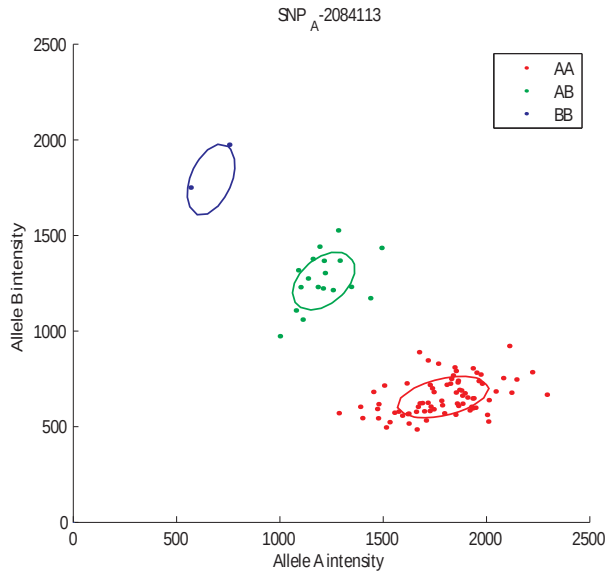
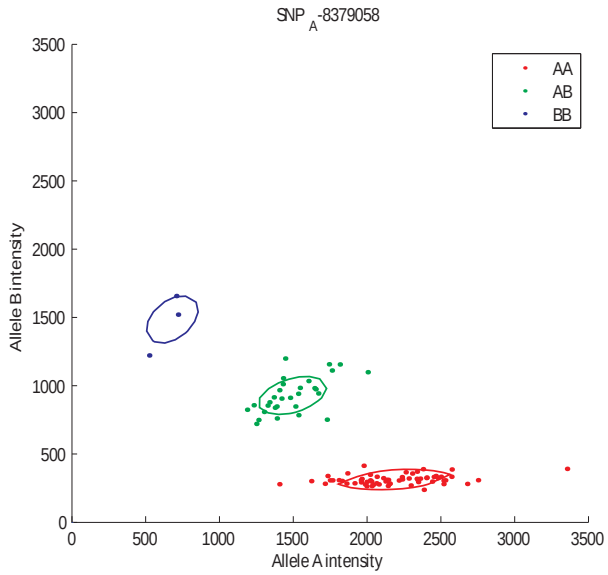
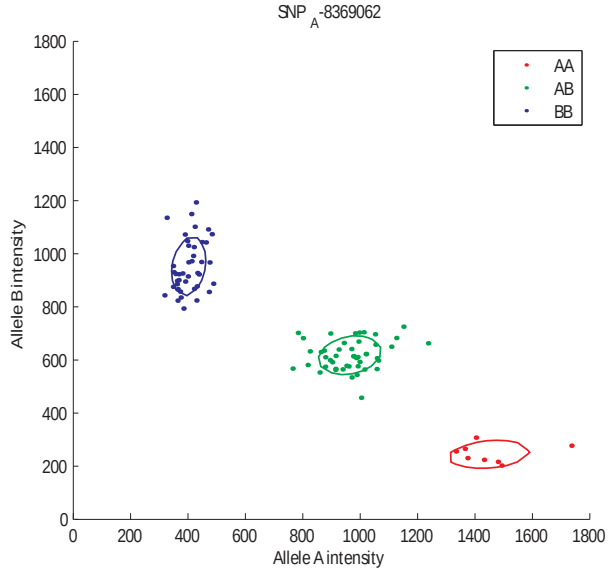
Integrated genotype calling and association analysis of SNPs, common copy number polymorphisms, and rare CNVs

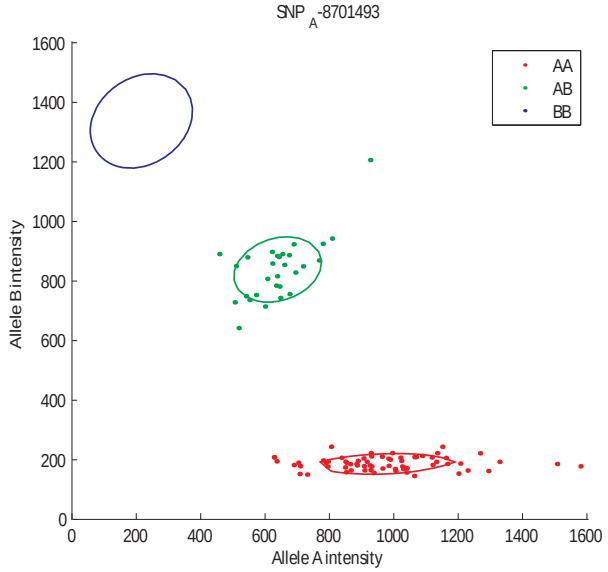
Joshua M Korn, Finny G Kuruvilla, Steven A McCarroll, Alec Wysoker, James Nemesh, Simon Cawley, Earl Hubbell, Jim Veitch, Katayoon Darvishi, Charles Lee, Marcia M Nizzari, Stacey B Gabriel, Shaun Purcell, Mark J Daly, and David Altshuler

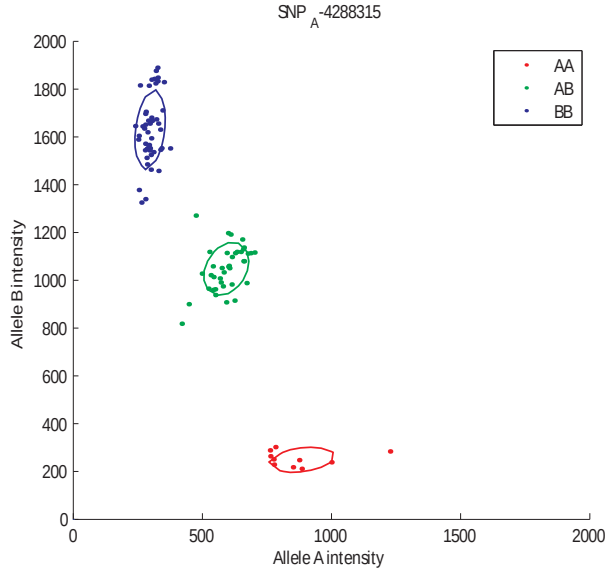
Supplementary Figure 1: A selection of random SNPs genotyped by Birdseed on chromosome 5

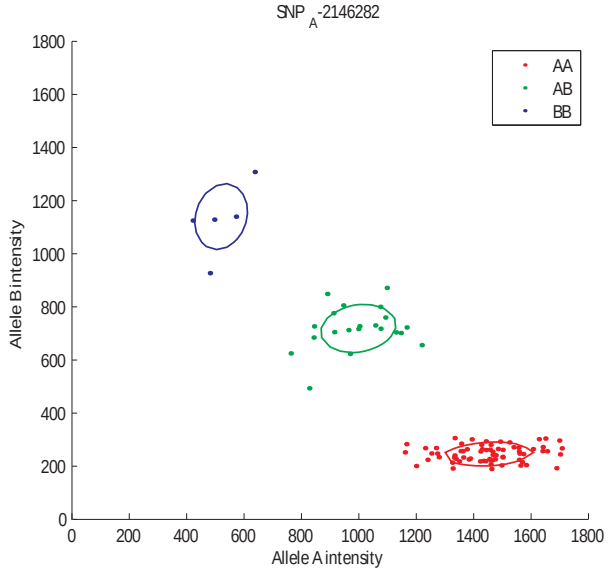


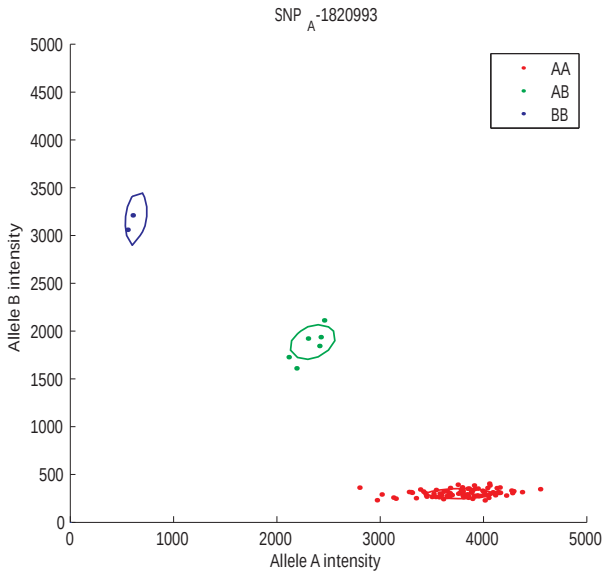


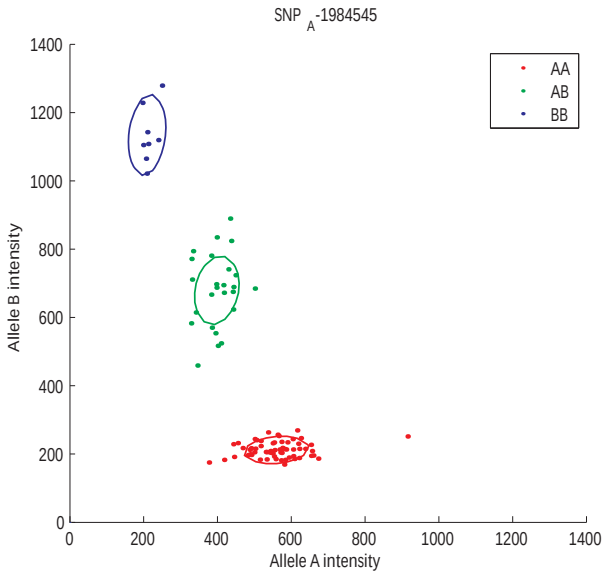


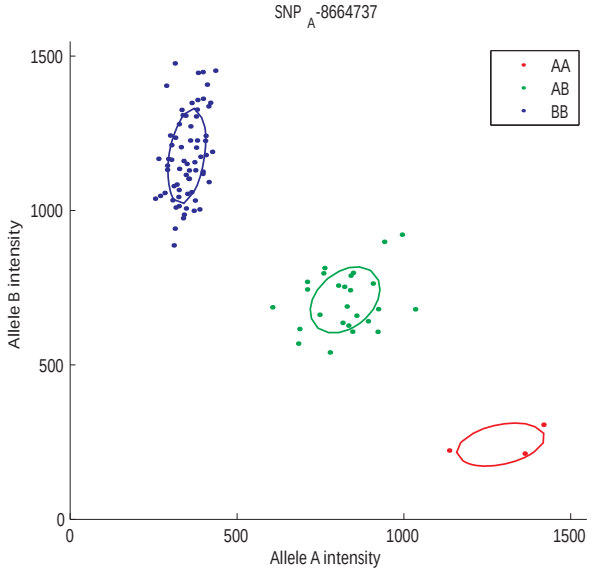


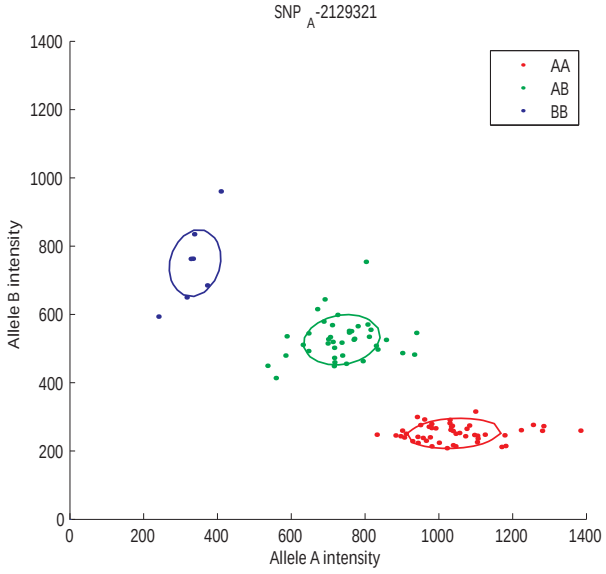


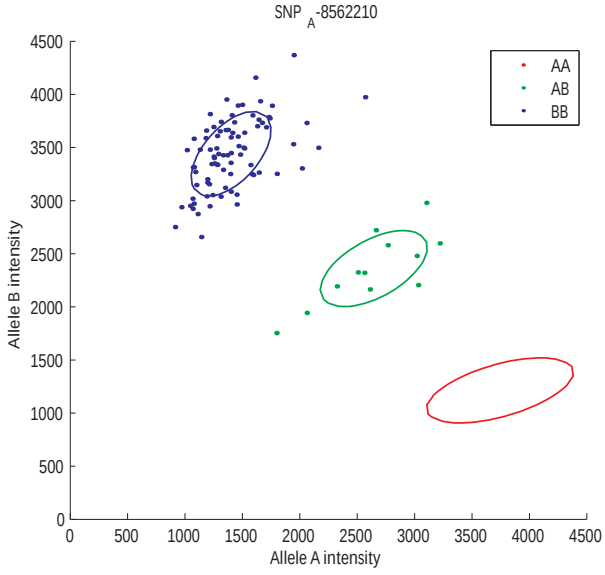


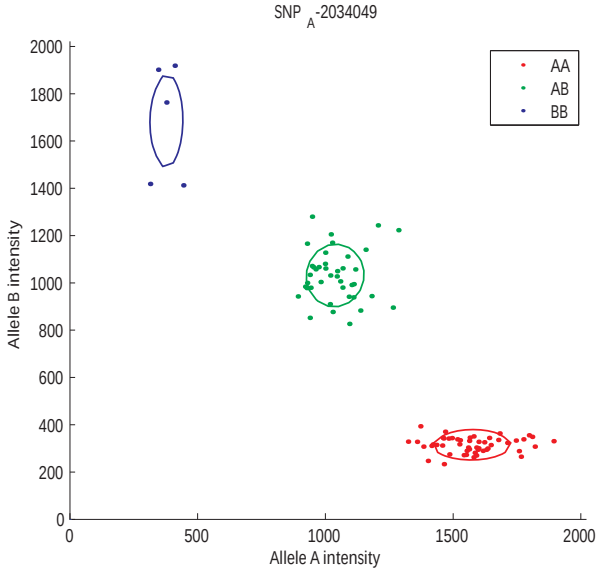

SNP_A-1820993
AA
AB
BB
Allele B intensity
Allele A intensity
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000

