# Inferring Ancestry in Admixed Populations using Microarray Probe Intensities

Chen-Ping Fu
Dept of Computer Science
University of North Carolina
Chapel Hill, NC 27599
ping@cs.unc.edu

Catherine E. Welsh
Dept of Computer Science
University of North Carolina
Chapel Hill, NC 27599
cwelsh@cs.unc.edu

Fernando Pardo-Manuel de Villena
Dept of Genetics
Lineberger Comprehensive Cancer Center
University of North Carolina
Chapel Hill, NC 27599
fernando@med.unc.edu

Leonard McMillan*
Dept of Computer Science
University of North Carolina
Chapel Hill, NC 27599
mcmillan@cs.unc.edu

## ABSTRACT

Numerous methods exist for inferring the ancestry mosaic of an admixed individual based on its genotypes and those of its ancestors. These methods rely on bialleic SNPs obtained from genotype calling algorithms, which classify each marker as belonging to one of four states (reference allele, alternate allele, heterozygous, or no call) based on probe hybridization intensity signals. We demonstrate that this conversion of probe intensities to discrete genotypes can lead to a loss of information and introduce errors via incorrect genotype calls.

We propose a method that directly infers ancestry from probe intensities by minimizing the intensity difference between a target individual and one or more of its ancestors. We demonstrate our method on mice from the developing Collaborative Cross (CC) genetic reference population, which are admixtures of a common set of eight ancestors. Our samples were genotyped using a 7.8K-marker Illumina Infinium platform called the Mouse Universal Genotyping Array (MUGA). We compare our reconstructions with a standard genotype-based method and validate our results using DNA sequencing data. Our algorithm is able to use information not captured by genotype calls and avoid errors due to incorrect calls.

## Categories and Subject Descriptors

J.3 [**Computer Applications**]: LIFE AND MEDICAL SCIENCES—*Biology and genetics*

*To whom correspondence should be addressed


*ACM-BCB* '12, October 7-10, 2012, Orlando, FL, USA


## General Terms

Algorithms

## Keywords

ancestry inference, probe intensities, dynamic programming

## 1. INTRODUCTION

The genomes of individuals from admixed populations are a mosaic of segments inherited from their ancestors. Mapping populations, in particular, are often derived from a set of inbred founders where each individual's genome is a mixture of founder haplotype segments. Ancestry inference on such an admixed individual refers to the problem of partitioning the individual's genome into haplotype blocks labeled with the contributing ancestor, with or without a given pedigree. The ability to infer ancestry accurately not only enables linkage and association mapping, but it also adds to our understanding of recombination.

Numerous methods have been proposed for inferring ancestry when given the genotypes of an individual and a set of ancestral haplotypes. Such methods generally use biallelic SNP data obtained from genotyping arrays or DNA sequencing as input. In humans, mapping ancestry is an essential step in admixture mapping, and methods such as HAPMIX [9], HAPAA [12], and LAMP [11] use HMM-based methods to infer the most likely ancestral blocks for each individual. While many methods require prior linkage disequilibrium analysis and use only unlinked markers, HAPMIX uses information from all neighboring markers and points out the amount of information lost by filtering linked markers. However, most of these methods accept genotypes from calling algorithms as ground truth and seldom discuss the artifact of calling errors, although LAMP does attempt to improve accuracy by analyzing sliding windows and taking a majority vote.

Algorithms for inferring ancestry in model organisms with known ancestors have also been proposed, such as HAPPY [8], a package for QTL mapping designed for outbred crosses. Methods for ancestry inference in recombinant inbred strains include two designed for the Collaborative Cross [6, 18], the

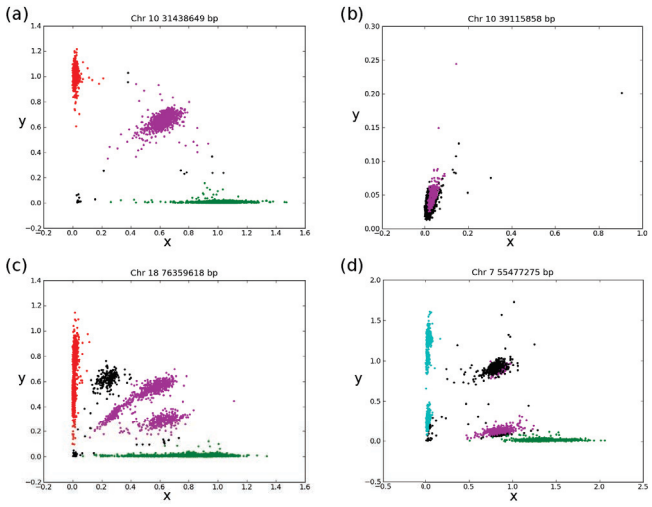


**Figure 1: Intensity plots of four markers, colored by genotype calls obtained from Illumina's GenomeStudio. Each point represents a single MUGA sample with its reference probe intensity on the x-axis and its alternate probe intensity on the y-axis. 'H' calls are colored magenta, 'N' calls are colored black, and the four nucleotides 'A', 'C', 'G', and 'T' are colored green, cyan, red, and blue, respectively. (a) A typical biallelic marker with two homozygous clusters and one heterozygous cluster. (b) A non-hybridizing marker with arbitrary 'H' calls. (c) A multiallelic SNP with several heterozygous clusters, one of which is uniformly called 'N'. (d) A multiallelic SNP with one heterozygous cluster alternately called both 'N' and 'H' due to batch effects in the calling algorithm.**

same strains upon which we test our algorithm [2]. GAIN [6], which was designed with the CC in mind, is an HMM-based algorithm that uses knowledge of the pedigree to efficiently infer ancestry probabilities. One assumption of GAIN and other existing methods is the use of high density genotypes. SNPs from high density arrays are often heavily filtered based on non-performing markers or questionable genotype calls. However, studies using low density arrays do not have the luxury of filtering out a significant percentage of SNPs and keeping only reliable genotype calls.

