1`.

The preamble for `ExampleDataFile1.txt` reads like:

```
NUMLOCI 96
MISSING_ALLELE *
POPCOLUMN_SEX
POPCOLUMN_REPRO_YEARS
POPCOLUMN_SPAWN_GROUP
OFFSPRINGCOLUMN_SAMPLE_YEAR
OFFSPRINGCOLUMN_AGE_AT_SAMPLING
```

Accordingly, the POP individuals have a column for `M/F/?` to denote sex of the individual; a column giving the spawning year of the individual (in this case those values are a single one of a number between 14 and 17, inclusive); and a spawning group column. In this file spawning groups are just designated with numbers between 0 and 8 inclusive. The OFFSPRING individuals all have a column giving the year that they were sampled, which in this case is 19 for all of them; they also have a column giving the possible ages of the fish at sampling, which in this case is `2-5` for all the offspring—this simply means that any fish caught in the fishery (or in the in-river sample) could be between the ages of 2 and 5, inclusive.

You should study the file `ExampleDataFile1.txt` carefully and make sure that you understand its format completely before making your own data set.

UNDERSTANDING THE OUTPUT

Output to the screen

SNPPIT prints a small amount of information out the screen while it is running—just enough to let you know that it is doing something. The part that typically takes the longest is that of computing *p*-values by simulation. The screen fills with dots to represent progress. It should be self explanatory. When run on the `ExampleDataFile1.txt` using the command `snppit -f ExampleDataFile1.txt` the output to the screen looks like:

```
DATA HAVE BEEN READ.  SUMMARIES APPEAR IN:  snppit_output_BasicDataSummary.txt


COMPUTING AN APPROPRIATE S-MAX
Compiling trio type probabilities for 10 parental collections
0.........
Performing Forward Step on 10 Collections of Trio Probabilities
0.........


EXCLUDING SINGLE PARENTS.  COLLECTION 1  FisherySample_19   with 161 indivs in collection.
Done with individual index:
0...................................................................................................
100............................................................
EXCLUDING SINGLE PARENTS.  COLLECTION 2  InRiverSampleYear19_Pop_0or1   with 18 indivs in collection.
Done with individual index:
0.................


FINDING NON EXCLUDED PARENT PAIRS.  COLLECTION 1  FisherySample_19   with 161 indivs in collection.
Done with individual index:
0...................................................................................................
100............................................................
FINDING NON EXCLUDED PARENT PAIRS.  COLLECTION 2  InRiverSampleYear19_Pop_0or1   with 18 indivs in collection.
Done with individual index:
0.................


COMPUTING THE FORWARD STEP AND PREPARING FOR BACKWARD STEP FOR ALL POPULATIONS
Compiling trio type probabilities for 10 parental collections
0.........
Performing Forward Step on 180 Collections of Trio Probabilities
0...................................................................................................
100...............................................................................


COMPUTING POSTERIORS:  COLLECTION 1  FisherySample_19      with 161 indivs in collection.
Done with individual index:
0...................................................................................................
100............................................................
COMPUTING POSTERIORS:  COLLECTION 2  InRiverSampleYear19_Pop_0or1      with 18 indivs in collection.
Done with individual index:
0.................


COMPUTING P-VALUES BY SIMULATION:  COLLECTION 1  FisherySample_19    with 161 indivs in collection
Done with individual index:
```

```
0...................................................................................................
100............................................................
COMPUTING P-VALUES BY SIMULATION:  COLLECTION 2  InRiverSampleYear19_Pop_Oor1    with 18 indivs in collection
Done with individual index:
0.................


PERFORMING FALSE DISCOVERY RATE CORRECTIONS


PRINTING FINAL PARENTAGE REPORT


SNPPIT PROGRAM EXECUTION COMPLETED.

Output is in the following files:
snppit_output_ParentageAssignments.txt -- Main output file that gives false disovery rates for all offspring with the most likely parents
snppit_output_BasicDataSummary.txt -- Basic information about the data that got read in.
snppit_output_ChosenSMAXes.txt -- Information about the smax vectors used in the analysis.
snppit_output_FDR_Summary.txt -- Offspring assigned to parents in each population, ranked by false discovery rate.
snppit_output_PopSizesAnPiVectors.txt -- Information about the sizes of the populations and the expected fraction of different trios thereby implied.
snppit_output_TrioPosteriors.txt -- Posterior probs for all non-excluded parent pairs of every offspring in the data file.


Questions, etc.? Send them to eric.anderson@noaa.gov
```

Output files

Most of the important output from SNPPIT is directed to files that will be written in the directory where the program was run. These output files all start with `snppit_output_` and end with a `.txt` extension. These files and a description of each are as follows:

`snppit_output_ParentageAssignments.txt` — Main output file that gives false discovery rates for all offspring with the most likely parents.

`snppit_output_BasicDataSummary.txt` — Basic information about the data that got read in.