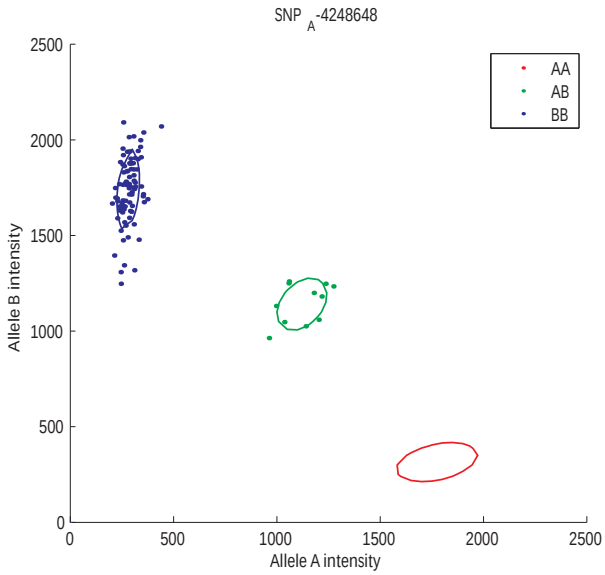

SNP_A-1984545
AA
AB
BB
Allele B intensity
Allele A intensity
0
200
400
600
800
1000
1200
1400

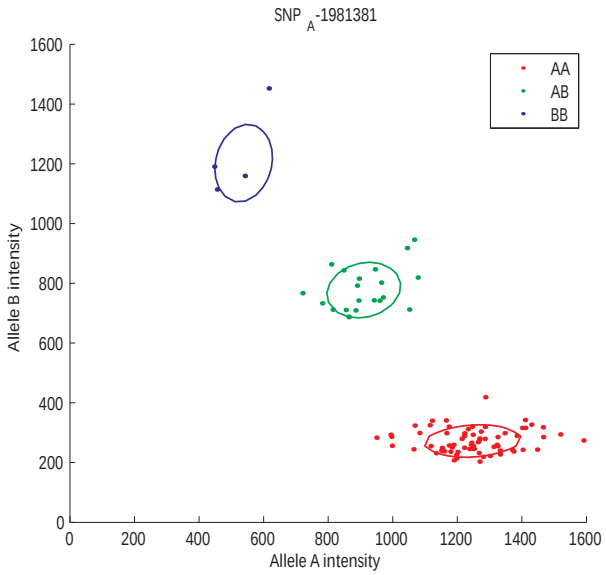

SNP_A-8664737
AA
AB
BB
Allele B intensity
Allele A intensity
0
500
1000
1500

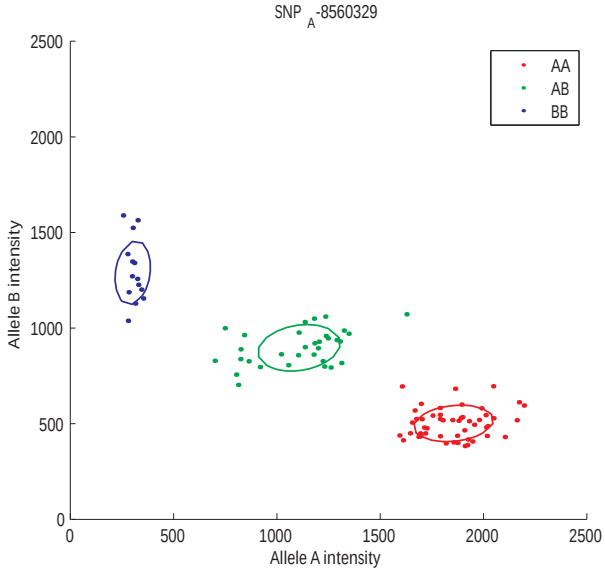

SNP_A-2129321
AA
AB
BB
Allele B intensity
Allele A intensity
0
200
400
600
800
1000
1200
1400

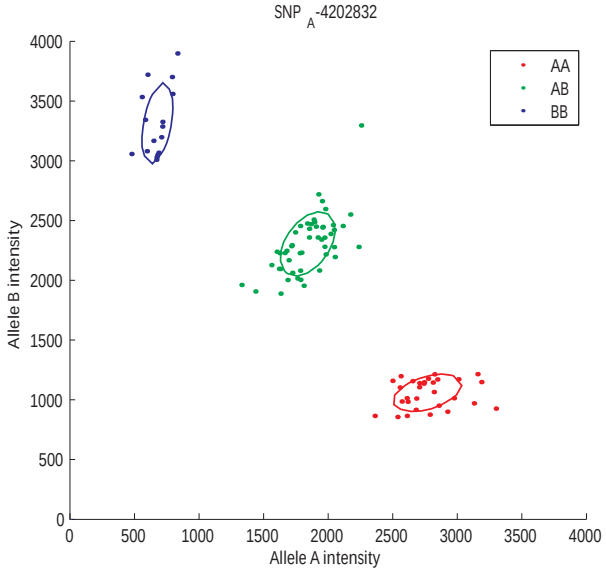

SNP_A-8562210
AA
AB
BB
Allele B intensity
Allele A intensity
0
500
1000
1500
2000
2500
3000
3500
4000
4500

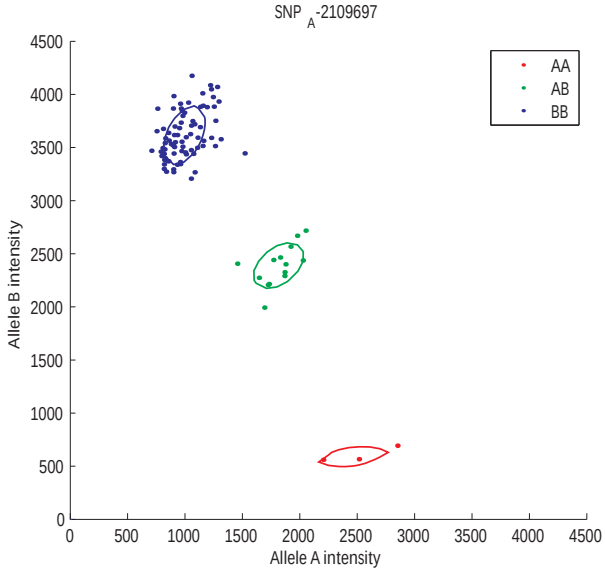

SNP_A-2034049
AA
AB
BB
Allele B intensity
Allele A intensity
0
200
400
600
800
1000
1200
1400
1600
1800
2000

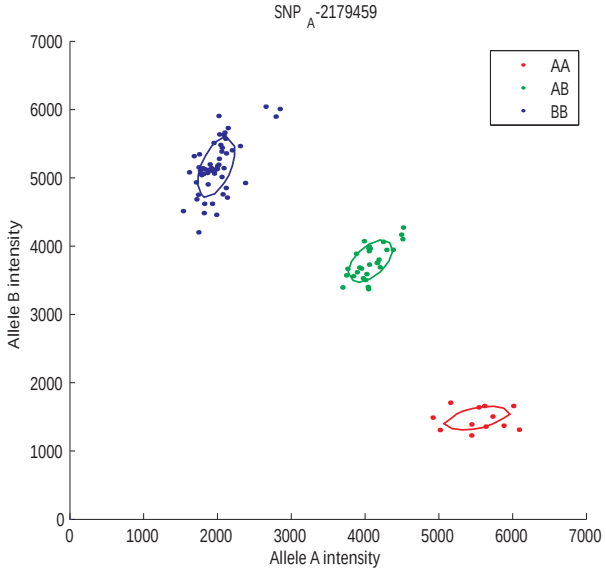

SNP_A-4248648
AA
AB
BB
Allele B intensity
Allele A intensity
0
500
1000
1500
2000
2500


SNP_A-1981381
AA
AB
BB
Allele B intensity
Allele A intensity
0
200
400
600
800
1000
1200
1400
1600

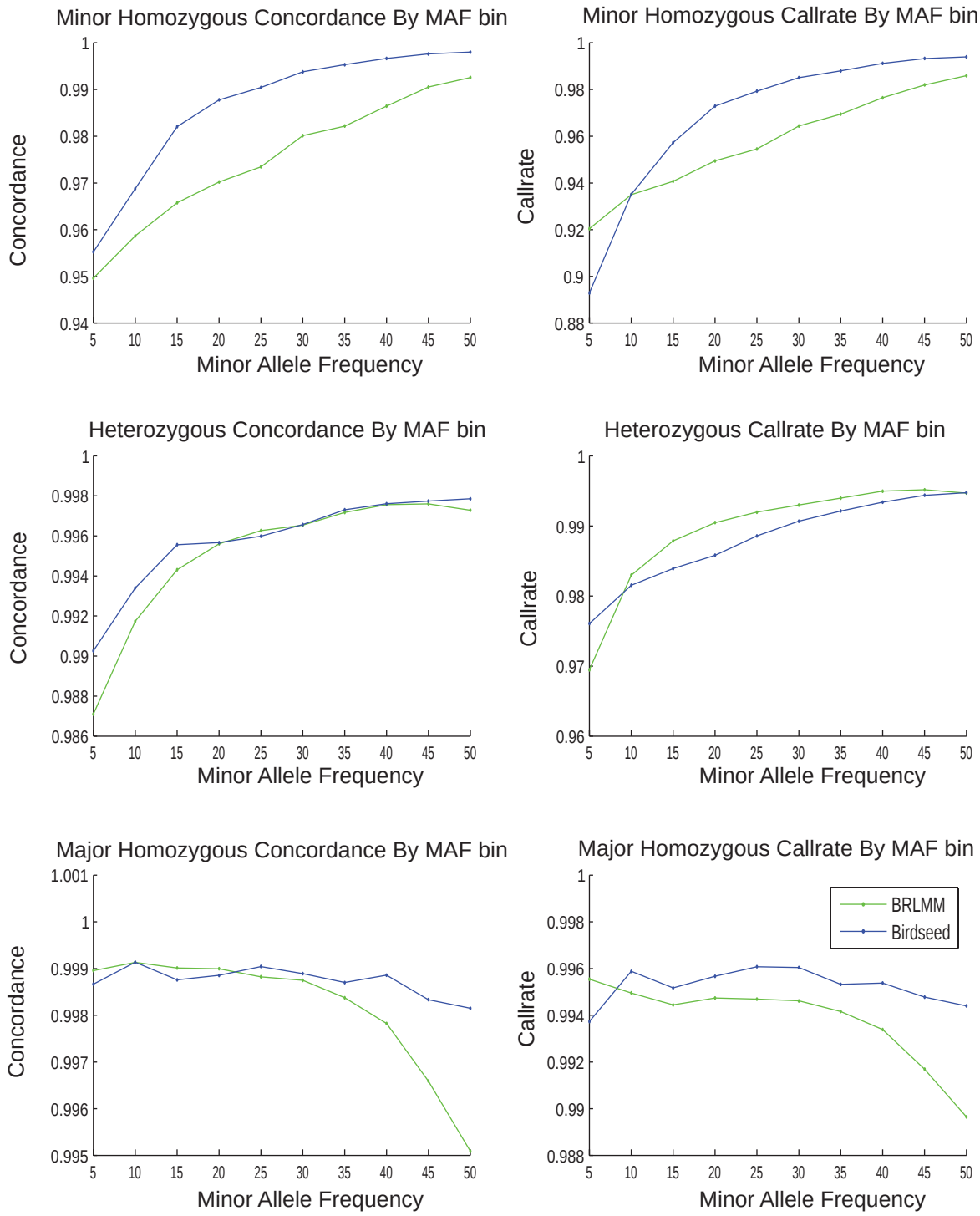

SNP_A-8560329
AA
AB
BB
Allele B intensity
Allele A intensity
0
500
1000
1500
2000
2500


SNP_A-4202832
AA
AB
BB
Allele B intensity
Allele A intensity
0
500
1000
1500
2000
2500
3000
3500
4000


SNP_A-2109697
AA
AB
BB
Allele B intensity
Allele A intensity
0
500
1000
1500
2000
2500
3000
3500
4000
4500


SNP_A-2179459
AA
AB
BB
Allele B intensity
Allele A intensity
0
1000
2000
3000
4000
5000
6000
7000

Supplementary Figure 2



Performance of BRLMM and Birdseed on 63 250K-Nsp HapMap samples.

The 63 samples were normalized together to create a single matrix of intensities (with 500K rows corresponding to the 250K Nsp SNPs on the array, with one for each of allele A and allele B, and with 63 columns corresponding to the 63 sampels). This matrix was then used as input to BRLMM (green) and Birdseed (blue). Birdseed shows significant improvement over BRLMM at capturing minor allele homozygotes in both call rate (a sample that was called a minor homozygote in HapMap was more likely to have a call that passes the confidence threshold in Birdseed than it was in BRLMM) as well as in concordance (a sample called a minor homozygote in HapMap was more likely to be called incorrectly in BRLMM than it was in Birdseed (only those calls passing the confidence threshold are considered). Both algorithms perform more poorly at calling rarely seen genotypes than they do at calling common genotypes, for both the homozygous and heterozygous calls.