Moreover, even the best genotype calling algorithms often miscategorize markers with atypical hybridization intensity patterns [4]. In genotype calling, probe hybridization intensities are converted to one of four genotype calls (reference allele, alternate allele, heterozygous, or no call) via a classification algorithm. This is a difficult problem and calling algorithms can generate questionable results when marker intensities deviate from the expected biallelic intensity pattern. Furthermore, many markers have unexpected intensity patterns due to polymorphisms in or around the target probe sequences [4]. Sometimes sequence variations within probes lead to a reduction in hybridization intensities, and other times they manifest as intensity patterns that can discriminate between more than two alleles (Figure 1). In either case, traditional genotype calling methods that assume biallelic SNPs do not correctly classify these markers. This results in a loss of information, or worse, incorrect calls.

We propose an algorithm for ancestry inference that does not require the conversion of hybridization intensities to discrete genotypes. The use of hybridization intensities from genotyping arrays is common in studies of copy number variation (CNV) [15]. We show that allele variations beside CNVs manifest as variations in hybridization intensities as well, and we try to implicitly capture these variants with our methods.

Unlike existing methods, we directly use hybridization intensities in our model, which attempts to minimize the distance in 2D intensities from the target individual to its ancestors. We do not filter any markers, allowing each marker to be potentially informative on low-density genotyping arrays. We implemented our method on CC strains genotyped with the 7,854-marker Mouse Universal Genotyping Array [3]. Using available DNA sequencing data as ground truth, our algorithm compared favorably to GAIN, which is sensitive to incorrect genotype calls and loses information in atypical markers.

# 2. MATERIALS AND METHODS

## 2.1 Materials

We implemented our methods on an admixed population of eight inbred mouse strains known as the Collaborative Cross (CC) [2, 3]. The CC is an ongoing community effort to develop a large panel of recombinant inbred mice derived from a set of eight genetically diverse laboratory strains. The set of founders consist of five classical inbred strains (A/J, C57BL/6J, 129S1/SvImJ, NOD/ShiLtJ, NZO/H1LtJ) and three wild-derived strains (CAST/EiJ, PWK/PhJ, WSB/EiJ). They were chosen to capture a high level of genetic diversity, representing on average 90% of known genetic variation across all 1-Mb intervals [10]. A single CC strain is derived from the eight founders through a funnel breeding scheme that consists of two generations of mixing crosses, followed by 20 or more generations of inbreeding (Figure 2). We applied our methods to samples at various stages of the inbreeding process ranging from 4 generations of inbreeding, which is near the peak of genetic diversity (with a large number of founder segments and significant heterozygosity), to 22 generations of inbreeding, where samples are expected to be nearly completely inbred [16]. We implemented our algorithm on 461 CC mice from [3], focusing our results here on three lines with available sequence data.

To ascertain the founder contributions and level of inbreeding of CC lines, we designed the Mouse Universal Genotyping Array (MUGA), a 7,854-marker array based on the Illumina Infinium platform [3]. Markers were selected to locally discriminate amongst the eight inbred founders, so that in the ideal case, a set of four founders would have the alternate allele for each marker, and a maximum of four adjacent markers would be sufficient to uniquely identify the founder contributing to a homozygous genomic segment. Therefore, despite being a low-density array, the MUGA provides highly informative markers that are well-suited for mapping genome ancestry. Our methods make use of normalized probe intensities returned by Illumina rather than the genotype calls [14].

To establish statistical distributions for each marker, we genotyped a minimum of eight samples for each CC founder. These were primarily biological replicates with a few technical replicates. We also genotyped at least two replicates

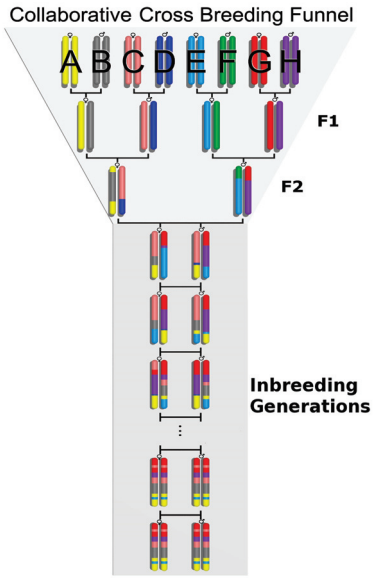


**Figure 2: Collaborative Cross breeding scheme. Each funnel has an ordered list of eight founders which are crossed for two generations, F1 and F2, then inbred for at least 20 generations to obtain recombinant inbred lines.**

of each of the 28 possible F1 combinations (ignoring the direction of the cross) for a total of 98 F1 samples. The founder samples were primarily male to provide models for each chromosome, whereas the F1 samples were primarily female to provide heterozygous models of X. We used these 65 founder samples and 98 F1 samples to learn a clustering model for each MUGA marker.