`snppit_output_ChosenSMAXes.txt` — Information about the smax vectors used in the analysis.

`snppit_output_FDR_Summary.txt` — Offspring assigned to parents in each population, ranked by false discovery rate.

`snppit_output_PopSizesAnPiVectors.txt` — Information about the sizes of the populations and the expected fraction of different trios thereby implied.

`snppit_output_TrioPosteriors.txt` — Posterior probs for all non-excluded (by Mendelian incompatibility) parent pairs of every offspring in the data file.

These files are all tab delimited txt files and can be imported into Excel, for example. The names of the columns are mostly self-explanatory, but the headings are defined here in alphabetical order, along with an explanation of which files they occur in:

FDC.est.to.pop (`FDR_Summary`) The estimated upper bound on the total number of false discoveries of parentage assignments to a particular population if you set your FDR cutoff just above this particular individual.

FDR (`FDR_Summary, ParentageAssignments`) The false discovery rate associated with accepting the current individual's parentage assignment but none of the individuals with higher p-values.

GivenPopSize (`PopSizesAnPiVectors`) The value of the pop size specified for this population.

Kid (`FDR_Summary, ParentageAssignments, TrioPosteriors`) A column of identifiers of kids.

KidMiss (`ParentageAssignments, TrioPosteriors`) The number of ungenotyped loci in the kid of a trio.

LOD (`ParentageAssignments, TrioPosteriors`) The natural logarithm of the likelihood of the parental trio hypothesis divided by the likelihood of the non parental hypothesis for a trio.

MI.Kid.Ma (`ParentageAssignments, TrioPosteriors`) The number of Mendelian incompatibilities between the kid and the ma in a trio.

MI.Kid.Pa (`ParentageAssignments, TrioPosteriors`) The number of Mendelian incompatibilities between the kid and the pa in a trio.

MI.Trio (`ParentageAssignments, TrioPosteriors`) The total number of Mendelian incompatibilities in a trio. This is $v_3^{(L)}$.

Ma (`FDR_Summary, ParentageAssignments, TrioPosteriors`) A column of identifiers for ma's.

MaMiss (`ParentageAssignments, TrioPosteriors`) The number of ungenotyped loci in the ma of a trio.

MaxP.Pr.Relat (`ParentageAssignments`) The trio relationship having highest posterior probability. See the "Flat Text Name" column of Table 2 for an explanation of symbols for trio relationship.

MendIncLoci (`ParentageAssignments`) A column holding a comma-separated list of the names of loci at which there were Mendelian incompatibilities at the inferred trio.

OffspCollection (`ParentageAssignments, TrioPosteriors`) The name of the offspring collection that the kid is from.

P.Pr.Max (`ParentageAssignments`) The posterior probability of the trio relationship having the highest posterior probability.

P.Pr.RRRRR (`ParentageAssignments, TrioPosteriors`) The posterior probability of trio relationship RRRRR. See the "Flat Text Name" column of Table 2 for an explanation of symbols for trio relationships.

Pa (`FDR_Summary, ParentageAssignments, TrioPosteriors`) A column of identifiers for pa's.

PaMiss (`ParentageAssignments, TrioPosteriors`) The number of ungenotyped loci in the pa of a trio.

PiVectorElements.... (`PopSizesAnPiVectors`) This sits atop a list of prior probabilities for the different possible trio relationships.

PopName (`FDR_Summary, ParentageAssignments, PopSizesAnPiVectors`) The name of the population that the ma and pa are from.

PopSizeSetBy (`PopSizesAnPiVectors`) The method by which the population sizes were set.

PopSizeUsedForPi (`PopSizesAnPiVectors`) The actual population size used to compute the $\boldsymbol{\pi}$ vector. This will typically be a little smaller than the GivenPopSize, owing to the fact that the $\boldsymbol{\pi}$ values have been precomputed for just a limited set of population sizes.

Pvalue (`FDR_Summary, ParentageAssignments`) The p value computed by simulation for a trio.

Rank (`TrioPosteriors`) For a given kid, this is the rank of the parent pair when ranked from largest to smallest posterior probability of being parental.

RankInFDR (`FDR_Summary`) Amongst all the kids assigned to parents within a given parental population, this is the rank of the individual when sorted from smallest to largest p-value (and hence also in the FDR).

SpawnYear (`ParentageAssignments`) The year in which the parent pair or a trio spawned. Note that in the case of multiple-year spawners, this is simply the earliest year in which both parents are known to have spawned—there could be other years when they both spawned! This is not an issue with semelparous species like chinook.

TotMaNonExc (`ParentageAssignments`) The total number of putative mothers that were not excluded by Mendelian incompatibility with the kid.

TotPaNonExc (`ParentageAssignments`) The total number of putative fathers that were not excluded by Mendelian incompatibility with the kid.

TotPairsNonExc (`ParentageAssignments`) The total number of putative parent pairs that were not excluded by Mendelian incompatibility with the kid.