Supplementary Table 1

In Silico Gender Mixing Results

Simulated deletion size	Percent simulated deletions discovered with LOD > 0	Mean LOD	Percent simulated deletions discovered with LOD > 1	Percent simulated deletions discovered with LOD > 2	Percent simulated deletions discovered with LOD > 5	Percent of those discovered with breakpoint tolerance of 1 probe for 3- and 5-probe deletions, within 2 probes for 10- . 20-, and 50-probe deletions
3	38%	1.38	20%	10%	0.2%	96%
5	79%	3.12	66%	51%	15%	93%
10	99.5%	9.29	98.5%	97.5%	87%	99%
20	100%	22.32	100%	100%	99.5%	94%
50	100%	53.55	100%	100%	100%	93%

* The simulation covered 101,920,000 unvaried probes, approximately 56.6 genome's worth of data. There were 14 false positive duplications and 1 false positive deletion (<1 per genome) discovered with LOD > 2 (the highest false positive LOD observed was 3.01). There were an additional 106 (approximately 2 per genome) with LOD between 1 and 2, and 929 (approximately 16 per genome) with LOD between 0 and 1. The majority (75%) of these latter category were 5 or fewer probes in length. The false discoveries had an average LOD of 0.44.

CNV event	Frequency (B and BB/-)		Effect model	SNP-only model	Combined SNP and CNP models				
				SNP2	SNP	SNP | CNP	CNV	CNV | SNP	SNP & CNP
Deletions	0.25	I.	{A,B} > { - }	0.05	0.10	0.05	0.97	0.97	0.94
		II.	{ A } > { B, - }	0.89	0.98	0.95	0.73	0.47	0.99
		III.	{ A, - } > { B }	0.88	0.92	0.95	0.24	0.47	0.94
		IV.	{ A } > { B } > { - }	0.36	0.66	0.45	0.90	0.82	0.92
	0.05	I.	{A,B} > { - }	0.05	0.13	0.05	0.50	0.43	0.40
		II.	{ A } > { B, - }	0.47	0.72	0.49	0.45	0.14	0.66
		III.	{ A, - } > { B }	0.47	0.41	0.49	0.05	0.14	0.40
		IV.	{ A } > { B } > { - }	0.16	0.39	0.16	0.47	0.27	0.46
Duplications	0.25	I.	{A,B} > { BB }	0.60	0.85	0.05	0.97	0.68	0.94
		II.	{ A } > { B, BB }	0.99	0.99	0.96	0.73	0.22	0.99
		III.	{ A, BB } > { B }	0.61	0.25	0.96	0.24	0.96	0.95
		IV.	{ A } > { B } > { BB }	0.92	0.95	0.45	0.89	0.09	0.92
	0.05	I.	{A,B} > { BB }	0.06	0.36	0.05	0.50	0.19	0.39
		II.	{ A } > { B, BB }	0.57	0.75	0.49	0.45	0.08	0.66
		III.	{ A, BB } > { B }	0.44	0.18	0.49	0.05	0.38	0.40
		IV.	{ A } > { B } > { BB }	0.25	0.56	0.17	0.48	0.06	0.47

Supplementary Table 2. Simulation study results. Power for 5% type I error, based on 10,000 replicates.
A quantitative trait was simulated and an effect of a single SNP simulated (alleles A, B) with either a deletion (-) or duplication (BB).
The exact effect size of combined SNP and CNV effects varies by condition, but is typically on the order of 1% of total phenotypic variance.
See next page for more information concerning the simulation and interpretation of this table.

To evaluate the performance of the combined SNP and CNP association model, we performed a simulation study, reported in Supplmentary Table 2. Simulating a single copy-number variant SNP for a sample of 1,000 individuals (10,000 replicates per condition), we considered four effect models (I-IV in Supplementary Table 2) for both common deletions (*A*, *B*, -) and duplications (*A*, *B*, *BB*); in addition, the frequency of the *B* allele and the deletion or duplication was fixed at either 25% or 5% each (and so the *A* allele was either 50% or 90%). In all cases the outcome variable was a quantitative trait. We also considered null model simulations (with no effect of either SNP or CNP on phenotype). The combined effect of both SNP and CNP varies across conditions but is typically on the order of 1% of the total phenotypic variance.

We evaluated the performance of six distinct tests under these 16 conditions. The first five tests are based on the true set of canonical and non-canonical genotype data, jointly representing both copy number and allelic variation, via the model introduced above:

CNP | SNP	Test of copy number controlling for allelic variation
SNP | CNV	Test of allelic variation controlling for copy number
SNP & CNP	Joint 2df test of copy number and allelic variation
SNP	Test only of allelic variation but using non-canonical genotypes
CNV	Test only of copy number variation

Aside from the joint test, all tests are 1df tests. The first three tests correspond to those outlined in the main text; the fourth and fifth are based on entering only the difference of the allele counts (*SNP*) or only the sum (*CNV*).

The sixth test is designed to approximate the performance of a "traditional" SNP genotype calling and analysis pipeline, considering only SNP effects and assuming canonical genotype data. For this test, homozygous deletions and allelically-heterozygous duplications (i.e. A/BB) were set to missing; single deletions and allelically-homozygous duplications were called as two-copy homozygotes (e.g. B/- and B/BB were called B/B). This test is labeled *SNP2*, indicating the assumption of diploid state. It is important to note that the genotyping “error model” specified here is somewhat optimistic: in practice one might expect increased rates of missing and incorrectly-called genotypes, further deteriorating the performance of *SNP2*.

Under the null, all tests under all conditions gave appropriate type I error rates of 5% (data not shown). The full pattern of results under the alternate (Supplmentary Table 2) represents a number of complex factors: power is influenced by the relative frequencies of events and alleles, whether the CNPs are deletions or duplications as well as which allele is duplicated, the *in silico* genotyping error model implicit in the *SNP2* test and the assumption of additivity across potentially 0 to 4 copies of an allele. Here we only focus on the most relevant key features.

In general, the new combined 2df test (*SNP & CNP*) performs well and is more powerful than the standard *SNP2* test, sometimes substantially more so. Similarly, the non-conditional tests framed within the combined model (*SNP* and *CNP*) perform well; perhaps of more interest is the performance of the conditional tests, that ask whether there is any effect of allele over and above that of copy number and vice versa. To illustrate the way in which the joint model’s conditional tests can disentangle SNP and CNP effects, consider the 9th row of Supplmentary Table 2, in which a common duplication *BB* has an effect relative to alleles *A* and *B*. That is, although the duplication is causal (and other SNPs in the same CNP might also be causal) this particular SNP has no influence on phenotype over and above the CNP. The standard *SNP2* test has 60% power in this case, arising from the correlation between the duplication and the *B* allele; the joint 2df test is more powerful however, giving 94% power in this scenario. More importantly, the test of *SNP | CNP* has only 5% power, which is the expected rate under the null given the 5% type I error rate specified. In contrast, the test of *CNP | SNP* has 68% power. In other words, in this new model would tend to *a)* be more likely to locate this locus in the first place and *b)* be able to show that this particular SNP is not associated with phenotype once the duplication has been taken into account.

Supplementary Note: Sensitivity of Birdsuite, and comparison to other algorithms

We evaluated the ability of three algorithms—Birdsuite, Nexus, and Partek—to identify a set of 893 independently discovered and validated CNVs. These reference CNVs had been identified in eight of the HapMap samples by fosmid end-sequence-pair (ESP) analysis and localized by complete resequencing or 200bp-resolution array CGH (from Kidd *et al.*, 2008, Supplementary Table 3).

We used the same CEL files (representing hybridization of 263 HapMap samples to the Affymetrix SNP 6.0 array) as input to each algorithm. Sensitivity was judged based on ability to recover the reference CNVs from Kidd *et al.* A reference CNV was determined to be "recovered" if the algorithm called a CNV in the same sample and at the same genomic location as the reference CNV. To determine whether the same locations had been identified, we used the criterion that the genomic region in the overlap of the reference and called CNV had to be at least 25% of the length of the region spanned by the reference and called CNV together.

We report results for Birdsuite (using Canary calls with an uncertainty < 0.1 together with Birdseye calls with a LOD score > 5), Partek (using default parameters; see below), Nexus (using default settings; see below), and Nexus with relaxed settings (see below). Since sensitivity to discover a CNV is strongly related to the number of probes spanned by that CNV, the analysis below is stratified on the number of probes on the Affymetrix SNP 6.0 array overlapped by each of the reference CNVs. Results are reported both in terms of absolute number (top) as well as in terms of percentage (bottom) of the reference CNVs recovered.

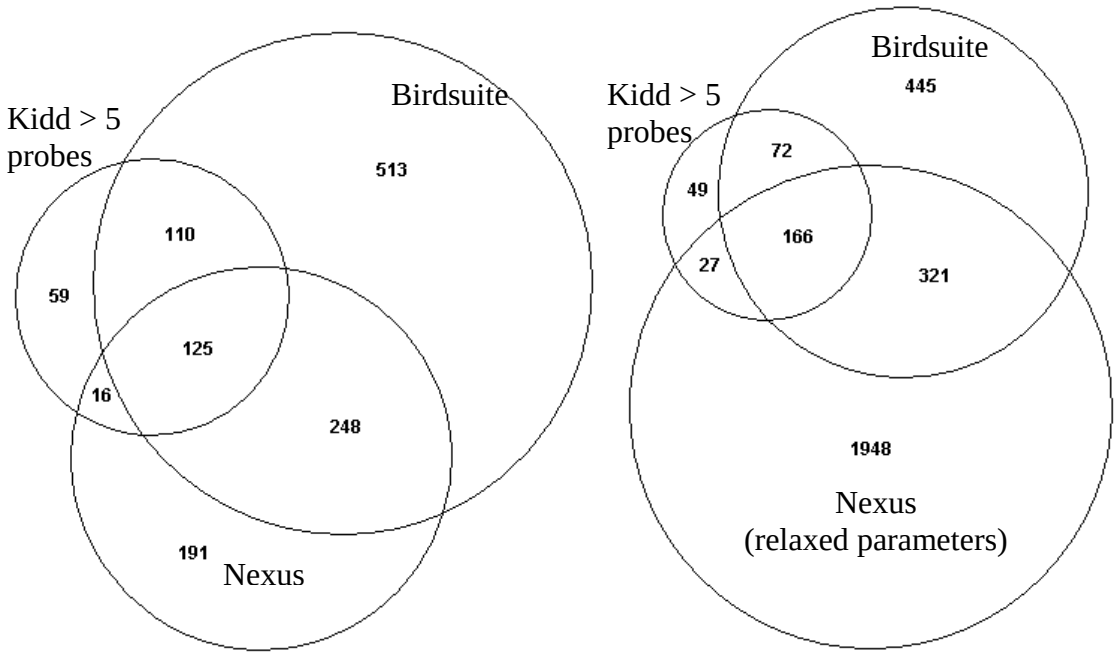
Sensitivity to recover CNVs from Kidd et al.		**Birdsuite**	**Partek**	**Nexus**	**Nexus (relaxed)**
Probes spanned by CNV	**Total in Category**	1344	2164	573	2474
<= 1 probes	325	12	4	2	4
2-5 probes	256	84	1	2	14
6-10 probes	112	69	2	15	47
11-20 probes	71	46	3	34	49
> 20 probes	129	121	15	94	96

Probes spanned by CNV	**Total in Category**	**Birdsuite**	**Partek**	**Nexus**	**Nexus (relaxed)**
<= 1 probes	325	3.7%	1.2%	0.6%	1.2%
2-5 probes	256	32.8%	0.4%	0.8%	5.5%
6-10 probes	112	61.6%	1.8%	13.4%	42.0%
11-20 probes	71	64.8%	4.2%	47.9%	69.0%
> 20 probes	129	93.8%	11.6%	72.9%	74.4%

As expected, the sensitivity of all algorithms was poor when the number of probes spanned by a CNV was small. For all algorithms, sensitivity increased with the number of probes spanned by the CNV.

Of course, the greater sensitivity observed for Birdsuite could in principle be due to the use of more-relaxed calling thresholds. We do not believe this to be the case, because Birdsuite appeared to be more stringent than Partek with default settings (1344 CNVs called vs. 2164) and than Nexus with relaxed settings (1344 CNVs called vs. 2474). Thus, despite having called fewer total segments as copy-number variable (than Partek and Nexus) using the settings above, Birdsuite identified more of the CNVs that were discovered independently by a sequencing-based approach. Additionally, while specificity is difficult to judge without attempting to experimentally validate a large number of new calls from each algorithm, we have demonstracted high specificity for Birdsuite calls by using reproducibility of copy number calls, comparison to quantiative PCR data, Mendelian inheritance checks, and simulated datasets.

The following Venn diagram shows sensitivity of Birdsuite and Nexus to those reference CNVs overlapping at least 6 probes on the Affymetrix 6.0. In addition to Birdsuite being more sensitive at default thresholds, relaxing the thresholds of Nexus recovers only moderately more reference CNVs (as well as more overlaps with Birdsuite CNVs), but at many times the total number of called CNVs. While Birdsuite has many called CNVs that are not in the reference dataset, most of these are due to the common CNPs whose accuracy has been demonstrated (McCarroll *et al.*, accompanying paper), and whose size is often below the threshold detectable by fosmid-end sequencing.