We have sequenced 3 of the the 461 CC strains that were genotyped using MUGA. Each sample has approximately 30X genomic coverage in the form of 100 bp, paired-end reads from an Illumina HiSeq 2000, with a mean fragment size of 300 bp. These data were aligned to a CC consensus genome using Bowtie 1.0 with the best-match criterion and allowance for 3 or fewer mismatches per read alignment. The CC consensus genome was formed by substituting the majority allele among the 8 founders into the NCBI Genome Reference Consortium Build 37 mouse reference genome [1] at the high confidence SNP positions as determined by the recent Welcome Trust mouse genome sequencing effort [5]. We used this aligned sequence data to validate the accuracy of our ancestry inference.

## 2.2 Algorithm overview

In contrast to genotype-based algorithms, we infer an ancestor mosaic directly from probe intensities to avoid problems introduced by limitations in genotype calling. Using intensities from replicate samples, we construct a 2D statistical model for each ancestor and find the set of ancestor intensities that best match the intensities of the target sample. We frame this problem as an optimization with penalties associated with making unnecessary transitions.

Given $n$ markers and $m$ inbred ancestors, our model minimizes the cumulative distance in 2D probe intensities from the individual sample to each of the $m' = m + \binom{m}{2}$ two-founder haplotype combination states ($m$ homozygous and $\binom{m}{2}$ heterozygous). Each of the $m'$ states has a representative cluster of probe intensities per marker that is pooled from the available founder and F1 replicates on MUGA. Transitions between different states are penalized via the addition of a transition penalty.

## 2.3 Creating reference clusters on MUGA

We have at least eight replicates of each CC founder on MUGA, as well as two or more replicates of each possible F1 combination. Each founder strain forms a repeatable 2D probe intensity cluster (Figure 3). To create reference clusters with increased statistical power, we pool together founders in common clusters (as determined by Hotelling's T-square test with a p-value threshold of 0.001) and estimate each final cluster's mean and covariance. In a second pass, we incorporate F1 samples. When the parental strains of the F1 map to a common cluster, we incorporate the F1 sample into the existing cluster model. When the parental strains of the F1 map to different clusters, we create a new heterozygous cluster model (Figure 3). We do not specify an expected number of alleles (clusters) prior to creating reference clusters, allowing for multiple homozygous alleles, each associated with one or more inbred founders. In extreme cases, a poorly performing marker might map all samples to a single cluster. Our model handles this case transparently, whereas traditional genotype calling makes arbitrary calls that are likely erroneous.

## 2.4 Distance model

Our goal is to assign the set of most likely ancestor states $\{f_1, f_2 ... f_i ... f_n\}$ for each marker at position $i$. The set of possible ancestor states $F$ contains $m' = m + \binom{m}{2}$ possible haplotype combinations given $m$ ancestral haplotypes[1]. At marker $i$, each ancestor state $f \in F$ has a cluster model $cluster(f, i)$ with a stored mean and covariance. We define the distance at each marker $i$ from the target sample to each ancestor state $f$ as the Mahalanobis distance [7] of the sample's 2D probe intensities to $cluster(f, i)$ . Our goal is to find the set of ancestor intensities that best models the target sample's intensities across the genome without excessive transitioning. Hence, denoting the target sample's 2D intensity vector as $x_i$ and the assigned ancestor as $f_i$ at marker $i$, we wish to minimize

$$D_M(x_1, f_1) + \sum_{i=2}^{n} D_M(x_i, cluster(f_i, i)) + penalty(f_{i-1}, f_i), \tag{1}$$

where $D_M(x_i, cluster(f_i, i))$ is the Mahalanobis distance from the 2D point $x_i$ to the reference cluster of $f_i$ at position $i$, and $penalty(f_{i-1}, f_i)$ is the transition penalty from the assigned state at marker i-1 to the state at marker i, defined below.

We set up a dynamic program to find the path which minimizes $dist_{p_1,q_n}$, the distance from state $p$ at the first marker to state $q$ at the last marker. The main dynamic programming recurrence then becomes

$$\begin{aligned} dist_{f_i=p, f_{i+1}=q} =& D_M(x_i, cluster(q, i)) + penalty(p, q) \\ &+ min\{dist_{f_0=r, f_i=p} | \forall r \in F\}. \end{aligned} \tag{2}$$

[1]With the exception of sex chromosomes on male samples, which only have m states.

