BRLMM: an Improved Genotype Calling Method for the GeneChip® Human Mapping 500K Array Set

Introduction

Highly accurate and reliable genotype calling is an essential component of any high-throughput SNP genotyping technology. The Dynamic Model (DM, [1]) which has been extensively used for the GeneChip® Human Mapping 100K Array Set and the GeneChip® Human Mapping 500K Array Set has proven to be very effective, however it is possible to do better. Rabbee & Speed recently developed a model called the Robust Linear Model with Mahalanobis distance classifier (RLMM, pronounced 'realm') which provided an improvement over DM on the Mapping 100K set [2,3,4]. We present here an extension of the RLMM model developed for the Mapping 500K product which provides a significant improvement over DM in two important areas – it improves overall performance (call rates and accuracy) and it equalizes the performance on homozygous and heterozygous genotypes. The difference between RLMM and this approach is the addition of a Bayesian step which provides improved estimates of cluster centers and variances, the new model is called BRLMM (pronounced 'B-realm').

The performance improvement is achieved by two main advances over the DM model. Firstly, RLMM (and hence BRLMM) performs a multiple chip analysis, enabling the simultaneous estimation of probe effects and allele signals for each SNP. Just as it has in the now reasonably mature field of probe-level expression analysis, accounting for probe specific effects results in lower variance on allele signal estimates. The second main source of improvement is the estimation of genotypes by a multiple-sample classification, borrowing information as necessary from other SNPs to better predict the properties of the underlying clusters corresponding to the {AA,AB,BB} genotypes. By contrast, the DM approach calls genotypes by analyzing the probe-level intensities one SNP and one chip at a time, using strong assumptions about what the underlying probe intensity patterns should look like in the context of each of the genotypes. RLMM and BRLMM make weaker assumptions about the behavior of probe intensities than does DM, making them far more robust in the presence of real-world data.

Figure 1 presents an overview of the BRLMM approach. The first step is to normalize the probe intensities and estimate allele signal estimates for each SNP in each experiment. The allele signal estimates are then transformed to a 2-dimensional space in which the underlying genotype clusters are 'well behaved' in terms of having similar variance for each of the clusters. In parallel we derive an initial guess for each SNP's genotype using the DM approach (with confidence threshold set to 0.17 for high stringency). We then look across SNPs to identify cases where there are at least a certain minimum number of examples of each of the 3 genotypes according to the initial guesses. This subset of SNPs is used to estimate a prior distribution on the typical cluster centers and variance-covariance matrices. Each SNP is then visited in turn and the cluster centers and variances implied by the initial genotype guesses are combined with the prior information in an ad-hoc Bayesian procedure to derive a posterior estimate of cluster

centers and variances (it is principally this step that distinguishes BRLMM from RLMM). Finally, a genotype and confidence score is assigned for each observation according to its Mahalanobis distance from the three cluster centers.

The remainder of this manuscript steps through each of the above steps in detail and then presents a detailed assessment comparing various aspects of BRLMM and DM performance.