These results demonstrate the importance of parsing CNVs into rare/undiscovered and common/known categories for analysis. This is particularly true for small events: by utilizing prior knowledge (the CNP map from McCarroll *et al*.) to genotype common CNPs, we can not only confidently detect and genotype more than 30% of events overlapping only 2-5 probes (a size typically insufficient to confidently discover CNVs *ab initio*), but also a higher percentage of large CNVs that may be difficult to detect because of decreased probe sensitivity (e.g. many are in segmental duplications, where

cross-hybridization leads to vastly different probe characteristics, interfering with *ab initio* algorithms). This division of structural variation into known/common and undiscovered/rare components also allows an algorithm such as Birdsuite to detect undiscovered/rare CNVs at more stringent thresholds without compromising the ability to genotype known/common CNPs.

Partek Parameters:
Segmentation Parameters: Minimum probesets = 10; P value = 0.001; signal to noise = 0.3
Region Report: Below = 1.7; Above = 2.3; P value = 0.01

Nexus Parameters:
Threshold setting; P value cut off = 0.05; Aggregate % cut off = 35; Min number of probes per segment = 5
High gain = 0.5; Gain = 0.2 ; Loss = -0.3; Big Loss = -0.7

Nexus Relaxed Parameters:
Rank Segmentation; Significant threshold = 0.0001; Max contiguous probe spacing(kbp) = 1000; Min number of probes per segment = 5
High gain = 0.5; Gain = 0.2 ; Loss = -0.3; Big Loss = -0.7

References:
Kidd, J.M. et al. Mapping and sequencing of structural variation from eight human genomes. *Nature* **453**, 56-64 (2008).
McCarroll, S.A., et al. Integrated detection and population genetic analysis of SNPs and copy number variation. *Nature Genetics*, accompanying paper.

Supplementary Methods

Note: All binaries and source code are available on the website:
http://www.broad.mit.edu/mpg/birdsuite

Canary: Algorithm

Overview

Canary's function is to genotype samples in regions of known copy-number variation, especially suited for regions of common variation.

Inputs

Summarized (median polish) intensities from a set of .cel files across a set of predefined probes in an external map.

A models file which contains the means, variances, and proportions of discrete genotype classes observed in those same regions. (The intensity data for this paper was taken from an average of two independent runs of the HapMap samples, one at Affymetrix, Inc. and the second at the Broad Institute of Harvard and MIT.)

Description of algorithm

Canary is a 1d Gaussian Mixture Model (GMM) to cluster samples into discrete copy number classes.

Initialization:

The initial conditions for each cluster are specified in a prior models file that contains CNP-specific estimates of cluster locations and variances (in terms of summarized and scaled intensity values) learned from samples of known genotype; a series of models are tested consisting of different number and combination of genotype clusters (since not all potential copy number levels are necessarily represented in the given dataset).
The following series of models is tested depending on parameters and CNP information, where use_af refers to a boolean parameter that when true, restricts the search space to only include alleles previously observed at that CNP.

I. If limiting to alleles observed in samples used to generate the models file, examine whether or not the models record the existence of a deletion only (i.e. 0 frequency for clusters at copy number 3 and 4.) If so, the following models are tested: [0], [2], [0,1], [1,2], and [0,1,2]. Similarly, if the CNP records a duplication only, these models are tested: [2], [4], [2, 3], [3, 4], [2,3,4].

II. If not limiting to alleles observed in samples used to generate the models file, or if the models record both a deletion and a duplication at the CNP, then the following models are tested: [0], [2], [4], [0,1], [1,2], [2,3], [3,4], [0,1,2], [1,2,3], [2,3,4], [0,1,2,3], [1,2,3,4], and [0,1,2,3,4].

For a given initialization of G gaussian clusters and N samples, cluster parameters are updated via a modified Expectation-Maximization (EM) algorithm, iteratively estimating cluster membership (E-step) and maximizing cluster parameters (M-step) (Dempster, 1977).

E-Step:

Standard posterior assignment probabilities are computed for each sample i (with intensity x_i) and cluster j (with mean μ_j, standard deviation σ_j, and frequency w_j) pair.

$$P(j \mid i) = K * (w_j / \sigma_j) * \exp(-(x_i - \mu_j)^2 / (2\sigma^2_j)) \quad \textbf{(1)}$$

where K is a normalization constant set such that $(\Sigma_{j=1 \text{ to } G} P(j \mid i)) = 1$. $P(j \mid i)$ reflects the relative probability sample i was drawn from the normal distribution j compared to the other Gaussians.

M-Step:
The parameters of each cluster are updated based on current cluster membership. Typical M-Step updates are modified to include s pseudopoints placed at the prior model's mean μ_{pj} for each cluster j, and to regularize the variances toward a common shared term.

$$w_j = (\Sigma_{i=1 \text{ to } N} P(j \mid i)) / N \quad \textbf{(2)}$$

$$\mu_j = [((1 / (s + w_j)) * \Sigma_{i=1 \text{ to } N} (P(j \mid i) * x_i))] + [(s / (s + w_j)) * \mu_{pj}] \quad \textbf{(3)}$$

$$\sigma_j^2 = (1 / w_j) * \Sigma_{i=1 \text{ to } N} (P(j \mid i) * (x_i - \mu_j)^2) \quad \textbf{(4)}$$

After all clusters have been updated, the variance terms are regularized to be similar to each other:

$$\text{Expected variance } \sigma_{avg}^2 = (1 / G) * \Sigma_{j=1 \text{ to } G} (\sigma_j^2) \quad \textbf{(5)}$$

$$\sigma_j^2 = (1 / (1 + r)) * \sigma_j^2 + (r / (1 + r)) * \sigma_{avg}^2 \quad \textbf{(6)}$$

where r is the regularization parameter.
EM is iterated for at least a minimum number of iterations until convergence (or a maximum number of iterations is reached), at which time the next model is tried.

Model Selection:
A series of heuristics are used to determine which GMM model is best (in addition to the relative probability of observing the data given the EM-fit model).

1. Bayesian Information Criterion (BIC): 2 * log-likelihood - G * log(n)
2. reward_closeness: how close the means of the fit data are to the prior model for that CNV
3. af: The difference in allele frequency of the fit data with that CNV prior model
4. hwe_penalty: a Hardy-Weinberg penalty
5. overlap: A penalty which describes how much the peaks of the fit data overlap each other

Genotyping:
Using the selected model, each sample i is genotyped as the copy number of the cluster j which maximizes the equation

$$P(j \mid i) = (w_j / \sigma_j) * \exp(-(x_i - \mu_j)^2 / (2\sigma^2_j)) \quad \textbf{(7)}$$

Samples are furthermore assigned a confidence reflecting the relative likelihood of belonging to the next-best cluster (a global confidence), as well as a local confidence reflecting their absolute probability. This local confidence is computed as a sigmoidal function of the number of standard deviations away from the cluster center a sample lies. To compute the global confidence score:

$$conf_{global\text{-}i} = P(j_{second\text{-}best} \mid i) / P(j_{best} \mid i) \quad \textbf{(8)}$$

To compute the local confidence score:

$$q_i = sqrt((x_i - \mu_j)^2 * (1 / \sigma^2_j) / 2) \quad \textbf{(9)}$$

$$conf_{local\text{-}i} = 1/(1 + \exp(p - q_i)) - 1/(1 + \exp(p)) \quad \textbf{(10)}$$

where q is the number of standard deviations a sample is from its assigned cluster, and p is the number of standard deviations beyond which the local confidence score quickly increases (the inflection point in Eq. **(10)**).

The overall confidence score is then:

$$(c * conf_{global\text{-}i}) + ((1 - c) * conf_{local\text{-}i}) \quad \textbf{(11)}$$

where c is the relative confidence contribution parameter, $0 <= c <= 1$.

Canary: Manual & Pseudocode

Overview

Canary's function is to genotype samples in regions of known copy-number variation, especially suited for regions of common variation.

Inputs

Summarized (median polish) intensities from a set of .cel files across a set of predefined probes in an external map.

A models file which contains the means, variances, and proportions of discrete genotype classes observed in those same regions. (The intensity data for this paper was taken from an average of two independent runs of the HapMap samples, one at Affymetrix, Inc. and the second at the Broad Institute of Harvard and MIT.

Outputs

Genotypes: 0, 1, 2, 3, and 4 at each predefined CNV in each sample, along with a confidence score ranging from 0 (most confident) to 1 (least confident)

Below is a pseudocode sketch of the algorithm. All code is open-source and can be downloaded free of charge from www.broad.mit.edu/mpg/birdsuite. For additional detail, see the actual source code.

There are a variety of flags and constants that can be adjusted according to the user's preference.
n: number of samples being clustered
use_af: Boolean flag on whether or not to draw on the allele frequency (af) estimates from the models file (default is True)
min_iter: the minimum number of times to execute the EM loop (set at 10)
var_reg: how much to regularize the variances (0 not at all, 1 fully) (r in Eq. **(6)**)
pseudopoint_factor: how much to "anchor" the EM algorithm to the model locations (smaller is more weight, default is 100) (n divided by s from Eq. **(3)**)
af_weight: how much to use allele frequency in the scoring of models (0 not at all, 1 fully)
hwe_weight: how much to use hwe in the scoring of models (0 not at all, 1 fully)
hwe_tol: if a non-zero hwe weight, below this start to penalize (set at 1e-4)
hwe_tol2: if a non-zero hwe weight, below this start to penalize (set at 1e-11)
rel_confidence_weight: a constant from 0 to 1 which determines how much to weight global penalties to local penalties (set to 0.8) (c in Eq. **(11)**)
trailoff_par: reflects how much to penalize points in the confidence score regarding distance from its closest peak (set at 3) (p in Eq. **(10)**)

I. If use_af is True, examine whether or not the models record the existence of clusters at 3 and 4. If not, store into a variable clustering_set to test the following models: [0], [2], [0,1], [1,2], and [0,1,2]. If there are no clusters are 0 or 1, then store into clustering_set the following models: [2], [4], [2, 3], [3, 4], [2,3,4].

II. If use_af is False, or if it is True and there are clusters at [0 or 1] and [3 or 4], then store into clustering_set to test the following models: [0], [2], [4], [0,1], [1,2], [2,3], [3,4], [0,1,2], [1,2,3], [2,3,4], [0,1,2,3], [1,2,3,4], and [0,1,2,3,4].

III. Invoke an Expectation-Maximization algorithm on each of the models in clustering_set, storing the clustering results derived from each of the configurations testing in clustering_set. The algorithm is a standard EM with the following exceptions:

1. After the M step, the variances are regularized to the mean of the variances by a factor var_reg (typically set to something like 0.4).
2. A number of pseudopoints are placed at the cluster center locations (derived from the models file). The number of points is: max(1,floor(n/pseudopoint_factor)).
3. The EM algorithm is forced to run at least min_iter times.

IV. Determine how well each of the cluster configurations after EM fits the data and the models. The components that weigh into assessing this fit are:

1. bic: 2 * log-likelihood - G * log(n)
2. reward_closeness: how close the means of the fit data are to the model for that CNV
3. af: The difference in allele frequency of the fit data with that CNV model
4. hwe_penalty: a Hardy-Weinberg penalty
5. overlap: A penalty which describes how much the peaks of the fit data overlap each other
6. The above five terms are put together in the following fashion: 5 * bic - abs(af * bic) + reward_closeness / (overlap + af + 1) - overlap - 30 * hwe_penalty

V. The model with the highest score is selected. The next step is imputation of clusters that are missing. If use_af is True, then the imputation is directional, which means that if the model does not describe clusters at positions 3 and 4, then clusters are not imputed in that "direction," i.e. they are imputed toward 0 and 1, but not upward. Similarly if the model does not describe clusters at points 0 and 1, then clusters are not imputed in that direction. Directional imputation translates the models clusters to map onto the fit data, and the desired cluster centers and variances are taken after the translation occurs. Model variances are used without modification for missing clusters.

VI. Confidences are generated from the final clusters. Given the computed clusters, confidence scores are computed for every CNV at every sample. The confidence score is a combination of exactly two terms: a "global" and a "local" penalty. The global penalty refers to how much a point appears to belong to one peak versus its next closest rival. The global penalty is simply the membership score of the EM algorithm of the second closest peak divided by the closest peak. The local penalty reflects how far away a point is from its closest peak, in a more absolute sense.

Birdseed: Algorithm

Overview

Birdseed's function is to assign AA, AB, and BB genotype calls to each sample with two copies of a SNP, and characterize genotype- and allele-specific probe responses for each SNP.

Conventions and terms (extrapolate where necessary):

N refers to the number of samples
G refers to the number of clusters
x_{ai} refers to the normalized intensity of allele A of sample i
x_{bi} refers to the normalized intensity of allele B of sample i
x_i refers to the paired normalized intensity (x_{ai}, x_{bi}) of sample i
μ_{AAa} refers to the expected mean of allele A of the AA cluster from the prior models
μ_{ABb} refers to the expected mean of allele B of the AB cluster
μ_{2a} refers to the mean of allele A of the 2nd cluster (not yet assigned to a genotype class; a cluster that will become a posterior model for a genotype class)
μ_3 refers to the 2d point (μ_{3a}, μ_{3b})
σ_{AAa} refers to the expected standard deviation in the allele A dimension of the AA cluster
σ_{ABb} refers to the expected standard deviation in the allele B dimension of the AB cluster
σ_{2a} refers to the standard deviation in the allele A dimension of the 2nd cluster
cor refers to the shared covariance term representing the correlation of noise in the A and B dimensions across all clusters.
Σ_3 refers to the 2d covariance matrix:

$$[[\sigma_{3a}^2,\ cor*\sigma_{3a}*\sigma_{3b})]$$
$$[cor*\sigma_{3a}*\sigma_{3b}),\ \sigma_{3b}^2]]$$

w_{AA} refers to the expected weight of the AA cluster
w_{AB} refers to the expected weight of the AB cluster
w_2 refers to the weight of the 2nd cluster
Σ_i refers to a summation of the following expression, over all samples from i = 1 to N

Inputs

Normalized and summarized (median polish) intensities from a set of .cel files for each SNP allele. This takes the form of a matrix with N columns and 2*M rows (ignoring headers), where N is the number of samples to be analyzed, and M is the number of SNPs. Note: for the version of birdseed included in apt-probeset-genotype, normalization and summarization can occur on the fly, and .cel files can be input directly.