**Table 1: Transitions between different states $p$ and $q$**

| $p$ is homozygous | $q$ is homozygous | $p$ and $q$ share a haplotype | Graphical depiction | $penalty(p, q)$ |
|---|---|---|---|---|
| yes | yes | no |  | mean $D_M$ between different homozygous clusters |
| yes/no | no/yes | yes |  | 1.5* mean $D_M$ between homozygous and heterozygous clusters |
| no | no | yes |  | 1.5* mean $D_M$ between different heterozygous clusters |
| yes/no | no/yes | no |  | 5.0*mean $D_M$ between homozygous and heterozygous clusters |
| no | no | no |  | 5.0*mean $D_M$ between different heterozygous clusters |

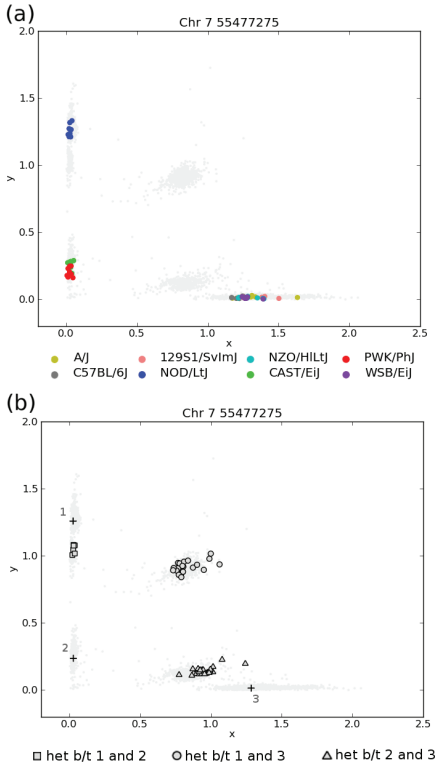


**Figure 3: Creating reference clusters. (a)To create homozygous reference clusters, we first pool all inbred founders with overlapping clusters as determined by replicate samples. The SNP shown here has three homozygous clusters. (b) We then create heterozygous reference clusters by pooling F1 samples between founders in different clusters. This SNP has three heterozygous clusters. We also refine homozygous clusters by adding F1 samples between founders in the same homozygous clusters. The means of homozygous clusters 1, 2, and 3 are shown as crosses. Data points for all samples are shown in the background to provide context. They are not used in the cluster modeling.**

Since our algorithm does not require knowledge of pedigree, transition penalties are based on observed differences in probe intensities rather than expected recombination frequency. Using our predetermined founder and F1 clusters, we calculate the mean Mahalanobis distance from homozygous clusters to other homozygous clusters, from heterozygous clusters to homozygous clusters, and from heterozygous clusters to other heterozygous clusters. Using these mean Mahalanobis distance values, we allow for transitions between homozygous states when we encounter a single SNP with typical Mahalanobis distance between two different homozygous clusters. Since heterozygous clusters typically have a smaller distance to all other clusters, the penalty to transitioning to or from heterozygous states is equivalent to 1.5 times the typical Mahalanobis distance between heterozygous states and other states. Transitions that suggest two independent recombination events at the same locus (coincident transitions) are rare and are penalized more heavily in our model. We set this penalty to be five times the mean Mahalanobis distance between different states. The set of possible transitions between state $p$ and state $q$, where $p \neq q$, are shown in Table 1. Transition penalties are symmetric, and there is no penalty value for staying in the same state, that is, $penalty(p, q) = penalty(q, p)$ and $penalty(p, p) = 0$.

For our CC dataset genotyped on MUGA, the penalty values are 0.082 between different homozygous states, 0.066 between heterozygous and compatible homozygous states, and 0.047 between compatible heterozygous states. Coincident transitions have penalty values of 0.22 and 0.16.

## 2.5 Refining recombination breakpoints

Determining recombination breakpoints between founders that share similar or identical sequences near transitions is a challenge. In this case, although the dynamic programming algorithm will specify a transition between some pair of adjacent markers, we report the breakpoint as an interval of ambiguity where the true breakpoint falls. On a transition from ancestor states $p$ to $q$, we start from the breakpoint assigned by the dynamic programming and extend the ambiguous interval both ways. We stop when we reach a left endpoint $i$ where $D_M(x_i, cluster(p, i)) < D_M(x_i, cluster(q, i))$ and $cluster(p, i) \neq cluster(q, i)$, as well as a right endpoint $j$ where $D_M(x_j, cluster(p, j)) > D_M(x_j, cluster(q, j))$ and $cluster(p, j) \neq cluster(q, j)$, where $x$ represents the target sample's intensities.

## 2.6 Funnel constraints

Assuming a founder order of $ABCDEFGH$ for a CC strain, heterozygous combinations of the initial founder mating pairs $AB$, $CD$, $EF$, and $GH$ cannot reappear in later generations, since the genomic material passed from an F1 cross is carried on a single haplotype in all subsequent generations [**?**, **?**]. When applying our algorithm to CC samples with available funnel information, we incorporate this constraint by removing these prohibited founder states.

# 3. RESULTS

## 3.1 Reference intensity clusters

We created reference clusters for 7,854 MUGA markers using a total of 65 CC founder samples and 98 CC F1 samples. The eight CC founders segregated into a single cluster

**Table 2: Number of SNPs that disagree between our algorithm vs. GAIN**

| | Total SNPs disagreeing between our algorithm and GAIN | SNPs where our algorithm agrees with sequence data | SNPS where GAIN agrees with sequence data |
|---|---|---|---|
| OR867m532 | 33,026 | 24,092 | 8,934 |
| OR1237m224 | 17,536 | 14,524 | 3,011 |
| OR3067m352 | 38,621 | 23,095 | 15,526 |
| Total | 89,183 | 52,144 (69.2%) | 27,471 (30.8%) |

for 1,104 markers. We observed the expected two homozygous clusters in 5,550 markers, with a reference allele cluster, an alternate allele cluster, and a single heterozygous cluster among our reference samples. The remaining 1,200 markers exhibit three or more clusters among the eight inbred founders, with 1,021 exhibiting three homozygous clusters, and 179 exhibiting four or more (Figure 4). The maximum number of homozygous clusters we observed was six.