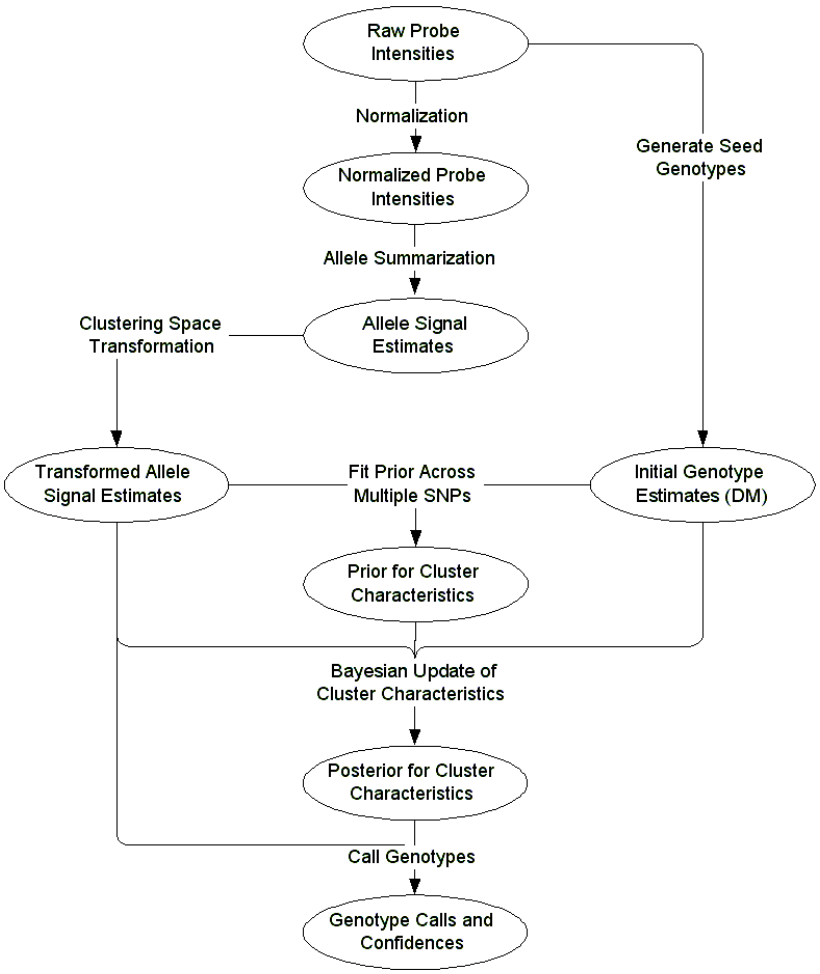

Figure 1: BRLMM algorithm workflow

Normalization and Allele Summarization

The normalization and allele summarization steps of the BRLMM algorithm consist of producing a summary value for each allele of a SNP in each experiment. The “A” allele summary value increases and decreases with the quantity of the “A” allele in the target genome, and similarly the “B” allele summary value increases and decreases with the quantity of the “B” allele in the target genome. These summary values are calculated to

remove extraneous effects – chip-chip variation, background, and the relative brightness of different probes on the array. This section explains the technical details of this summarization process, which is similar to that used on expression arrays.

For each SNP of interest, the array contains multiple probes designed to hybridize to each allele of the SNP. The intensities of these features typically vary together in systematic ways for each genotype of the SNP. We therefore summarize these intensities in a single value for the features corresponding to each allele, the "signal" for that allele. (Note: due to cross-hybridization with the alternate allele, this signal does not directly correspond to the concentration of the perfectly matched allele.) The intensities of the probes matched to the "A" allele are expected to decrease with decreasing quantities of the "A" allele, and similarly for the "B" allele probes. Since these change in opposite directions, we summarize the probes for each allele as independent signals. Therefore, for each SNP in each experiment, we obtain two values – an "A" signal and a "B" signal, which summarize the probes.

From the field of expression analysis on arrays, we know how to summarize several probes to a single signal value effectively. We need to account for extraneous effects on the probe intensity that vary from experiment to experiment (normalization), account for potential differences in background from chip to chip (background adjustment), and account for the systematic differences in feature intensity due to probe composition (feature effects). While there are many options available for each of these effects, we have chosen to use off-the-shelf options: quantile normalization at the feature level, no background adjustment, a log-scale transformation for the perfect match intensities, and a median polish to fit feature effects to the data obtaining a signal. This is exactly the same methodology that can be applied to summarize an expression array and produce a signal for a probe-set.

Quantile normalization is performed as in the literature – the intensities on each chip are ranked, and then the average intensity across experiments for each rank of intensity is substituted within each experiment for the given rank. [If R(I) is the rank of intensity within a chip, and Q(R) is the average intensity for a given rank, the quantile normalized intensity within a chip is Q(R(I))]. Because the quantile function is slowly varying and smooth, we approximate the Q(R) function for each chip with a linear interpolation for processing speed ["sketch" normalization]. This allows us to normalize millions of data points per chip rapidly with compact summaries of the data.

Several background adjustments were explored during development, and we settled on using no adjustment for background. Unlike expression arrays, the target concentrations are well above background for the majority of the fragments containing SNPs. For this assay, background adjustment was not useful for downstream genotyping, and therefore the (normalized) perfect match intensities are used without adjustment for background.