A file describing the gender of each sample. Note: for the version of birdseed included in apt-probeset-genotype, gender is calculated on the fly, and this file is unnecessary.

A models file which contains the means, variances, and proportions of discrete genotype classes observed for each SNP in HapMap. (The default models file was built using a run of the HapMap samples at Affymetrix, Inc.—file available online.)

Optionally, a file listing for each SNP which samples are not expected to have two copies and thus should be excluded from Birdseed for that SNP (typically this is a file generated

from Canary results). SNPs where all samples are expected to have two copies need not be included in the file.

Optionally, a file listing which SNPs have unusual copy number for males, females, or both, and what the expected copy number is for each gender. (I.e. chrX, chrY, and mitochondrial SNPs.)

Description of algorithm

Birdseed is a 2d Gaussian Mixture Model (GMM) that clusters diploid samples into the canonical SNP genotype classes AA, AB, and BB. For the most part, the algorithm is a 2-dimensional GMM analogous to the 1-dimensional Canary. Slight modifications are made to support sex and mitochondrial chromosomes (see Manual & Pseudocode).

Initialization:

The initial conditions for each cluster are based off a prior models file that contains SNP-specific estimates of cluster locations and variances learned from samples of known genotype; a series of models are tested consisting of different number of clusters (since not all potential SNP genotype classes are necessarily represented in the given dataset). Priors are scaled in a SNP-specific manner by default to get them into the same intensity space as the samples. By default, this scale s is calculated separately for each SNP as the average distance of a sample from the origin divided by the weighted average of the prior model means from the origin.

$$n = (\Sigma_i \text{ sqrt}(I_{ai}^2 + I_{bi}^2)) / N \quad \textbf{(12)}$$

$$d = (\Sigma_{c=AA,AB,BB} (w_c + 0.1) * \text{sqrt}(\mu_{ca}^2 + \mu_{cb}^2)) / \Sigma_{c=AA,AB,BB} (w_c + 0.1) \quad \textbf{(13)}$$

$$s = \text{numerator} / \text{denominator} \quad \textbf{(14)}$$

The means in the SNP's priors are scaled by s, while the variances are scaled by s^2.

The following series of models is tested with the specified initializations, in the specified order; unlike Canary however, first the number of clusters is determined followed by a labeling process to align the EM-fit clusters to genotype classes, as opposed to assuming the label up-front (except in the case of 3 clusters, where the labels must be AA, AB, and BB):

A. G = 1 (a single cluster model)

--no initialization, since EM procedure is not required

B. G = 2 (a 2-cluster model)

$\mu_1 = \mu_{AA}$

$\mu_2 = \mu_{BB}$

$\sigma^2_{1a} = \sigma^2_{2a} = (2/3) * (\sigma^2_{1a}$ from G = 1 model)

$\sigma^2_{1b} = \sigma^2_{2b} = (2/3) * (\sigma^2_{1b}$ from G = 1 model)

cor = 0.0

$w_1 = w_2 = 0.5$

C. G = 3 (a 3-cluster model)

$\mu_1 = \mu_{AA}$

$\mu_2 = \mu_{AB}$
$\mu_3 = \mu_{AB}$
$\Sigma_1 = \Sigma_{AA}$
$\Sigma_2 = \Sigma_{AB}$
$\Sigma_3 = \Sigma_{BB}$
cor = weighted mean of cor term in $\{\Sigma_{AA}, \Sigma_{AB}, \Sigma_{BB}\}$
$w_1 = w_2 = w_3 = 1/3$

D. G = 3 (a 3-cluster model with a different initialization)
μ_{1a} = maximum A allele intensity observed in samples
μ_{1b} = minimum B allele intensity observed in samples
μ_{3a} = minimum A allele intensity observed in samples
μ_{3b} = maximum B allele intensity observed in samples
$\mu_{2a} = (\mu_{1a} + \mu_{3a}) / 2$
$\mu_{2b} = (\mu_{1b} + \mu_{3b}) / 2$
$\Sigma_1, \Sigma_2, \Sigma_3$ are calculated equivalent to the way that expected variance is calculated (see Eq. **(23)**), using Σ_{AA}, Σ_{AB}, and Σ_{BB} as input.
Σ_1 and Σ_3 are initialized to further be var_start times larger.
Σ_2 is initialized to further be var_start / 20 as large.
cor = 0
$w_1 = w_2 = w_3 = 1/3$

For a given initialization, cluster parameters are updated via a modified Expectation-Maximization (EM) algorithm, iteratively estimating cluster membership (E-step) and maximizing cluster parameters (M-step).

E-Step:
Standard posterior assignment probabilities are computed for each sample i (with intensity x_i) and cluster j (with mean μ_j, standard deviation σ_j, and frequency w_j) pair.

$$P(j \mid i) = K * (w_j / |\Sigma_j|^{1/2}) * \exp(-(x_i - \mu_j)^T * \Sigma_j^{-1} * (x_i - \mu_j) / 2)) \quad \textbf{(15)}$$

where K is a normalization constant set such that $(\Sigma_{j=1 \text{ to } G} P(j \mid i)) = 1$. $P(j \mid i)$ reflects the relative probability sample i was drawn from the normal distribution j compared to the other Gaussians.

M-Step:
The parameters of each cluster are updated based on current cluster membership. Typical M-Step updates are modified to include s pseudopoints placed at the prior model's mean μ_{pj} for each cluster j when G == 3 (s is 0 for G<3), to utilize a single term representing noise correlation in the two alleles (dimensions) instead of the typical calculation for cross-correlation (since crosstalk remains constant regardless of cluster), and to regularize the variances toward a single function when G >= 2. The correlation term is forced to be positive during the first few iterations to avoid fitting a bad local optimum (see manual & pseudocode for more details).

$$w_j = (\Sigma_{i=1 \text{ to } N} P(j \mid i)) / N \quad \textbf{(16)}$$

$$\mu_j = [((1 / (s + w_j)) * \Sigma_{i=1 \text{ to } N} (P(j \mid i) * x_i))] + [(s / (s + w_j)) * \mu_{pj}] \quad \textbf{(17)}$$

$$\sigma_{jaa}^2 = (1 / w_j) * \Sigma_{i=1 \text{ to } N} (P(j \mid i) * (x_{ia} - \mu_{ja})^2) \quad \textbf{(18)}$$

$$\sigma_{jbb}^2 = (1 / w_j) * \Sigma_{i=1 \text{ to } N} (P(j \mid i) * (x_{ib} - \mu_{jb})^2) \quad \textbf{(19)}$$

$$\sigma_{jab}^2 = (1 / w_j) * \Sigma_{i=1 \text{ to } N} (P(j \mid i) * (x_{ia} - \mu_{ja}) * (x_{ib} - \mu_{jb})) \quad \textbf{(20)}$$

After all clusters have been updated, the variance terms are regularized to be similar to each other (Eq. **(23)**: m below is the expected slope of how standard deviation increases with the mean—m is taken as given, and we regress to fit the intercept b of the line y = mx + b, where y is standard deviation and x is mean):

$$\text{cor} = \Sigma_{j=1 \text{ to } G} (w_j * \sigma_{jab}^2 / (\sigma_{jaa} * \sigma_{jbb})) \quad \textbf{(21)}$$

$$\sigma_{jab}^2 = \max(c_1 - (\text{iter}-1)/c_2, \text{cor}) * (\sigma_{jaa} * \sigma_{jbb}) \quad \textbf{(22)}$$

$$\text{Expected std } \sigma_{exp\text{-}j} = [(1 / G) * \Sigma_{k=1 \text{ to } G} (\sigma_k - m * \mu_k)] + (m * \mu_k) \quad \textbf{(23)}$$

$$\sigma_j = (w_j / (w_j + r)) * \sigma_j + (r / (w_j + r)) * \sigma_{exp\text{-}j} \quad \textbf{(24)}$$

where c_1 is a parameter set to be high to ensure clusters have positive correlation in the noise of the A and B dimensions, and c_2 sets how quickly this value decays as the number of iterations increases in Eq. **(22)**, and r is a regularization parameter in Eq. **(24)**.

EM is iterated for at least a minimum number of iterations until convergence (or a maximum number of iterations is reached), at which time the next model is tried.

Unobserved Cluster Imputation:
When G<3, unobserved clusters are imputed. First the EM-fit clusters are labeled with the appropriate genotype class(es), based primarily off Euclidean distance between μ_j and μ_{AA}, μ_{AB}, and μ_{BB}; however, Hardy-Weinberg principles and relative allele frequencies in HapMap can be used to help guide this process (see Manual & Pseudocode for more details). Once labeled, missing clusters are imputed using regression coefficients learned when creating the prior models file; see below.

Model Selection:
A series of heuristics are used to determine which GMM model is best (in addition to the relative probability of observing the data given the EM-fit model).

1. The standard BIC information criterion (penalizing higher-order models)
2. How closely the final means (μ_1, μ_2, μ_3) match the expected (μ_{AA}, μ_{AB}, μ_{BB})
3. How positive the cor term is
4. How close the two wingspans are in length (dist(μ_1, μ_2) versus dist(μ_2, μ_3))

5. Whether a single cluster is likely only explaining an outlier

For a description of how these terms are weighted, see Manual & Pseudocode

Genotyping:
Using the selected model, each sample i is genotyped as the copy number of the cluster j which maximizes the equation

$$P(j \mid i) = (w_j / |\Sigma_j|^{1/2}) * \exp(-(x_i - \mu_j)^T * \Sigma_j^{-1} * (x_i - \mu_j) / 2)) \quad \textbf{(25)}$$

Samples are furthermore assigned a confidence reflecting the relative likelihood of belonging to the next-best cluster (a global confidence), as well as a local confidence reflecting their absolute probability. The local confidence is computed as a sigmoidal function of the number of standard deviations away from the cluster center a sample lies. To compute the global confidence score:

$$conf_{global\text{-}i} = P(j_{second\text{-}best} \mid i) / P(j_{best} \mid i) \quad \textbf{(26)}$$

To compute the local confidence score:

$$q_i = sqrt((x_i - \mu_j)^T * \Sigma_j^{-1} * (x_i - \mu_j) / 2) \quad \textbf{(27)}$$

$$conf_{local\text{-}i} = 1/(1 + \exp(p - q_i)) - 1/(1 + \exp(p)) \quad \textbf{(28)}$$

where q is the number of standard deviations a sample is from its assigned cluster, and p is the number of standard deviations beyond which the local confidence score quickly increases (the inflection point in Eq. **(28)**).

The overall confidence score is then:

$$(c * conf_{global\text{-}i}) + ((1 - c) * conf_{local\text{-}i}) \quad \textbf{(29)}$$

where c is the relative confidence contribution parameter, $0 <= c <= 1$.

Birdseed: Manual & Pseudocode

Overview
Birdseed's function is to assign AA, AB, and BB genotype calls to each sample with two copies of a SNP, and characterize genotype- and allele-specific probe responses for each SNP.

Inputs
Normalized and summarized (median polish) intensities from a set of .cel files for each SNP allele. This takes the form of a matrix with N columns and 2*M rows (ignoring headers), where N is the number of samples to be analyzed, and M is the number of SNPs. Note: for the version of birdseed included in apt-probeset-genotype, normalization and summarization can occur on the fly, and .cel files can be input directly.

A file describing the gender of each sample. Note: for the version of birdseed included in apt-probeset-genotype, gender is calculated on the fly, and this file is unnecessary.

A models file which contains the means, variances, and proportions of discrete genotype classes observed for each SNP in HapMap. (The default models file was built using a run of the HapMap samples at Affymetrix, Inc.—file available online.)

Optionally, a file listing for each SNP which samples are not expected to have two copies and thus should be excluded from Birdseed for that SNP (typically this is a file generated from Canary results). SNPs where all samples are expected to have two copies need not be included in the file.

Optionally, a file listing which SNPs have unusual copy number for males, females, or both, and what the expected copy number is for each gender. (I.e. chrX, chrY, and mitochondrial SNPs.)

Outputs
Genotypes: 0, 1, and 2 (corresponding to AA, AB, and BB calls) at each SNP in each sample, along with a confidence score ranging from 0 (most confident) to 1 (least confident). Samples that were excluded (due to expected copy number other than two) are assigned a genotype of "-1".

Optionally, a file analogous to the models file which contains the means, variances, and proportions of discrete genotype classes observed for each SNP in the data. These represent the posterior models that characterize genotype- and allele-specific probe responses for each SNP.