Of the 6,750 informative markers, we have a mean of 2.21 homozygous clusters per SNP, or 3.66 clusters per SNP including heterozygous clusters. Thus, using our reference clusters, each SNP provides more information than a typical genotype call with 2 homozygous and 1 heterozygous alleles. This is especially advantageous for low-density platforms such as the MUGA and allows us to break ties between similar founders and refine recombination breakpoints, as discussed below.

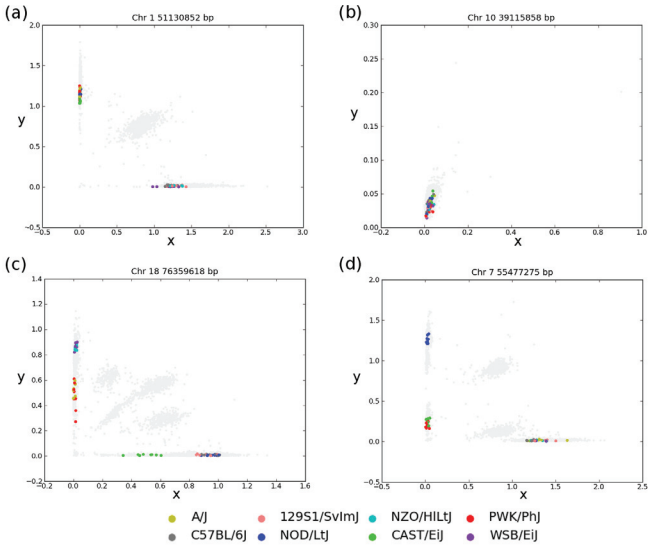


**Figure 4: Intensity plots with replicates of CC founders highlighted and all other samples drawn as background. Founders with overlapping clusters are pooled to create a single homozygous cluster. (a) A typical biallelic marker with the expected two homozygous clusters. (b) A poorly performing marker with a single cluster. (c) A marker with four homozygous clusters. (d) A marker with three homozygous clusters.**

## 3.2 Comparisons using sequencing data

Using available sequencing data as ground truth, we compared the predictions of our algorithm to those of GAIN, a genotype-based method optimized for animals with complex pedigrees such as CC animals [6]. GAIN uses knowledge of the breeding funnel and generations of inbreeding to approximate transition probabilities in a hidden Markov model. As with most genotype-based methods, GAIN infers heterozygous genotypes and requires genotypes from only the inbred founders. We used the consensus genotype calls given by Illumina's GenomeStudio software [14] from all samples of each CC founder. Since GAIN requires that all founders be called a homozygous allele at each marker, we filtered the 7,854 MUGA markers by eliminating all markers where a CC founder's consensus call was "H" or "N," as well as markers where all eight founders have the same call, leaving 5,782 markers. In comparison, our algorithm uses every marker. This includes 6,750 informative markers with more than one cluster, nearly 1000 markers more than the ones used by GAIN.

We ran our algorithm and GAIN on three CC samples with DNA sequencing data available: OR867m532, OR1237m224, and OR3067m352. We examined the non-ambiguous regions where the two methods disagree and imputed high-confidence SNPs from the Wellcome Trust Sanger Institute [5] for these regions based on the inferred ancestries. We then estimated the true genotypes by examining the aligned reads at each SNP locus, considering only loci with a coverage of ten or more reads. Loci where the second most common nucleotide showed up with a frequency of more than 0.2 were declared heterozygous.

Of all high-confidence Sanger SNPs in regions where the two methods disagree, 69.2% of SNPs imputed using ancestor assignments from our method agree with the sequencing data, compared to 30.8% of SNPs imputed from GAIN that agree with sequencing (Table 2). With the assumption that our aligned sequencing data and Sanger SNPs are correct, loci with imputed SNPs that differ from the sequencing data most likely result from erroneous ancestry assignments. As seen in the sample plot of chromosomes 3 and 5 on OR1237m532 (Figure 6), errors in GAIN are often driven by incorrect genotype calls, where a single miscalled genotype can result in an incorrect assignment. These incorrect genotype calls often occur in markers with intensity clusters that do not separate as well as typical biallelic intensity clusters, and the discretization from intensities to genotype calls in these cases easily lead to errors in algorithms relying on correct genotype calls.