To account for systematic differences in relative brightness between features, we fit the standard log-scale additive model to the probes for each allele separately: $\log(I_{i,j}) = f_i + t_j$

$+\varepsilon_{i,j}$, where f_i is the effect due to feature i across experiments, t_j is the effect with experiment j responding to the genotype of the SNP and the relative quantity of the fragment on which it is located (because of cross-hybridization to the other allele it cannot be interpreted as simply the effect due to the concentration of target for allele A), and $\varepsilon_{i,j}$ is the multiplicative error for the observation. We fit this model using the standard median polish procedure for f and t, and for each experiment output the fitted value for t as the signal for that allele. For identifiability, we require sum(f) = 0. The output signal value is retransformed to lie on the original linear intensity scale: signal = exp(t).

These stages constitute the normalization and allele summarization portion of the algorithm. At the end of these steps, we have for each SNP in each experiment two signal values: one for the "A" allele probe set, and one for the "B" allele probe set. Each SNP therefore has a 2xN matrix of values output – 2 signals for each of N experiments. This output matrix is then used to evaluate each SNP for the genotype present in each experiment.

Clustering Space Transformation

Now that we have signals for the two alleles of the SNP across all experiments, we will be evaluating distances between a prototype (cluster center) for a given genotype (AA, AB, BB) and the actual data seen in any one experiment. However, raw "signal" value, while very useful for expression analysis, is not perfectly suited for genotype cluster analysis (figure 2a). We transform each pair of signals for each experiment into a space with properties more suitable for evaluating genotypes.