TotUnkNonExc (`ParentageAssignments`) The total number of putative parents of unknown sex that were not excluded by Mendelian incompatibility with the kid.

Part III

Hatchery and Fishery Simulations

INTRODUCTION

Simulations were undertaken with the dual objectives of 1) ensuring that the SNPPIT software works correctly and can handle large inference problems, and 2) assessing how accurately we can expect to infer the parentage of hatchery fish should parentage based tagging (PBT) be undertaken on a very large scale. To parameterize these simulations, we used allele frequencies estimated from 10 different populations for the 96 SNPs that comprise the Southwest Fisheries Science Center's standardized panel of SNPs selected for utility in PBT and genetic stock identification (GSI) from California to Washington. The populations used in the simulation extend from the Sacramento River in the south to the Lower Columbia River in the north and are described in "Simulated Hatcheries and Allele Frequencies" below.

Pedigrees over multiple generations connecting the fish in these simulated hatcheries and genetic data upon those fish were generated using the program SPIP (Anderson & Dunham, 2005). A chosen fraction of the fish spawned at each hatchery were included in the parent data base. A fraction of their offspring were intercepted in simulated fisheries. These aspects of the simulations are described in "Genetic Simulation Procedures." The parentage of fish in these simulations was assessed using the program SNPPIT as described in the section "Analysis With SNPPIT" The outcome of this is reported in "Results."

SIMULATED HATCHERIES AND ALLELE FREQUENCIES

The SWFSC Molecular Ecology Team at the Fisheries Ecology Division has been actively involved in SNP discovery for PBT. Some 100 new SNP assays for chinook have been developed in our group. These were added to SNPs previously discovered by the PSC-funded GAPS SNP discovery efforts, and from this combined set of SNPs, 96 were chosen on the basis of ease of genotyping and on minor allele frequency in California populations. We chose SNPs primarily to have a high minor allele frequency (and thus be useful for PBT); however some weight was given to selecting SNPs that showed large allele frequency differences between certain stocks, and which, consequently, will be useful for genetic stock identification. The complete details of this panel of 96 SNPs will be given in a forthcoming publication (Clemento et al. in prep). Here, however, I summarize in Table 4 the frequency of SNPs from 10 populations that we use to parameterize our simulations for PBT accuracy assessment.

It is remarkable to note that, although these SNPs were selected largely for their utility for PBT in California, the final column in Table 4 shows that the power of these SNPs for PBT is still quite high farther to the north, for example in the Lower Columbia populations and in the Chetco.

Table 4: Population allele frequencies used in simulations. Columns headed by $[a, b)$ give the number of SNPs found in the population at which the estimated minor allele frequency (frequency of the least frequent allele in the population) is between a and b. The FPR is the expected rate of false positives for trios in the C_U^U category, calculated at a level with a false negative rate of 0.05. This is a convenient measure of the power for parentage inference. See Anderson & Garza (2006) for details. The "Simulated Size" refers to how large the simulated hatchery endowed with these allele frequencies was. The sizes are defined in the section "Genetic Simulation Procedures"

#	Population Name	Abbreviation	Simulated Size	$[0.0, 0.1)$	$[0.1, 0.2)$	$[0.2, 0.3)$	$[0.3, 0.4)$	$[0.4, 0.5]$	FPR
1	Feather R Hatchery Fall	FRHfall	Large	13	15	22	20	27	3.75e-12
2	Kalama R Hatchery	Kalama	Large	15	13	19	22	28	3.80e-12
3	Cowlitz R Hatchery	Cowlitz	Large	14	17	15	30	21	8.47e-12
4	Feather R Hatchery Spring	FRHsp	Medium	10	16	19	25	27	2.00e-12
5	Klamath R Iron Gate Hatchery	KlamIGH	Medium	22	16	15	21	23	4.64e-11
6	Trinitry R Hatchery Fall	TrinityH	Medium	23	19	18	17	20	6.21e-11
7	Rogue R Spring Run	RogueSp	Medium	15	13	24	23	22	4.25e-12
8	Upper Sacramento R Late Fall	UpSacLF	Small	13	16	23	26	19	5.01e-12
9	Sacramento R Winter Run	SacWinter	Small	26	17	13	25	16	2.19e-10
10	Chetco R	Chetco	Small	10	16	27	26	18	3.08e-12

GENETIC SIMULATION PROCEDURES

Chinook Demography

For our simulations of hatchery chinook, we used an "average" chinook life history. The elements of this life history are as follows, each aspect being named by the SPIP command used to generate it:

[`-A 5`] The maximum age of any individual is 5 years. There are no spawners of age 6 years or older.

[`--fem-prob-repro 0 0 .3 .6 1.0`] Females of age 3 during the spawning season have a 30% probability of spawning. 4 year olds have a 60% probability of spawning, and any remaining females of age 5 have a 100% probability of spawning.