Unlike genotype-based methods, our reference clusters can make use of markers where ancestors have "H" or "N" calls, and they can discriminate between ancestors with the same genotype call but have different hybridization intensity patterns. For example, our algorithm defines a recombination breakpoint between 15,059,945 bp and 15,922,708 bp on chromosome 17, where the ancestor of OR1237m224 transitions from homozygous WSB/EiJ (purple in Figure 5) to homozygous CAST/EiJ (green). GAIN reports the recombination breakpoint to be between 14,675,894 bp and 18,347,703

bp, a region 2.8Mb larger than that reported by our algorithm. From the pileups of our aligned sequencing reads, we can refine the true breakpoint to the 5Kb region centered around 15,060,000 bp. Our algorithm is able to more precisely discriminate the breakpoint region due to a marker with an "N" genotype call and a marker with three homozygous clusters flanking the breakpoint. GAIN does not consider the marker immediately upstream of the true breakpoint at 15,059,945 bp since four of the eight CC founders are called "N" at the locus, along with the target individual. However, WSB/EiJ and CAST/EiJ clearly segregate into separate clusters at the marker, with the target individual falling in WSB/EiJ's reference cluster, which is recognized by our algorithm. The marker downstream of the true breakpoint at 15,922,708 bp has three homozygous reference clusters, with WSB/EiJ and CAST/EiJ sharing the same genotype calls but segregating into different clusters. GAIN is unable to differentiate between the two founders at that marker due to their shared genotype call, but our algorithm is able to assign the target individual to CAST/EiJ's reference cluster (Figure 5).

## 3.3 Other platforms and populations

Our algorithm works across different genotyping platforms. In addition to testing on the MUGA, we have also tested our algorithm on CC animals genotyped with the 600K-marker Mouse Diversity Array (MDA) [17], a high-density genotyping array on the Affymetrix platform. Since we have fewer replicates of CC founders and F1s genotyped on the MDA, instead of creating reference clusters and calculating Mahalanobis distances, we used other distance measures, such as 2D Euclidean and Manhattan distances, to calculate distance between the individual sample and each ancestor. In the case of F1 strains without available samples, we approximate the intensities of the F1 by taking the mean intensities of its two parental strains. This approximation has given us results similar to those of using real F1 samples. Though results are not shown for the MDA due to space limitations, our algorithm outperforms GAIN on the MDA as well.

Although we have focused our results on samples from the CC, our algorithm has been implemented on other populations that have been genotyped on MUGA as well. We have tested our algorithm on heterogeneous stocks such as the Diversity Outbred (DO) population being developed at The Jackson Laboratory [13], as well as transgenic, knockout, and knockin mice from the Mutant Mouse Regional Resource Centers (MMRRC). These mice are derived from two or more ancestors. For an ancestor that is not a CC founder, we assign a most likely CC founder-derived reference cluster at each MUGA marker and run our algorithm with the most likely set of reference clusters representing the ancestor. Since the CC founders capture most genetic diversity in the mouse, the reference clusters we created using CC founders and F1s work well for modeling non-CC ancestors as well (Figure 7). An online tool implementing our algorithm on the MUGA can be found at www.csbio.unc.edu/CCStatus.

# 4. DISCUSSION

Existing methods for ancestry inference assume accurate genotype calls that capture all variance within a marker. However, markers may capture multiallelic information due to unexpected polymorphisms in the target probe sequence,

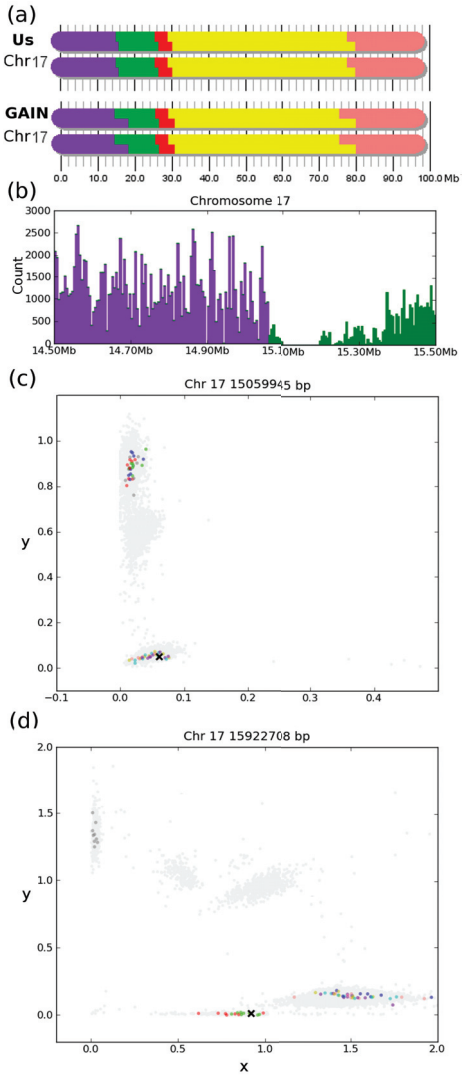


**Figure 5: Intensity information better resolves a recombination breakpoint on chromosome 17 of sample OR1237m224. (a) Assigned ancestry from our algorithm (top) and GAIN (bottom). Both algorithms show a recombination breakpoint between WSB/EiJ (purple) and CAST/EiJ (green) around 15Mb. Our algorithm shows the region containing the breakpoint as 15,059,945 - 15,922,708 bp, while GAIN shows the region as 14,675,894 - 18,347,703 bp. (b) Sequencing data pinpoints the breakpoint to a 5Kb region centered around 15.06 Mb. Here, we show the number of SNPs from aligned reads which are informative between WSB/EiJ and CAST/EiJ, colored by the SNP's allele. (c) The marker immediately upstream of the true breakpoint. CC founders are highlighted and OR1237m224 is marked as "x." This marker was filtered by GAIN due to the high number of 'N' calls among CC founders, but the sample falls within the cluster with the WSB/EiJ allele. (d) A marker downstream of the true breakpoint. WSB/EiJ and CAST/EiJ share the same genotype call at the marker, so GAIN cannot discriminate between the two. However, WSB/EiJ and CAST/EiJ fall in different reference clusters, so we can accurately assign the sample to the cluster containing CAST/EiJ.**