Brief description of algorithm
In brief, the algorithm utilizes Expectation-Maximization as follows. The models are used for initialization. Each sample is then assigned a probability of belonging to each cluster (estimation). Next, each cluster is redefined based off the samples that belong to it (maximization), as well as being tethered to the expected location of the model. New

cluster definitions are also forced to share certain covariance properties with each other. The estimation and maximization steps are iterated until convergence, at which point one can assign the likelihood of the model (the final Gaussian parameters). The likelihood of the model is dependent on how well the model explains the observed data as well as how well the model fits certain expectations (for example, that the clusters are evenly spaced). Birdseed chooses between models built from different initializations and between 1, 2, and 3 clusters explaining the data. If the best model has fewer than 3 clusters, genotype classes corresponding to clusters not in the model are imputed to increase sensitivity to rare genotypes. The resulting 3 clusters represent the probe responses to each genotype class on the particular batch being run. Special considerations are used on the X, Y, and mitochondrial chromosomes.

Description of algorithm

Below is a pseudocode sketch of the algorithm. All code is open-source and can be downloaded free of charge from www.broad.mit.edu/mpg/birdsuite. For additional detail, see the actual source code.

There are a large number of flags and constants that can be adjusted according to the user's preference. However, the default values have been tested and are appropriate for the vast majority of typical inputs.

--std_slope: (m in Eq. **(23)**). Expected slope of cluster standard deviation versus cluster mean intensity. Default: 0.062. Each SNP has 3 clusters (representing AA, AB, and BB classes). One expects the variance in the A dimension to be larger for clusters in which the A mean is larger. This slope explains that expected relationship. The default was empirically derived using non-polymorphic probes for which clustering is unnecessary. Permissible range: [0,Inf)

--epsilon: Tolerance at which to stop EM. Default: 0.001. Permissible range: (0,1)

--var_start: Variances are initialized to be var_start times the expected variances. Default: 1.1. Permissible range: (0,Inf)

--cluster_distance_ratio_cutoff: The ratio of adjacent cluster means in each direction must exceed this value. Default: 0.85. Setting this to 1.0 would prevent, for example, the AB cluster having a higher mean intensity in the A dimension than the AA cluster, or a higher mean intensity in the B dimension than the BB cluster. Permissible range: [0,Inf)

--merged_cluster_threshold: If this distance between two clusters gets lower than this, EM is aborted. Default: 0.025. Permissible range: [0,Inf)

--small_cluster_weight_threshold: When EM is fitting a 3-cluster model, the log-likelihood is penalized by small_cluster_penalty if the weight of any cluster is less than this threshold. Default: 0.01. Setting this higher can decrease the probability of outliers defining their own cluster. Permissible range: [0,1]

--small_cluster_penalty: Default: 10.0. Permissible range: (-Inf,Inf)

--low_hom_weight_fraction: When EM is fitting a 2-cluster model (where the clusters represent either AA and AB, or AB and BB), a cluster cannot be assigned to a homozygous class if its weight is below low_hom_weight_fraction (note: modified by low_hom_sample_inflation). Default: 0.5. Permissible range: [0,1)

--low_hom_sample_inflation: When EM is fitting a 2-cluster model and the number of samples analyzed is small, one might expect the homozygous class could have a lower weight than otherwise expected. Therefore the low_hom_weight_fraction is multiplied by the number of samples divided by (the number of samples plus low_hom_sample_inflaction). Default: 100. Permissible range: [0, Inf)

--starting_cluster_weight: All clusters are assigned to have at least this weight. Default: 0.05. Permissible range: [0,0.33333]

--expected_wingspan_ratio: A wingspan is defined as the distance between a homozygous cluster and the heterozygous cluster. The log likelihood is penalized by unbalanced_wingspan_penalty if the ratio of wing spans is above this number. Default: 1.15. Permissible range: [1,Inf)

--unbalanced_wingspan_penalty: Default: 5.0. Permissible range: [-Inf,Inf]

--min_covar: The covariance term reflecting the correlation of variance in the A dimension and variance in the B dimension is not allowed to be below this. Default: -0.7. Permissible range: (-1,1)

--max_covar2: The covariance term reflecting the correlation of variance in the A dimension and variance in the B dimension is not allowed to be above this. Default: 0.95. Permissible range: (-1,1)

--max_covar1: c_1 in Eq. **(22)**. The covariance term is forced to be at least max_covar1 minus the EM iteration number divided by covar_floor_decay for the first covar_floor_decay iterations. This ensures clusters begin with positive correlation (which is expected due to cross-hybridization) as opposed to negative correlation (which can happen by a single cluster describing more than one genotype class). Default: 0.9. Permissible range: (-1,1)

--covar_floor_decay: : c_2 in Eq. **(22)**. Forced positive covariance decays over this iteration scale. Default: 8. Permissible range: [0,max_iter]

--max_iter: Stop EM after max_iter iterations. Default: 50. Permissible range: [1,Inf)

--low_covar_threshold: If the covariance term is below low_covar_threshold, penalize the log likelihood of the model by low_covar_penalty * (low_covar_threshold-covar). Default: 1.0. Permissible range: [-1,1]

--low_covar_penalty: How much to penalize a covariance term lower than low_covar_threshold. Default: 15.0. Permissible range: (-Inf,Inf)

--wing_length_delta_penalty: How much to penalize the log likelihood of a model based on differences between the prior model's winglengths and the posterior model's winglengths. Default: 50.0. Permissible range: (-Inf,Inf)

--mean_dist2: If two neighboring clusters have means that are this close, the log likelihood of the model is penalized. Default: 1.2. Permissible range: [0,Inf]

--lambda3: How much the log likelihood of models having close neighboring clusters are penalized. Default: 2.5. Permissible range: [0, Inf]

--bic_weight: How much to penalize the log likelihood of a model based on the number of clusters the model fit. Default: 1.0. Permissible range: (-Inf,Inf).

--anchor_weight: How strongly clusters are anchored to the priors when fitting 3 clusters with EM, expressed in number of pseudocounts (not a percentage). (s in Eq. **(17)**).Default: 1.0. Permissible range: [0,Inf)

--max_anchor_percentage: When fitting 3 clusters, clusters are anchored using the

minimum of anchor_weight and max_anchor_percentage*numsamples pseudocounts. Default: 5.0. Permissible range: [0,Inf)

--cluster_variance_regularization_factor: Determines how much variances are regularized to be fit by a single term in each dimension, as opposed to fit separately for each cluster. Default: 1.0. Permissible range: [0,Inf)

--var_mult: If a cluster is imputed (as opposed to fit directly with EM), the expected variance of the cluster is multiplied by (var_mult^2). This helps recover rare genotype classes. Default: 1.2. Permissible range: [0,Inf).

--hom_hom_penalty. When fitting a model with only 2 clusters, penalize the assignment of those 2 clusters to the AA and BB classes by hom_hom_penalty. This is because one does not typically expect to see the examples of each homozygous state without also observing the heterozygous state. Default: 2.1. Permissible range: [1,Inf).

--mono_het_penalty. When fitting a model with only 1 cluster, penalize the assignment of that 1 cluster to the AB class by mono_het_penalty. One does not typically expect to only observe heterozygous samples without any homozygous samples as well. Default: 999999999. Permissible range: [1,Inf).

--allow_unlikely_clusters: Allows a 2-cluster model to be assigned to AA and BB genotype classes, as well as allowing a 1-cluster model to be assigned to the heterozygous genotype class. Default: true

--disallow_unlikely_clusters: Does not allow a 2-cluster model to be assigned to the AA and BB genotype classes, nor does it allow a 1-cluster model to be assigned to the heterozygous genotype class. Mutually exclusive with --allow_unlikely_clusters. Default: false

--two_cluster_low_observation_penalty_factor: When fitting a model with 2 clusters, penalize an alignment that indicates the observed homozygous cluster was rare in the input models file. The penalty is (this factor + total num observations in input model) / (this factor + num observations of the observed homozygous cluster in input model). Thus, large numbers remove dependence on input allele frequencies, while small numbers increase such a dependence. Default: 10. Permissible range: [0, Inf).

--final_weight_min: After EM, when assigning discrete genotype classes to samples, assume each cluster has a weight at least equal to final_weight_min. Default: 0.333. Permissible range: [0,1].

--relative_distance_confidence_weight: When assigning discrete genotype classes to samples, there are 2 inputs into the confidence of the assignment: The relative likelihood of a sample coming from the second-best assignment, and the absolute likelihood a sample comes from the best assignment. The larger this weight, the more the confidence is determined by the former input. Default: 0.8. (c in Eq. **(29)**). Permissible range: [0,1].

--std_inflection_point: When assigning discrete genotype classes to samples, this determines the function that relates absolute likelihood a sample comes from the best assignment to a confidence score. Default: 4.0. (p in Eq. (**28)**). Permissible range: [0, Inf]

--correction-factor: Use the supplied value to transform prior models into the same intensity space as the samples. Default: not used. Permissible range: (0,Inf)

--snp_specific_correction_factor: Determine the value to transform prior models into the same intensity space as the samples on the fly for each SNP, using the sample intensity data itself. Default: enabled. Disable by specifying either --correction-factor or

--average-correction-factor.

--average_correction_factor: Determine the value to transform prior models into the same intensity space as the samples once for the entire dataset, using the mean intensity of the entire input dataset. Default: disabled.

Conventions and terms (extrapolate where necessary):

EM refers to Expectation-Maximization

N refers to the number of samples

I_{ai} refers to the intensity of allele A of sample i

I_{bi} refers to the intensity of allele B of sample i

μ_{AAa} refers to the expected mean of allele A of the AA cluster from the prior models

μ_{ABb} refers to the expected mean of allele B of the AB cluster

μ_{2a} refers to the mean of allele A of the 2nd cluster (not yet assigned to a genotype class; a cluster that will become a posterior model for a genotype class)

μ_3 refers to the 2d point (μ_{3a}, μ_{3b})

σ_{AAa} refers to the expected standard deviation in the allele A dimension of the AA cluster

σ_{ABb} refers to the expected standard deviation in the allele B dimension of the AB cluster

σ_{2a} refers to the standard deviation in the allele A dimension of the 2nd cluster

cor refers to the shared covariance term representing the correlation of noise in the A and B dimensions across all clusters.

Σ_3 refers to the 2d covariance matrix:

$[[\sigma_{3a}^2, cor*\sigma_{3a}*\sigma_{3b})]$

$[cor*\sigma_{3a}*\sigma_{3b}), \sigma_{3b}^2]]$

w_{AA} refers to the expected weight of the AA cluster

w_{AB} refers to the expected weight of the AB cluster

w_2 refers to the weight of the 2nd cluster

Σ_i refers to a summation of the following expression, over all samples from i = 1 to N

Pseudocode:

For each SNP:

I. Only consider samples expected to have 2 copies at that SNP (exclude samples based on gender/SNP location and based on exclusions input from Canary).

II. Scale the prior model to be in the space intensity space as the samples. By default, this scale is calculated separately for each SNP as the average distance of a sample from the origin divided by the weighted average of the prior model means from the origin (as in Eq.s **(12)-(14)**.

III. Fit 4 potential models to the data.

A. The data came from a single cluster

i. Determine the parameters of this clusters

$\mu_{1a} = \Sigma_i I_{ai} / N$

$\mu_{1b} = \Sigma_i I_{bi} / N$

$\sigma^2_{1a} = \Sigma_i (I_{ai} - \mu_{1a})^2$

$\sigma^2_{1b} = \Sigma_i (I_{bi} - \mu_{1b})^2$

$cor = (\Sigma_i (I_{ai} - \mu_{1a})* (I_{bi} - \mu_{1b})) / (\sigma_{1a} * \sigma_{1b})$
$w_1 = 1 - 2*$starting_cluster_weight

ii. Determine which genotype class this cluster corresponds with
Choose the genotype class whose prior mean expectation has the smallest Euclidean distance to μ_1. Setting --disallow-unlikely-clusters stops the closest mean from being the AB class. The distance to the AB class is other multiplied by mono_het_penalty.

iii. Impute clusters corresponding to remaining genotype classes
1) Imputation is done using regression parameters learned during prior model generation for the mean. See below.
The final clusters are taken as a weighted average of the above and the original prior expectation of where each cluster should lie *including* the cluster fit with the supplied data. The weights for the averaging depend on --anchor_weight and --max_anchor_percentage.
2) Imputation of variances is as follows:
Standard deviation of a cluster along a particular dimension is assumed to increase proportional the mean of the cluster. The slope of this increase is supplied (see --std_slope). The intercept is allowed to vary, and is fit using regression given the non-imputed covariance matrices. This intercept along with std_slope is then used to derive the expected variances for each cluster given its mean.