[`--male-prob-repro 0 0.05 .45 .61 1.0`] A small fraction (5%) of age-2 males spawn as jacks each year. Males of age 3 during the spawning season have a 45% probability of spawning. 4 year olds have a 61% probability of spawning, and any remaining males of age 5 have a 100% probability of spawning.

[`--fem-postrep-die 1 1 1 1 1`] All females die after spawning.

[`--male-postrep-die 1 1 1 1 1`] All males die after spawning.

[`-f 0 0 .5 1 1`] On average, females of age 3 produce only half as many offspring as do females of age 4 or 5.

[`-m 0 .5 1 1 1`] On average males spawned at age 2 as jacks have only half the fitness as do males spawned at age 3, 4, or 5.

[`--fem-rep-disp-par .25`] Within any age group of females, the ratio of the expected number of offspring to the variance in the number of offspring is .25. This corresponds roughly to creating a ratio of effective number of female spawners (of a given age) to the actual number of female spawners (of a given age) equal to 1/4. This is in the range of N_e/N ratios computed for salmon populations. This is important to model, since it increases the fraction of related, but non-parental, trios in the population.

[`--male-rep-disp-par .25`] As with females, the effective number of male spawners of a given age class is roughly one quarter of the actual number of male spawners.

Simulated Hatchery Sizes

To reflect a range of hatchery sizes, as indicated in Table 4, each set of allele frequencies was simulated in a population that was either Large, Medium, or Small in size. The sizes of these simulated hatcheries are expressed in terms of the average number of male and female spawners in each generation shown in Table 5. The Large hatcheries correspond to a handful of very large

Table 5: Average number of male and female spawners under the Large, Medium, and Small hatchery scenarios.

Sex and Age	Large	Medium	Small
Male 2 yrs	336	135	31
Male 3 yrs	2,594	1,038	253
Male 4 yrs	1,767	696	174
Male 5 yrs	999	401	97
Female 3 yrs	1,823	723	182
Female 4 yrs	2,314	915	230
Female 5 yrs	1,373	541	136
Total Male	5,696	2,270	555
Total Female	5,510	2,179	548
Total Male+Female	11,206	4,449	1,103

hatcheries, such as the fall chinook programs at Feather River Hatchery or Coleman National Fish Hatchery. The Medium hatcheries are meant to represent smaller scale hatcheries which still produce in the several millions of smolts each year, for example Feather River Hatchery's spring chinook program, or the Trinity River fall chinook hatchery program. Finally, the Small hatcheries mimic small programs propagating possibly endangered or threatened runs like the Sacramento River Winter Run program at Coleman National Fish Hatchery.

Spawning, Fishery, and Genetic Sampling

All individuals in the spawning pool each year were mated together randomly according to two different mating policies. The first was full-factorial mating in spawner groups of four males and four females (SG4). Under this scenario, there are four females and four males in each spawning group, and all 16 possible matings are done. The second scenario was one-by-one mating (SG1), in which each female was mated with exactly 1 randomly selected male, and no male was spawned with more than one female.

Regardless of the mating policy, the genotypes of the spawned fish were aggregated together in groups of approximately 100 males and 100 females that could possibly be mates with one another. This corresponds to a situation in which 100 males and 100 females are spawned each day, and the day of spawning is recorded for each individual. The term that has been used by us in the SWFSC for this type of data organization is "day-bucketing" referring to the practice of saving all the male and female tissue samples from each day in separate ethanol-filled containers. Thus, for any single "day" of spawning at a hatchery, there are $100 \times 100 = 10^4$ pairs of parents that must be considered as possible parent pairs in the parent data base.

The simulations were run for 21 years forward in time. This "warm-up" period allows the accrual of relatedness between members of the population which could, in theory, make it more difficult to correctly infer parentage. In years 14 to 17, a fraction of approximately G of the spawners were sampled. Values of G explored were 0.25, 0.5, and 1.0. Spawners were not sampled randomly; rather, to achieve $G = 1.0$, fish were sampled on all spawning days. For $G = 0.5$ spawners were sampled only every other day of spawning, and for $G = 0.25$ spawners were sampled only every fourth day of spawning. In this way, we ensured that the mates of any male (or female) in the parent data base would also be included in parent the data base.

Every year, males and females of age ≥ 2 were subjected to a 10% probability of being captured in a fishery, genotyped, and included in a fishery sample for that year. The fish sampled this way in spawning year 19 (*i.e.*, those fish that could be 2-year-olds born of parents that spawned at year 17, 3-year-olds born of parents spawned at year 16, 4-year-olds born of parents spawned at year 15, or 5-year-olds born of parents spawned at year 14) were included in our fishery sample. These were the fish whose parentage was assessed from parents in the parent data base. This represents the difficult case in which the fishery sample is a mixture of unknown proportions of fish from 10 different hatcheries. If PBT is successful here, than it will certainly perform well in inferring parents of fish returning to a particular hatchery when there is strong prior probability that the fish were produced at the hatchery to which they are returning.