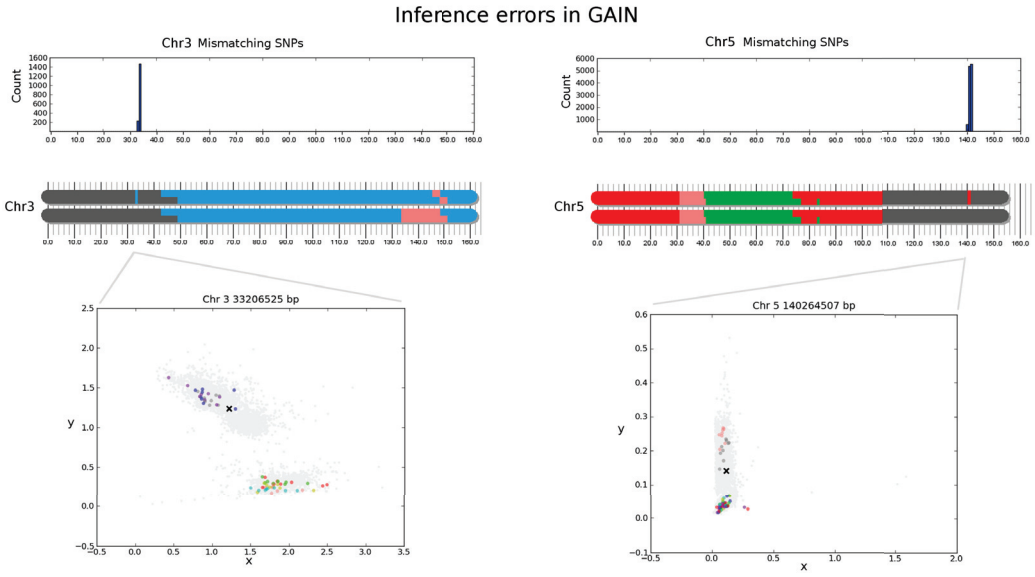


**Figure 6:** **Errors in GAIN are often due to questionable genotype calls. Here we show results from GAIN on chromosomes 3 (left) and 5 (right) of sample OR1237m224. The histograms on the top show the SNPs with alleles imputed from GAIN that differ from sequence data, out of all SNPs in regions where our assignments differ from GAIN's. This suggests the small heterozygous segments assigned by GAIN on both chromosomes are erroneous. GAIN's ancestry assignment is depicted in the middle, and the bottom plots show SNPs where the sample is called 'H'. In both chromosomes, the errors occur in regions of markers where the sample is called 'H' yet has an intensity vector close to the correct homozygous cluster (dark gray). CC inbred founders are highlighted in the intensity plots, and the intensity of OR1237m224 is marked "x."**

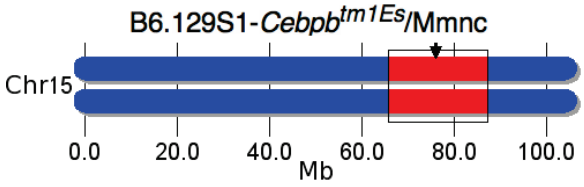


**Figure 7:** **The ancestry of a transgenic mouse from the Mutant Mouse Regional Resource Centers. This strain is homozygous on a C57BL/6J (blue) background, with a target mutation on chromosome 15 at 76.08Mb (denoted by the arrow) carried by an ES cell line derived from 129S6/SvEvTac (red). Our algorithm finds the region contributed by 129S6/SvEvTac (in box), which can be useful in predicting which SNPs are in linkage disequilibrium with the target allele.**

and we have observed a substantial number of markers in multiple genotyping platforms which consist of more than two homozygous intensity clusters. Our ancestry inference algorithm clusters ancestors based on probe intensities and solves a shortest distance optimization problem from the intensities of the target individual to those of a set of ancestors. By using probe intensities instead of discretized genotype calls, we obtain more information from multiallelic markers and markers with many "N" calls, and we eliminate errors due to incorrect genotype calls in markers with atypical intensity patterns.

Low-density arrays fundamentally limit the resolution of detectable ancestral segments due to the sparsity of markers. However, our algorithm still may not capture small ancestral segments that span just one or two markers, especially if intensity clusters are not as well-separated within these markers. Developing a marker-based penalty model based on the distances at each marker, instead of using the same penalties across all markers, would help us better resolve the correct ancestors in these regions.

We have demonstrated that probe hybridization intensities provide valuable information that is often lost after genotype calling. Although some perceive intensities as noisy data, our intensity-based ancestry inference produces good results that eliminate noise originating from incorrect genotype calls. Furthermore, we are able to specify recombination regions more precisely due to additional information from intensities. Intensity-based methods can be used to solve many other problems that traditionally rely on discretized genotype calls, such as haplotype phasing or calculating genomic similarity between strains. Using intensity-based methods eliminates the need for genotype calling, which is time-consuming and subject to errors. In cases where discretized genotype calls are desired, genotype calling algorithms that allow for an arbitrary number of alleles per marker could lead to more accurate results than would traditional biallelic calls.