B. The data came from 2 clusters
i. Invoke an EM algorithm to determine the parameters of these clusters
1) Initialize the clusters
$\mu_1 = \mu_{AA}$
$\mu_2 = \mu_{BB}$
$\sigma^2_{1a} = \sigma^2_{2a} = (2/3) * (\sigma^2_{1a}$ from IIIAi)
$\sigma^2_{1b} = \sigma^2_{2b} = (2/3) * (\sigma^2_{1b}$ from IIIAi)
$cor = 0.0$
$w_1 = w_2 = 0.5$
2) While the model continues to explain the data better:
a) Estimate the membership of each sample to each cluster (Standard E step)
b) Estimate the parameters of each cluster with the following modifications to the standard M-step:
There is a single covariance term representing the correlation of noise in the A and B dimensions, calculated as the weighted average of this term in each cluster.
The covariance term is forced to be positive during the first few iterations. (See --max_covar1).
If the means of two clusters get too close, EM is

aborted.
Variances are regularized to be similar to each other as follows: The expected variance is calculated as in IIIAiii2, using the typical EM-fit variances to determine std_intercept. The final variance is a weighted average of the original fitted variance and the expected variance.
(See --cluster_variance_regularization_factor.)

ii. Determine which two genotype classes the clusters represent
This determination is dependent on:
the Euclidean distance between the clusters and the prior expectation
Hardy-Weinberg principles (see --low_hom_sample_inflation)
relative allele frequencies in HapMap
(see --two_cluster_low_observation_penalty_factor)
Whether an AA/BB is sought (see --allow_unlikely_clusters)

iii. Impute cluster corresponding to the remaining genotype class
Equivalent to IIIAiii

C. The data came from 3 clusters
i. Invoke an EM algorithm to determine the parameters of these clusters
1) Initialize the clusters
$\mu_1 = \mu_{AA}$
$\mu_2 = \mu_{AB}$
$\mu_3 = \mu_{AB}$
$\Sigma_1 = \Sigma_{AA}$
$\Sigma_2 = \Sigma_{AB}$
$\Sigma_3 = \Sigma_{BB}$
cor = weighted mean of cor term in $\{\Sigma_{AA}, \Sigma_{AB}, \Sigma_{BB}\}$
$w_1 = w_2 = w_3 = 1/3$
2) Same as IIIBi2, with the following additional modification to the M-step:
a number of pseudopoints are placed at the expected mean for each cluster. The pseudopoints placed at μ_{AA} are forced to belong to cluster 1, the pseudopoints placed at μ_{AB} are forced to belong to cluster 2, and the pseudopoints placed at μ_{BB} are forced to belong to cluster 3. See --anchor_weight and --max_anchor_percentage
ii. Cluster 1 is forced to represent the AA cluster, cluster 2 the AB cluster, and cluster 3 the BB cluster.
iii. No cluster imputation is necessary, since we fit all clusters with EM

D. Identical to C, except for the initialization step which is as below:
1) Initialize the clusters

μ_{1a} = maximum A allele intensity observed in samples
μ_{1b} = minimum B allele intensity observed in samples
μ_{3a} = minimum A allele intensity observed in samples
μ_{3b} = maximum B allele intensity observed in samples
$\mu_{2a} = (\mu_{1a} + \mu_{3a}) / 2$
$\mu_{2b} = (\mu_{1b} + \mu_{3b}) / 2$
$\Sigma_1, \Sigma_2, \Sigma_3$ are calculated equivalent to the way that expected variance is calculated in IIIAiii2, using Σ_{AA}, Σ_{AB}, and Σ_{BB} as input.
Σ_1 and Σ_3 are initialized to be yet var_start times larger
However, Σ_2 is initialized to be var_start / 20 as large.
cor = 0
$w_1 = w_2 = w_3 = 1/3$

IV. Select the best model (one of IIIA, IIIB, IIIC, or IIID). This is primarily dependent on the ability of the model to explain the data (that is, the relative probability of observing the data given each model), but models are also penalized on the following criteria to ensure the final clustering matches reasonable expectations about the shape and relative distribution of the AA, AB, and BB clusters:

The standard BIC information criterion (penalizing higher-order models)
How closely the final means (μ_1, μ_2, μ_3) match the expected (μ_{AA}, μ_{AB}, μ_{BB})
How positive the cor term is
How close the two wingspans are in length (dist(μ_1, μ_2) versus dist(μ_2, μ_3))
Whether a single cluster is likely only explaining an outlier

See descriptions of parameters for more explanation

V. Output the clusters corresponding to the best model

VI. Assign an AA, AB, or BB genotype to each sample, as well as a confidence score reflecting the certainty of the assigned genotype

A. The log-likelihood of observing a sample i given a genotype x is calculated

B. Sample i is assigned to genotype x resulting in the largest log-likelihood

C. Sample i is assigned a confidence corresponding to the weighted average of the relative likelihood of the sample's intensity given a different genotype (compared to the assigned genotype) and the absolute likelihood of the sample's intensity given the assigned genotype, as in Eq.s **(26)-(29)**.

Modifications for X, Y, and mitochondrial chromosomes:

0) On the X chromosome, females are clustered as above. Males are clustered separately as above, with the following modifications. On the Y chromosome, males are clustered as above, with the following modifications. On the mitochondrial chromosomes, all samples are clustered together, with the following modifications:

1) A special prior model line exists in the prior models file describing the location of the A/null and B/null clusters. This is used in lieu of the typical prior.

2) Only allow clusters to correspond to homozygous genotype classes in step IIIAii.

3) In step IIIB, parts ii and iii are skipped.

4) Do not attempt step III-C or III-D above (fitting the data with 3 clusters)

Generation of "prior models" file for Birdseed:

The 270 canonical HapMap samples were genotypes on the Affymetrix SNP5.0 and SNP6.0 platforms. The resulting CEL files were quantile normalized and data from all SNPs were summarized using median polish, affording two allele intensity measurements per sample per SNP for the A and B alleles. For most of these SNPs, HapMap genotypes are available. (When no genotyping data is available, see below.) Using the HapMap calls, the sample data can be further summarized into "cluster" data. The two-dimensional mean of all samples of a given genotype can be computed as well as their covariance matrix. The number of observations that contributed to a given cluster are also recorded.

Data from (up to 50,000) SNPs in which all genotypes in each of the three clusters (AA, AB, and BB) are observed more than a defined number of times (num_points, typically equal to 6) are stored. We designate these SNPs as "fully observed." These data become the basis of the regression equations to infer the locations of unobserved clusters in other SNPs.

Five separate cases of predictions are possible in SNPs that are not fully observed:

Case 1: Only AA observed, predict AB and BB
Case 2: Only AA and AB observed, predict BB
Case 3: Only AB and BB observed, predict AA
Case 4: Only BB observed, predict AA and AB
Case 5: Only AA and BB observed, predict AB (very rare)

Regarding cluster position, simple linear regression was performed on the cluster centers on the case of the fully observed SNPs. This is possible since all clusters have been observed, making the fully observed SNPs suitable as training data. Covariances are handled identically except the log of the covariance was used for on-diagonal terms while the signed square root was used for off-diagonal terms. The quality of these regression equations can be tested with a simple r^2 metric. (The left hand side of each prediction equation designates which clusters were observed. The right hand side designates which clusters are being predicted. Underscore followed by a single letter designates prediction of a mean, while underscore followed by two letters designates a covariance.)

Prediction	r^2
AA -> AB_a	0.996
AA -> AB_b	0.952
AA -> BB_a	0.854
AA -> BB_b	0.943
AA + AB -> BB_a	0.946
AA + AB -> BB_b	0.998
AB + BB -> AA_a	0.998
AB + BB -> AA_b	0.941

BB -> AA_a	0.941
BB -> AA_b	0.852
BB -> AB_a	0.952
BB -> AB_b	0.996
AA + BB -> AB_a	0.998
AA + BB -> AB_b	0.998
AA -> AB_aa	0.993
AA -> AB_ab	0.893
AA -> AB_bb	0.987
AA -> BB_aa	0.979
AA -> BB_ab	0.573
AA -> BB_bb	0.977
AA + AB -> BB_aa	0.987
AA + AB -> BB_ab	0.636
AA + AB -> BB_bb	0.990
AB + BB -> AA_aa	0.990
AB + BB -> AA_ab	0.635
AB + BB -> AA_bb	0.987
BB -> AA_aa	0.976
BB -> AA_ab	0.565
BB -> AA_bb	0.979
BB -> AB_aa	0.986
BB -> AB_ab	0.892
BB -> AB_bb	0.840
AA + BB -> AB_aa	0.994
AA + BB -> AB_ab	0.913
AA + BB -> AB_bb	0.995

As can be seen, the regression equations furnish good predictive power. Hence the fully observed SNPs furnish a set of regression equations, which were applied to SNPs where at least one genotype class was not observed. The directly observed genotype clusters combined with those inferred using the regression were used to furnish a map of "prior models" for all SNPs on the SNP5.0 and SNP6.0 microarrays. Since the number of observations of each genotype class is also output in prior models file, one can determine which clusters were inferred, and which have parameters estimated directly from data.

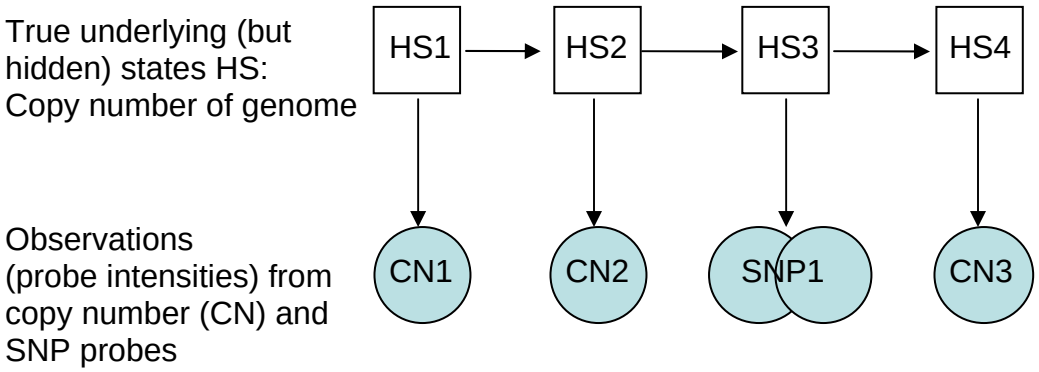
In the case where no genotype data was available (and thus not even a single genotype class was observed and labeled), the model generated for that SNP was simply the mean of the fully observed SNPs, the so-called "grand mean" and "grand covariance." This generic model was necessary because Birdseed requires a model for each SNP genotyped.

Birdseye: Algorithm

Overview
Birdseye's function is to discover regions of variable copy number, especially those that are rare or *de novo*.

Description of algorithm
Birdseye is a Hidden Markov Model (HMM) to find regions of variable copy number in a sample.

Model:
The hidden state is the true copy number of the individual's genome; the observed states are the normalized intensity measurements of each probe on the array.



Modeling Emission Probabilities for CN probes for HS=2, HS=0,1,3,4:
For each copy number probe, we empirically estimate the emission probability of intensities for an underlying hidden state of 2 copies as a normal distribution with mean and variance equal to the trimmed mean and variance of the intensities of all samples in the batch (excluding those already determined to be copy-variable via Canary, as well as 10% of samples with the highest intensities and 10% of samples with the lowest intensities.) Given this normal distribution, we impute the emission probability of intensities for an underlying state of 0 or 1 copies as normal distributions whose mean and variance properties are determined using regression parameters learned from a combination of probes on chromosome X (for which we know males have 1 copy and females 2 copy) and single SNP alleles (for which BB samples have 0 copies of allele A, AB samples have 1 copy of allele A, and AA samples have 2 copies of allele A). The emission probabilities of intensities for an underlying state of 3 or 4 copies is then imputed assuming the differences in like parameters between the model for each copy number state increases as a power law.

Modeling Emission Probabilities for SNP probes for HS=2, HS=0,1,3,4:
For each SNP the hidden state actually includes allelic information in addition to copy number. For this reason (and because we use the Viterbi algorithm to determine the most

probable path), the emission probability of intensities for an underlying hidden state of 2 copies is a union of the three normal distributions specified by Birdseed that model the AA, AB, and BB clusters; the HMM utilizes whichever of these distributions results in the maximal likelihood as the emission probability for an underlying state of 2 copies. The means and variances of the BB, AB, and AA clusters encode SNP-specific information regarding the intensity of each allele in response to 0, 1, and 2 copies of the A allele, respectively, and similarly for 2, 1, and 0 copies of the B allele; these are used directly to estimate the means and variances of the null, A, B, AAB, ABB, and AABB clusters. (See "Extrapolating a Cluster Set" in Fawkes section). As with copy number probes, increasing dosages of each allele are then imputed using a power law.

Sample-specific Emission Probabilities:
The probe-specific emission probabilities above are furthermore modified by a sample-specific factor. For each sample, the average distance from the expected cluster mean (for which expected copy number mean is used, see "Modeling Transition Probabilities" below) is computed in terms of standard deviations. For most samples, this should be approximately 1. However, for noisy samples, their intensities will tend to be farther from the mean than expected, and can be significantly greater than 1. The standard deviations of each emission probability density function is thus multiplied by this sample-specific metric. This metric also provides a good measure of sample-chromosome quality; noisy sample-chromosomes should be discarded from downstream analyses.

Modeling Transition Probabilities:
The transition probabilities between underlying copy number states is asserted such that transitioning out of a state reflecting normal copy number (typically 2, but varying for the sex chromosomes) is low, while transitioning within the same state or returning to normal copy number is relatively high. For each chromosome, first a global copy-number of that chromosome is estimated across all probes; this is used to determine the normal copy number of that chromosome for that sample. We note the algorithm is fairly robust to reasonably variations in the transition matrix, and that events can be filtered based on associated LOD scores such that purposefully setting permissive transition probabilities may be logical.

Furthermore, transition probabilities are dependent on the genomic distance between two consecutive probes. The transition penalty is computed as follows, where d is the distance, N is normal copy number, and X and Y are different abnormal copy numbers:
$p(\text{switch N to X}) = 0.005 * (1 - e^{-\text{dist} / 20000})$
$p(\text{switch X to N}) = 0.5 * (1 - e^{-\text{dist} / 20000})$
$p(\text{switch X to Y}) = 0.0025 * (1 - e^{-\text{dist} / 20000})$
The probability of self-transitions (N to N, or X to X) are computed as 1 minus the probability of transitions to other copy numbers. The transition probabilities (0.5% and 50%) and length constant (20kb) are chosen based on the approximate number and size of CNVs observed per genome as observed in the accompanying manuscript SAM, FKG et al.