Genotyping error was simulated by processing the data set once it was in SNPPIT format. I used a program written in C which changed the type of each SNP allele in the data set, independently, with probability 0.005. This corresponds to a per-locus genotyping error rate of about 1%—considerably higher than is believed to afflict most SNPs genotyped on a Fluidigm EP-1 platform (the genotyping platform being adopted as the standard choice for salmon studies). Since the simulated genotyping error rate was, perhaps, higher than what will ultimately be realized in real data, it is possible that greater accuracy of parentage assignments can be obtained in practice than reported here in the simulations.

Analysis With SNPPIT

For each replicate a data set was compiled in SNPPIT format that included both the parent data base and the fishery sample. With complete sampling ($G = 1.0$), the size of each of these files was about 100 Mb in text format. These data sets were analyzed by SNPPIT . With complete sampling ($G = 1.0$) SNPPIT required roughly 1.5 hours to analyze each replicate data set. For smaller values of G the size of the parent data base was smaller and each replicate took less time (roughly 30 minutes for $G = 0.5$ and 10 for $G = 0.25$). The output of SNPPIT was used to assign fish to parents so as to maintain a false discovery rate of less than 1 in 200. We say that our "desired" false discovery rate was less than 0.005. The results were compared with the simulated pedigrees and the accuracy of the parentage assignments was compared. Additionally, the number of individuals in the fishery with parents in the parent data base that were not included amongst the set of parentage assignments was recorded.

Table 6: Representative numbers of fish in the parent data base under the approximate sampling fractions G. These are rounded numbers from a single replicate simulation experiment. Variation between replicates was small. The total parent data base size is the number of parent genotypes from all possible years of spawners at all the hatcheries.

Hatchery Size	$G = 1.0$	$G = 0.5$	$G = 0.25$
Large	44,750	22,600	11,500
Medium	17,800	9,100	4,800
Small	4,400	2,400	1,600
Total Parent Data Base Size:	219,000	111,500	58,600

The `--mi-fnr` option of SNPPIT was set so that 0.005 was the expected fraction of offspring having so many Mendelian incompatibilities with their true parents that they would be discarded from further consideration on the basis of Mendelian incompatibility alone.

SNPPIT can take, as an advanced input, the fraction of trios, formed by randomly drawing individuals from the parental generation and from the fishery sample, expected to be of different relationship types (the types in $\mathscr{R}$ from Part I). These fractions were estimated by a simple recursive program (not described here) using the demographic parameters and observed numbers of spawners in the simulated hatchery each year. Thus, the probabilities π_r (for $r \in \mathscr{R}$) were estimated for Large, Medium, and Small hatcheries using data that will typically be available in hatchery programs (approximate number of spawners of different ages each year, average number of males mates per female spawned, approximate N_e/N ratio, *etc.*).

RESULTS

Number of Sampled Fish

The number of fish sampled for the parent data base and from the fishery varied little between replicate runs or between the different mating policies. Of course, the parent data base was smaller when a smaller fraction, G, of the parents was sampled. The fishery sample was always around 18,900 fish. Table 6 shows representative numbers of sampled fish from a single replicate under the different sampling fractions. These numbers represent, to my knowledge, the largest parentage inference exercises—simulated or real—attempted by any software. Previous tests on other available software programs showed that other programs are unable to handle data sets of this size (or even much much smaller, in some cases).

Accuracy of Parentage Assignments

In almost all simulations, the fraction of offspring assigned to parents that were assigned to the incorrect parents (*i.e.*, the false discovery rate) was less than 1 in 200. This is an excellent result, since individuals were assigned parentage so as to maintain a desired false discovery rate of less than 1 in 200. Additionally, in almost all cases the rate at which offspring whose parents were in the parent data base were not assigned parentage (the false negative rate) was less than 0.1. With 100% sampling of the parents, this false negative rate was appreciably lower. The results for mating policy SG1 are shown in Figure 2 and the results for mating policy SG4 are in Figure 3. It appears that accuracy was slightly reduced in the SG1 case relative to the SG4 case. This is likely due to the fact that under the SG1 scenario, all siblings are full siblings, whereas under the SG4 policy, only a quarter of siblings are full siblings, on average, and the remaining ones are half siblings. Full siblings of the true parents (or off the kid) in the parent data base may be misidentified as parents.

CONCLUSIONS AND DISCUSSION

It is apparent that the software is capable of handling large PBT scenarios with ease, and that a panel of 96 SNPs provides adequate power for inferring the parents of fishery samples. At the SWFSC we are currently undertaking the genotyping necessary to develop the parent data bases for select chinook hatcheries in California. It is also quite clear that the procedure for controlling the false discovery rate works in the sense that very few of the replicates had a higher rate of incorrect parentage assignments than the desired FDR of 0.005. However, in some cases, it appears that the actual false discovery rates imposed are conservative. If one is concerned primarily with avoiding incorrect parentage assignments, this is all right. However, if one is also concerned about the false negative rate (the rate at which the parents of offspring are not identified in the parent data base, even when they are in it), then the conservative feature of the FDR procedure employed here is unfortunate, since it increases the false negative rate.