## 5. ACKNOWLEDGEMENTS

We would like acknowledge Dr. Daniel Pomp from the University of North Carolina and Dr. George Weinstock and The Genome Institute at Washington University in St. Louis for generating the whole genome sequence data used in the validation experiments.

This work is supported by the Center of Excellence in

Genome Sciences (P50HG006582/P50MH090338) and The Carolina Center to Characterize and Maintain Mutant Mice (U42OD010924).

## 6. REFERENCES

[1] D.M. Church, L. Goodstadt, L.D.W. Hillier, M.C. Zody, S. Goldstein, X. She, C.J. Bult, R. Agarwala, J.L. Cherry, M. DiCuccio, et al. Lineage-specific biology revealed by a finished genome assembly of the mouse. *PLoS biology*, 7(5):e1000112, 2009.

[2] G.A. Churchill, D.C. Airey, H. Allayee, J.M. Angel, A.D. Attie, J. Beatty, W.D. Beavis, J.K. Belknap, B. Bennett, W. Berrettini, et al. The collaborative cross, a community resource for the genetic analysis of complex traits. *Nature genetics*, 36(11):1133–1137, 2004.

[3] Collaborative Cross Consortium. The genome architecture of the collaborative cross mouse genetic reference population. *Genetics*, 190:389–401, 2012.

[4] J.P. Didion, H. Yang, K. Sheppard, C.P. Fu, L. McMillan, F.P.M. de Villena, and G.A. Churchill. Discovery of novel variants in genotyping arrays improves genotype retention and reduces ascertainment bias. *BMC genomics*, 13(1):34, 2012.

[5] T.M. Keane, L. Goodstadt, P. Danecek, M.A. White, K. Wong, B. Yalcin, A. Heger, A. Agam, G. Slater, M. Goodson, et al. Mouse genomic variation and its effect on phenotypes and gene regulation. *Nature*, 477(7364):289–294, 2011.

[6] E.Y. Liu, Q. Zhang, L. McMillan, F.P.M. de Villena, and W. Wang. Efficient genome ancestry inference in complex pedigrees with inbreeding. *Bioinformatics*, 26(12):i199–i207, 2010.

[7] P.C. Mahalanobis. On the generalized distance in statistics. In *Proceedings of the National Institute of Sciences of India*, volume 2, pages 49–55. New Delhi, 1936.

[8] R. Mott, C.J. Talbot, M.G. Turri, A.C. Collins, and J. Flint. A method for fine mapping quantitative trait loci in outbred animal stocks. *Proceedings of the National Academy of Sciences*, 97(23):12649, 2000.

[9] A.L. Price, A. Tandon, N. Patterson, K.C. Barnes, N. Rafaels, I. Ruczinski, T.H. Beaty, R. Mathias, D. Reich, and S. Myers. Sensitive detection of chromosomal segments of distinct ancestry in admixed populations. *PLoS genetics*, 5(6):e1000519, 2009.

[10] A. Roberts, F. Pardo-Manuel de Villena, W. Wang, L. McMillan, and D.W. Threadgill. The polymorphism architecture of mouse genetic resources elucidated using genome-wide resequencing data: implications for qtl discovery and systems genetics. *Mammalian Genome*, 18(6):473–481, 2007.

[11] S. Sankararaman, S. Sridhar, G. Kimmel, and E. Halperin. Estimating local ancestry in admixed populations. *The American Journal of Human Genetics*, 82(2):290–303, 2008.

[12] A. Sundquist, E. Fratkin, C.B. Do, and S. Batzoglou. Effect of genetic divergence in identifying ancestral origin using hapaa. *Genome research*, 18(4):676–682, 2008.

[13] K.L. Svenson, D.M. Gatti, W. Valdar, C.E. Welsh, R. Cheng, E.J. Chesler, A.A. Palmer, L. McMillan, and G.A. Churchill. High-resolution genetic mapping using the mouse diversity outbred population. *Genetics*, 190(2):437–447, 2012.

[14] Illumina Technote. Infinium genotyping data analysis, 2010.

[15] K. Wang, M. Li, D. Hadley, R. Liu, J. Glessner, S.F.A. Grant, H. Hakonarson, and M. Bucan. Penncnv: an integrated hidden markov model designed for high-resolution copy number variation detection in whole-genome snp genotyping data. *Genome research*, 17(11):1665–1674, 2007.

[16] C.E. Welsh and L. McMillan. Accelerating the inbreeding of multi-parental recombinant inbred lines generated by sibling matings. *G3: Genes| Genomes| Genetics*, 2(2):191–198, 2012.

[17] H. Yang, Y. Ding, L.N. Hutchins, J. Szatkiewicz, T.A. Bell, B.J. Paigen, J.H. Graber, F.P.M. de Villena, and G.A. Churchill. A customized and versatile high-density genotyping array for the mouse. *Nature methods*, 6(9):663–666, 2009.

[18] Q. Zhang, W. Wang, L. McMillan, J. Prins, F. Pardo-Manuel de Villena, and D. Threadgill. Genotype sequence segmentation: Handling constraints and noise. *Algorithms in Bioinformatics*, pages 271–283, 2008.