Tracing a Path of Copy Number States of the Genome:
The emission probabilities and the transition matrix are combined to find a path $S = \{s_1, s_2 .. s_n\}$ representing the state at each probe that maximizes the (log) probability of observing the data

$$\log(P(x_1 .. x_n)) = \Sigma_{i=1 \text{ to } n} [\log(P(x_i \mid s_i)) + \log(P(s_i \mid s_{i-1}))] + \log(P(\text{state}==2 \mid s_n)) \quad \textbf{(30)}$$

where the first term in the sum $[\log(P(x_i \mid s_i))]$ represents the log of the relative probability of the observed intensity given the state (and equivalent to the formulae given above in the Canary and Birdseed sections), the second term in the sum $[\log(P(s_i \mid s_{i-1}))]$ represents the log of the probability of transitioning to the current state given the previous state (s_0 is assumed to be normal copy number), and the last term outside the sum $[\log(P(\text{state}==2 \mid s_n))]$ forces a transition to end at normal copy number. The maximization is carried out using the standard Viterbi algorithm. The resulting path V can then be segmented into regions of continuous copy number, and these segments are output.

Assigning a Score to Each Discovered CNV:
The LOD score for a given segment is the log of the relative likelihood of the path including the event ($S_i = \{F_{1i}, C_i, C_i, C_i, \ldots F_{2i}\}$, where F_{1i} reflects the left flanking copy number, F_{2i} the right flanking copy number, and C_i the copy number of the segment) versus the path that maintains consistent copy number with one or both of the flanking regions to the event (the maximum of $S_i = \{F_{1i}, F_{1i}, F_{1i}, F_{1i}, \ldots F_{2i}\}$ and $S_i = \{F_{1i}, F_{2i}, F_{2i}, F_{2i}, \ldots F_{2i}\}$). The relative likelihood of a path S_i is computed as above.

Birdseye: Manual & Pseudocode

Overview
Birdseye finds ranges of constant copy number in each sample, operating one chromosome at a time. The breakpoints of these segments is output, along with a LOD score reflecting the probability of the event. Birdseye can also be used to assign a copy number state without segmentation, useful for testing previously discovered CNVs in new samples (for example, to determine if a CNV found in a child is also present in one of his parents). However, this functionality is not described in detail here. See the website (http://www.broad.mit.edu/mpg/birdsuite/) for more details.

Inputs
Normalized and summarized (median polish) intensities from a set of .cel files for each SNP allele on a single chromosome, in chromosomal order. This takes the form of a matrix with N columns and $2*M_s$ rows (ignoring headers), where N is the number of samples to be analyzed, and M_s is the number of SNPs. Required header columns include the chromosomal location of each SNP.

Normalized intensities from a set of .cel files for each CN probe on a single chromosome, in chromosomal order. This takes the form of a matrix with N columns and M_c rows (ignoring headers), where N is the number of samples to be analyzed, and M_c is the number of CN probes. Required header columns include the chromosomal location of each CN probe.

A file with M_s rows describing the batch-specific, allele-specific probe responses to normal copy number for SNPs (i.e. the posterior clusters output from Birdseed).

A similar file with M_c rows describing the batch-specific probe responses to normal copy number for CN probes.

Outputs
A file with one row per segment of constant copy number for each sample. Each segment is annotated with the sample number, the copy number state, the chromosome, the start breakpoint, the end breakpoint, and the score.

Brief description of algorithm
In brief, the algorithm is a Hidden Markov Model to determine copy number. While the each sample is independent of all other samples, all samples in a batch are processed concurrently for speed and ease of code. The input files represent emission probabilities of probes to a sample with 2 copies. Emission probabilities of probes to samples of 0, 1, 3, and 4 copies are imputed. This emission probability for a given sample is modified by the apparent noise of that sample. A particular transition matrix specifying the probability of transitioning within the same copy number state (high probability) and between copy number states (low probability) is assumed; copy number states are assumed to be discrete and in the range [0,maxCopyNumber]. The Viterbi best path of

copy number states is then calculated for each sample. Each discrete segment of the path is then scored for how much it increases the probability of observing the given data versus if no transition occurred at one or both of its breakpoints. The segments and scores are output for use in Fawkes and for CNV studies.

This algorithm differentiates itself from other CNV HMMs through seamless integration of CN and SNP probes, leaving the data in the original intensity space (so SNPs retain 2-dimensional information), batch- and probe-specific characteristics (so inherently noisier probes are effectively downweighted), sample-specific characteristics (so false positive rates should not vary wildly from one sample to the next), and the ability to assign a score to each reported event (such that the precise values of the transition matrix are less important, since one can filter events post-Birdseye).

Options
maxCopyNumber: The maximum copy number state to test each sample as having. Default: 4. Permissible range: [0, Inf)

Pseudocode
I. Load all data.

II. Calculate sample-specific variance correction factor. This is calculated as the square of the average distance represented in standard deviations of a sample's intensity from the mean of a CN probe. However, it is not allowed to vary below 4/9 or above 9/4. This prevents samples with abnormal amounts of copy number variation on a chromosome from being vastly undercalled.

III. Assert the transition matrix as follows:

The probability of transitioning from copy number two to any other copy number is 0.005 * (1-e^(-distance_between_probes / 20000)).

The probability of transitioning from a copy number other than two to copy number two is 0.5 * (1-e^(-distance_between_probes / 20000)).

The probability of transitioning from a copy number other than two to a different copy number that is also not two is 0.0025 * (1-e^(-distance_between_probes / 20000)).

The probability of remaining at the same copy number can then be filled in such that the sum of transition probabilities from a particular copy number state totals 1.

These values approximate the number and size of CNVs we expect to see in a sample. In practice, the algorithm is robust to reasonable variation of these values, and since CNVs are assigned scores reflecting their probability, filtering can also happen downstream. We do not that these values may be inappropriate for different platforms that have more or fewer probes per unit distance along the chromosome. For example, an array tiled more densely with probes should have a lower probability of transitioning between copy number state. This transition matrix may also benefit from a change of parameters for the case of samples expected to have a large number of transitions (e.g. samples taken from tumors).

IV. Compute the Viterbi best path using standard HMM techniques.

A. The sample is forced to begin with copy number = 2.

B. At each successive locus, the relative log likelihood of the sample's intensity at the locus is computed for each potential copy number state as a point estimate.

i. For a given SNP, copy-variable emission probabilities are computed as detailed below in Fawkes-->extrapolating a cluster set.

ii. For a given CN probe, copy-variable emission probabilities are computed using a regression built from chromosome X. For probes on chromosome X that show differential signal between males and females, simple linear regression was able to fit the male (1-copy) mean intensity given the female (2-copy) mean intensity with 5% standard error. These regression coefficients are used to impute the 1-copy emission probability of a CN probe given the 2-copy emission probability (that is an input to the algorithm). Emission probabilities for other copy number states are imputed by assuming a power law whose order was determined by using SNP data in a single allele (where samples of various genotype then represent 0, 1, or 2 copies of that allele).

iii. Due to gross artifacts on the chip (such as scratches), the data is not necessarily Gaussian, and huge outliers can occur. To minimize their impact, no sample is ever considered to be more than 3 standard deviations away from any given Gaussian cluster. Since noise varies from sample to sample, the variance associated with each cluster is multiplied by the value calculated in II.

iv. The relative log-likelihood of each copy number state is computed. For SNPs whose emission probability of a given copy number is represented by multiple clusters/gaussians, the maximum of the log-likelihoods from any of these clusters is used as the log-likelihood of that copy number state.

C. The total log-likelihood of each copy number state at the current locus is defined as the sum of:

i. The total log-likelihood at the previous locus of the copy number state that was transitioned from.

ii. The log-likelihood of the particular transition.

iii. The point estimate of the log-likelihood of each copy number state at the current locus as computed in B.

D. The path that maximizes the total log-likelihood at the current locus as computed in C is stored.

E. After reaching the end of the chromosome, the sample is forced to transition to copy number state = 2. We once again store the path that maximizes the total log-likelihood.

F. The Viterbi best path is traced back along the chromosome.

V. For each segment of consistent copy number, compute the relative log-likelihood of the data given a change in copy number compared to the log-likelihood of the data if the given segment were the same as that of the flanking regions. This represents the LOD score associated with each event. (This is not computed for events spanning more than 1000 probes for efficiency.)

Fawkes

Overview
Fawkes produces SNP genotypes of the form {n, m, c}, in which n is the non-negative integer number of copies of the A allele, m is the non-negative number of copies of the B allele, and c is the floating-point confidence of this call, where 0 is most confident and 1 is least confident.

Inputs
Fawkes reads the following inputs:

- Birdseed cluster sets: these are the diploid cluster sets that birdseed found for all the SNPs for which diploid calls were made, plus synthesized diploid cluster sets for all SNPs that birdseed processed but did not make diploid calls. The cluster sets file also contains haploid cluster sets for the SNPs for which birdseed made haploid calls.
- Copy number ranges: For each sample, a list of {genomic range, copy number}. Any genomic range that is not covered by an entry in this list with a confident score is assigned a no-call. (These data are produced by blending Canary and Birdseye output.)
- Locus for each SNP. Note that some SNPs do not have a locus and are handled specially.
- Gender of each sample to be called.
- Special SNPs: List of SNPs of unusual copy number, along with the expected copy number for each SNP for males and females.
- Allele summaries for each SNP-sample combination.

Processing
For each SNP:

1. Load the diploid cluster set for the SNP. Calculate various values based on the diploid cluster set that will be used to extrapolate other cluster sets. (c.f. Calculating Cluster Metadata below.)
2. If a haploid cluster set exists for the SNP, load it also.
3. Load the SNP locus, if it exists.
4. If the SNP locus is not loaded, determine the expected copy number for males and the expected copy number for females for the SNP.
5. For each sample:
    i. If the SNP locus was loaded, look up the copy number for this sample and locus in the copy number range data.
    ii. If the SNP locus was not loaded, look up the gender of the sample and determine the expected copy number for this SNP and gender.
    iii. Extrapolate the cluster set for the copy number for this sample using the values calculated in step (a) above. (c.f. Extrapolating a Cluster Set below.)

iv. Read the A and B allele summaries for the sample, and determine the call and confidence as compared to the cluster set using the same algorithm that birdseed uses.

Calculating Cluster Metadata

When starting to process a SNP, several values are calculated to facilitate extrapolating cluster sets. The diploid cluster set can be viewed as defining, for both the A and B alleles, the intensity at copy number 0, 1 and 2.

For each of the two alleles, the mean ratio μ_r is stored, i.e.

$\mu_{ra} = (\mu_{AAa} - \mu_{ABa}) / (\mu_{ABa} - \mu_{BBa})$ **(31)**

$\mu_{rb} = (\mu_{BBb} - \mu_{ABb}) / (\mu_{ABb} - \mu_{AAb})$ **(32)**

The variance of a cluster is assumed to increase as the mean increases. For each of the two alleles, the slope of the standard deviation σ_s is stored:

$\sigma_{sa} = (\sigma_{AAa} - \sigma_{ABa}) / (\mu_{ABa} - \mu_{BBa})$ **(33)**

$\sigma_{sb} = (\sigma_{BBb} - \sigma_{ABb}) / (\mu_{BBb} - \mu_{BBb})$ **(34)**

The underlying A and B allele frequencies w_a and w_b are estimated using the frequencies of the AA, AB and BB clusters:

$w_a = 1 - w_b = ((2 * w_{AA}) + w_{AB}) / 2$

The covariance term "cor" from one cluster of the diploid cluster set is stored as in Eq. **(21)**. This is assumed to be the normalized covariance for all clusters.

Extrapolating a Cluster Set

If the copy number of a sample for a SNP is n, then n+1 clusters are created, corresponding to the genotypes A=n,B=0; A=n-1,B=1; …; A=1,B=n-1; A=0,B=n

The mean of each of these clusters is determined by induction, where C below represents the copy number, and μ_{aC} represents the mean of allele A when there are C copies of that allele :

μ_{a0}, μ_{a1}, and μ_{a2} are encoded directly in the SNP information:

$$\mu_{a0} = \mu_{BBa} \; ; \; \mu_{a1} = \mu_{ABa} \; ; \; \mu_{a2} = \mu_{AAa}$$

When C > 2, the following formula is used inductively:

$$\mu_{aC} = \mu_{a(C-1)} + \mu r_a * (\mu_{a(C-1)} - \mu_{a(C-2)}) \quad \textbf{(35)}$$

Similarly, μ_{b0}, μ_{b1}, … μ_{bG} are computed for allele B.

The variance for each cluster and allele is similarly computed:

$$\sigma_{a0} = \sigma_{BBaa} \; ; \; \sigma_{a1} = \sigma_{ABaa} \; ; \; \sigma_{a2} = \sigma_{AAaa}$$

$$\sigma_{aC} = \sigma_{a(C-1)} + \sigma_{sa} * (\mu_{a(C)} - \mu_{a(C-1)}) \quad \textbf{(36)}$$

Similarly, σ_{b0}, σ_{b1}, … σ_{bG} are computed for allele B.

For any genotype j, with N copies of allele A, and M copies of allele B, the cluster is then computed as follows:

$$\mu_j = (\mu_{aN} , \mu_{bM}) \quad \textbf{(37)}$$

$$\Sigma_j = \quad [[\sigma_{aN}^2, cor*\sigma_{aN}*\sigma_{bM})] \quad [cor*\sigma_{aN}*\sigma_{bM}), \sigma_{bM}^2]] \quad \textbf{(38)}$$

$$w_j = w_a^N * w_b^M * (n+m-1)! / ((n!) * (m!)) \quad \textbf{(39)}$$