One of the drawbacks of this method is that there does not seem to be an easily-implemented way to accurately estimate the false negative rate. This may become problematic in fisheries management contexts since some estimate of the false negative rate will be required in order to expand the samples to provide an estimate of total fishery impacts on any particular stock. The false negative rate could, of course, be estimated using simulations like those undertaken here, but it seems that a direct estimate may be preferable. In fact, future elaboration of this work in a fully-Bayesian framework may provide better estimates of the total fraction of fish from different stocks and year classes in a particular fishery. This might also ultimately use more of the information in the data and thus also provide more accurate parentage assignments. The implementation of such a scheme would not be trivial, especially while accounting for the multiple possible trio relationship categories and large parent data base size. However, the experience gained in this project will benefit any future endeavors in the Bayesian framework.

A further area that will benefit from extra work is in creating a joint analysis of parentage inference and genetic stock identification—individuals that can be identified to parent pair are assigned back to their parents, while those whose parents cannot be identified might still be assigned to the correct population. Such an inference problem leads to an interesting hierarchical mixture model

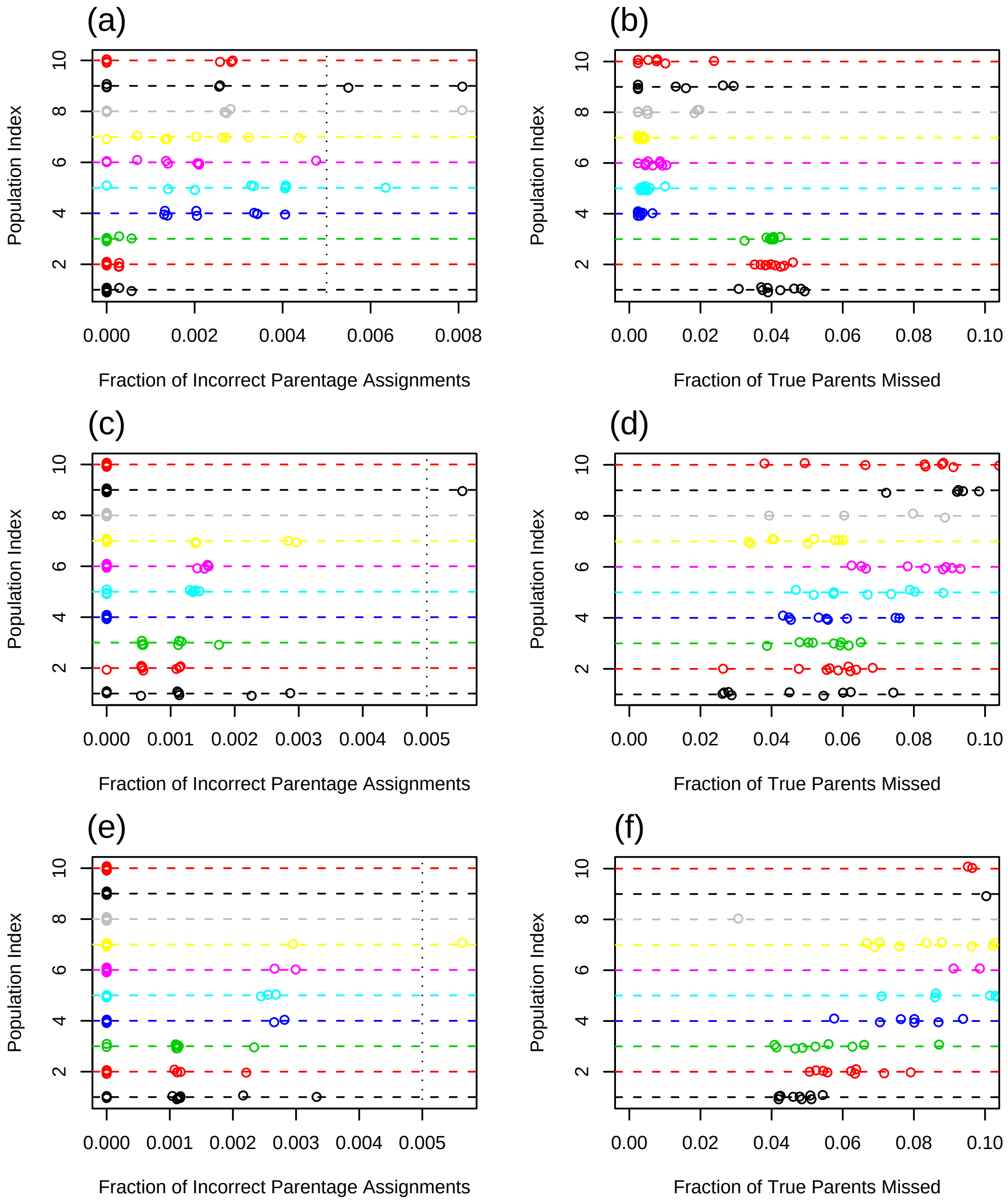

Figure 2: Results from SG1 mating policy. Left column of panels shows the true rate of false discovery (*i.e.*, true fraction of all parentage assignments that were incorrect); right column shows the fraction of offspring with parents in the data base that were not assigned to their parents (the false negative rate). Top row is for $G = 1.0$, middle is $G = 0.5$, and bottom is $G = 0.25$. Each dot in a plot is the result specific to one of the ten hatcheries in one of the replicate runs. Results for hatchery #1 (see first column in Table 4) are at a height of 1, and for hatchery 10 at a height of 10, with others in between.

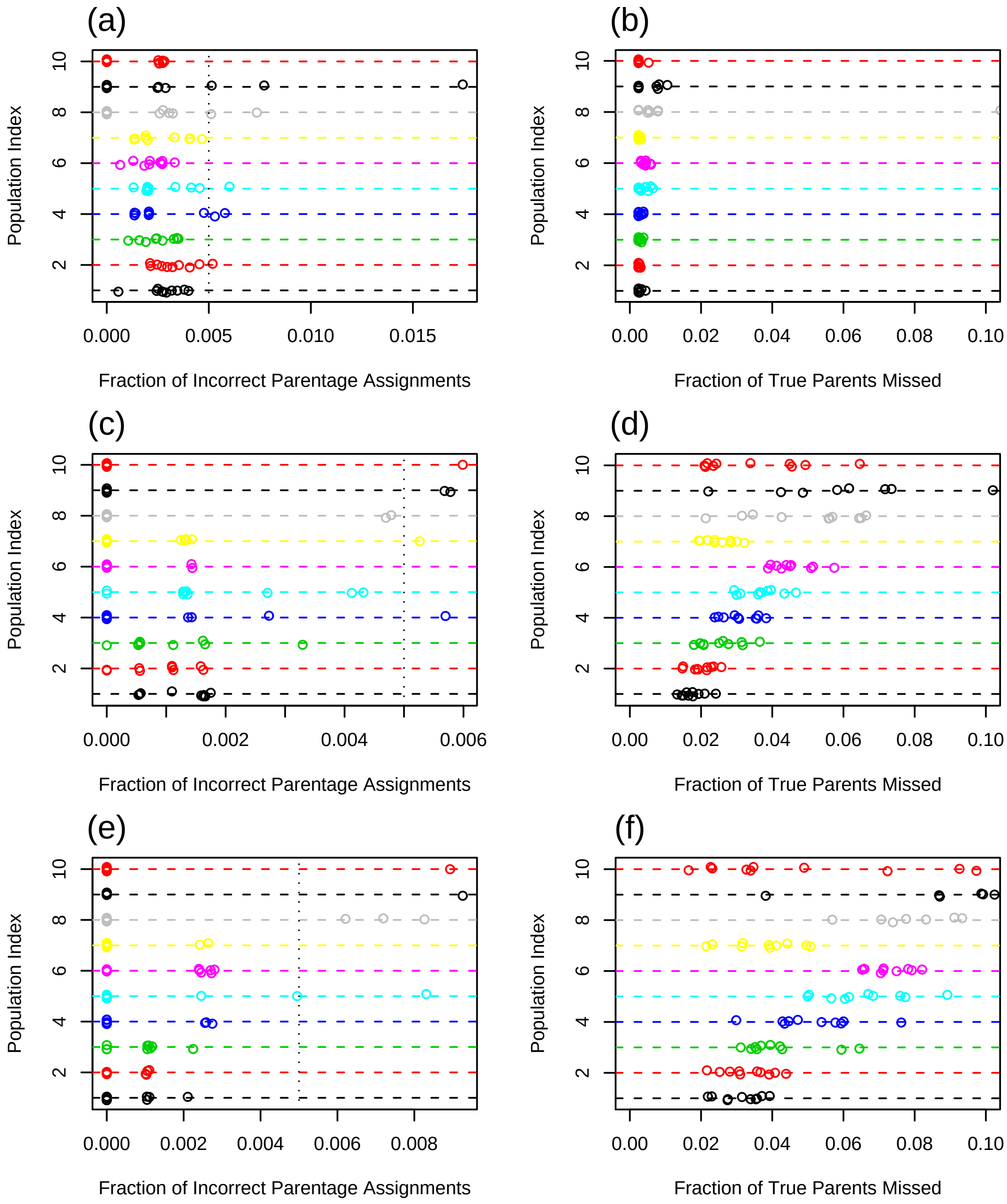

Figure 3: Results from SG4 mating policy. Left column of panels shows the true rate of false discovery (*i.e.*, true fraction of all parentage assignments that were incorrect); right column shows the fraction of offspring with parents in the data base that were not assigned to their parents (the false negative rate). Top row is for $G = 1.0$, middle is $G = 0.5$, and bottom is $G = 0.25$. Each dot in a plot is the result specific to one of the ten hatcheries in one of the replicate runs. Results for hatchery #1 (see first column in Table 4) are at a height of 1, and for hatchery 10 at a height of 10, with others in between.

and certain computational challenges, but it would allow the greatest leverage of genetic data for fisheries management. These are currently topics being investigated in my research group.

LITERATURE CITED

Anderson, E. C. & Dunham, K. K. 2005 SPIP 1.0: a program for simulating pedigrees and genetic data in age-structured populations. *Molecular Ecology Notes* **5**, 459–461.

Anderson, E. C. & Garza, J. C. 2006 The power of single nucleotide polymorphisms for large-scale parentage inference. *Genetics* **172**, 2567–2582.

Baum, L. E., Petrie, T., Soules, G., & Weiss, N. 1970 A maximization technique occurring in the statistical analysis of probabilistic functions on Markov chains. *Annals of Mathematical Statistics* **41**, 164–171.

Benjamini, Y. & Hochberg, Y. 1995 Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J Roy Stat Soc B Met* **57**, 289–300.

Benjamini, Y. & Hochberg, Y. 2000 On the adaptive control of the false discovery rate in multiple testing with independent statistics. *Journal of Educational and Behavioral Statistics* **25**, 60–83.

Cercueil, A., Bellemain, E., & Manel, S. 2002 PARENTE: computer program for parentage analysis. *J Hered* **93**, 458–9.

Duchesne, P., Godbout, M. H., & Bernatchez, L. 2002 PAPA (package for the analysis of parental allocation): a computer program for simulated and real parental allocation. *Molecular Ecology Notes* **2**, 191–193.

Elfstrom, C. M., Smith, C. T., & Seeb, J. E. 2006 Thirty-two single nucleotide polymorphism markers for high-throughput genotyping of sockeye salmon. *Molecular Ecology Notes* **6**, 1255–1259.

Fahrenkrug, S., Freking, B., Smith, T., Rohrer, G., & Keele, J. 2002 Single nucleotide polymorphism (SNP) discovery in porcine expressed genes. *Anim Genet* **33**, 186–195.

Hadfield, J. D., Richardson, D. S., & Burke, T. 2006 Towards unbiased parentage assignment: combining genetic, behavioural and spatial data in a Bayesian framework. *Molecular Ecology* **15**, 3715–30.

Hankin, D. G., Clark, J. H., Deriso, R. B., Garza, J. C., Morishima, G. S., Riddell, B. E., Schwarz, C., & Scott, J. B. 2005 Report of the expert panel on the future of the coded wire tag recovery program for Pacific salmon. Technical report, Pacific Salmon Commission.

Hayes, B. J., Nilsen, K., Berg, P. R., Grindflek, E., & Lien, S. 2007 SNP detection exploiting multiple sources of redundancy in large EST collections improves validation rates. *Bioinformatics* **23**, 1692–3.

Heaton, M. P., Harhay, G. P., Bennett, G. L., Stone, R. T., Grosse, W. M., Casas, E., Keele, J. W., Smith, T. P., Chitko-McKown, C. G., & Laegreid, W. W. 2002 Selection and use of SNP markers for animal identification and paternity analysis in U.S. beef cattle. *Mammalian Genome* **13**, 272–81.

Kalinowski, S. T., Taper, M. L., & Marshall, T. C. 2007 Revising how the computer program CERVUS accommodates genotyping error increases success in paternity assignment. *Mol Ecol* **16**, 1099–1106.

Marshall, T. C., Slate, J., Kruuk, L. E. B., & Pemberton, J. M. 1998 Statistical confidence for likelihood-based paternity inference in natural populations. *Molecular Ecology* **7**, 639–655.

Meagher, T. R. & Thompson, E. A. 1986 The relationship between single parent and parent pair likelihoods in genealogy reconstruction. *Theoretical Population Biology* **29**, 87–106.

Meagher, T. R. & Thompson, E. A. 1987 Analysis of parentage for naturally established seedlings within a population of *Chamaelirium luteum* (Liliaceae). *Ecology* **68**, 803–812.

Neff, B. D., Repka, J., & Gross, M. R. 2001 A Bayesian framework for parentage analysis: the value of genetic and other biological data. *Theor Popul Biol* **59**, 315–31.

Pemberton, J. M. 2008 Wild pedigrees: the way forward. *Proc Roy Soc B* **275**, 613–21.

Thompson, E. A. 1976a A restriction on the space of genetic relationships. *Annals of Human Genetics* **40**, 201–204.

Thompson, E. A. 1976b Inference of genealogical structure. *Social Science Information* **15**, 477–526.

Thompson, E. A. & Meagher, T. R. 1987 Parental and sib likelihoods in genealogy reconstruction. *Biometrics* **43**, 585–600.