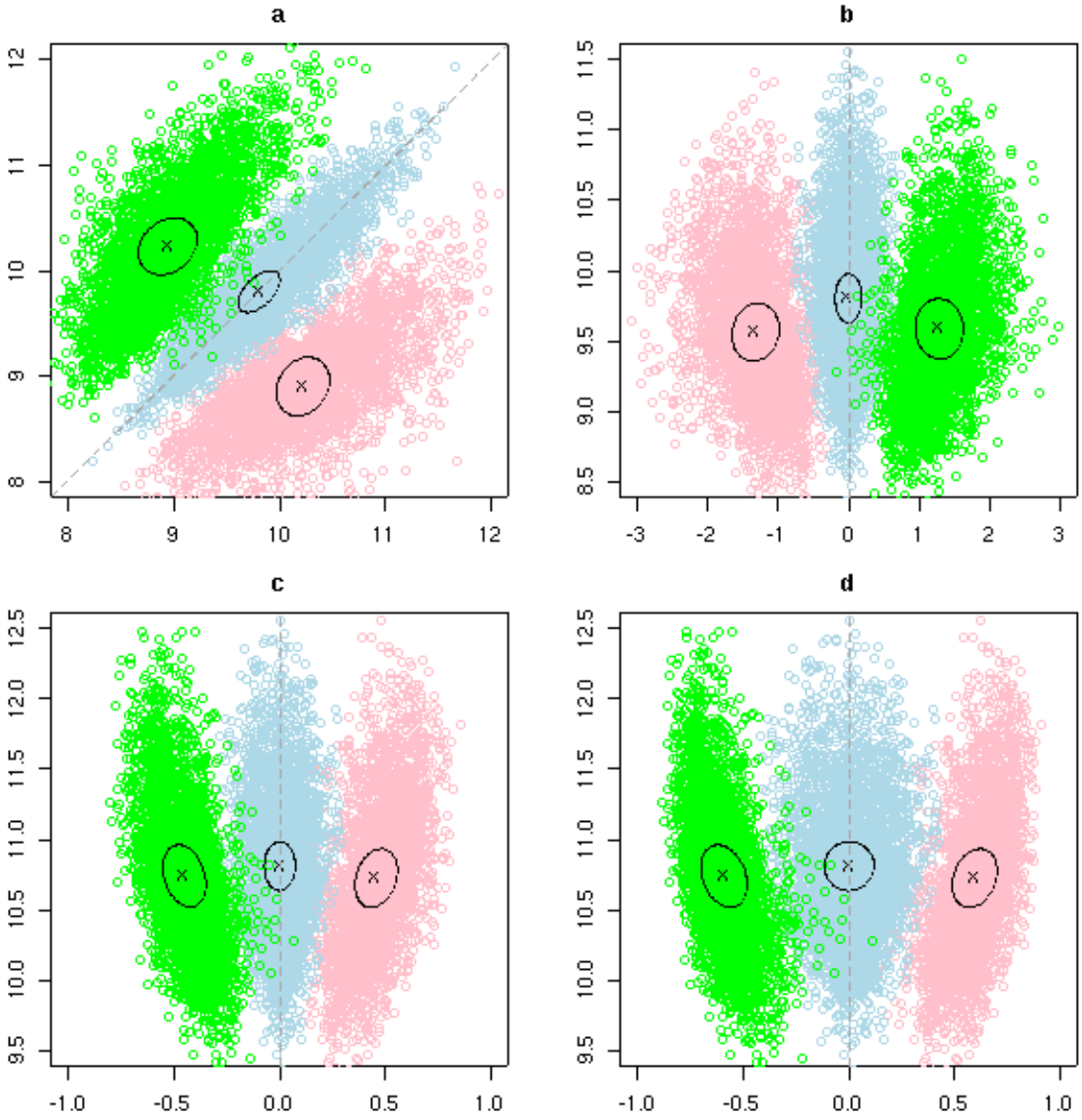

Figure 2: Clustering Space Transformations. For each of a variety of clustering space transformations the cluster centers (as estimated using DM with a stringent 0.17 threshold) are determined. Each plotted point corresponds to an estimate of the cluster center for one genotype in one SNP, with color indicating genotype (AA is red, AB is blue, BB is green). The black x's denote the grand mean of cluster centers for each genotype, and the black ellipses are derived by taking the average within-cluster variance and covariance – thus the ellipses are to be interpreted as representative of the 'typical' variance of a cluster. Sub-plot (a) plots the untransformed allele signal estimates on log2 scale. Sub-plot (b) is motivated by the MvA or MA plots from expression and plots $(\log_2(S_A)+\log_2(S_B))/2$ on the y-axis against the log2 ratio on the x-axis. Sub-plot (c) plots the signal strength $\log_2(S_A + S_B)$ against the allele contrast $(S_A - S_B)/ (S_A + S_B)$. Sub-plot (d) is similar to (c) but the contrast is transformed by the Cluster-Center-Stretch (CCS) transformation (see figure 3).

The desirable qualities for such a space include approximate independence of the difference between genotypes and the magnitude of signal, and controlling the variation within the various clusters to be comparable. For example, the standard "MvA" or "MA" transformation used to plot expression analysis could be applied to the two signals, resulting in $M = \log(S_A)-\log(S_B)$ and $A = (\log(S_A)+\log(S_B))/2$. This isolates most of the difference between genotypes into the M axis, leaving a mostly irrelevant "brightness" component in the A axis.

However, this MvA transformation space is sub optimal, because it increases the variation asymmetrically for homozygous and heterozygous genotypes – in the presence of an AA genotype S_B will be near zero and hence highly variable on the log scale, and conversely for S_A in the presence of BB. The result is that the MvA transformation artificially makes the homozygous clusters more broadly variable than the heterozygous cluster (figure 2b). This causes points to be more often miscalled homozygous than heterozygous because the distance to the homozygous cluster tends to be under-estimated, as it is scaled by the observed standard deviation, leading to heterozygote dropout.

We therefore wish to use a space in which the spread of homozygous clusters can be controlled, even when a signal estimate is near zero, and where the typical variation can be adjusted to be similar between heterozygous and homozygous genotype clusters. Let us define two axes: Contrast = $(S_A-S_B)/(S_A+S_B)$ and Strength = $\log(S_A+S_B)$. Strength of course measures the overall brightness, which is mostly independent of genotype, and Contrast is a quantity that will depend most strongly on genotype ranging from -1 for the ideal BB genotype to +1 for the ideal AA genotype. As seen in figure 2c this transformation still has the property that the homozygous clusters tend to display more variability than the heterozygous, and so we further generalize the Contrast axis to define a Transformed Contrast = $\text{asinh}(K(S_A-S_B)/(S_A+S_B))/\text{asinh}(K)$, where K is a tuning constant. Figure 3 shows the functional form of this transformation for different values of K. The effect of varying K is to change the amount of "stretch" of the difference between A and B signals when the difference is small (i.e. likely to be heterozygous), vs. the difference between A and B signals when the difference is large (i.e. likely to be homozygous), thus K can be used to balance the variability in homozygous and heterozygous genotypes and remove any heterozygous dropout. By experimentation across several data sets, we ascertained that the value K=4 worked well to balance the variation of genotype clusters (figure 2d).

While many other transformations of the data could be used, this space worked well for clustering genotypes while avoiding heterozygous dropout. We therefore implemented this as "Contrast Center Stretch" (CCS) option within the software, and cluster in this transformed signal space.

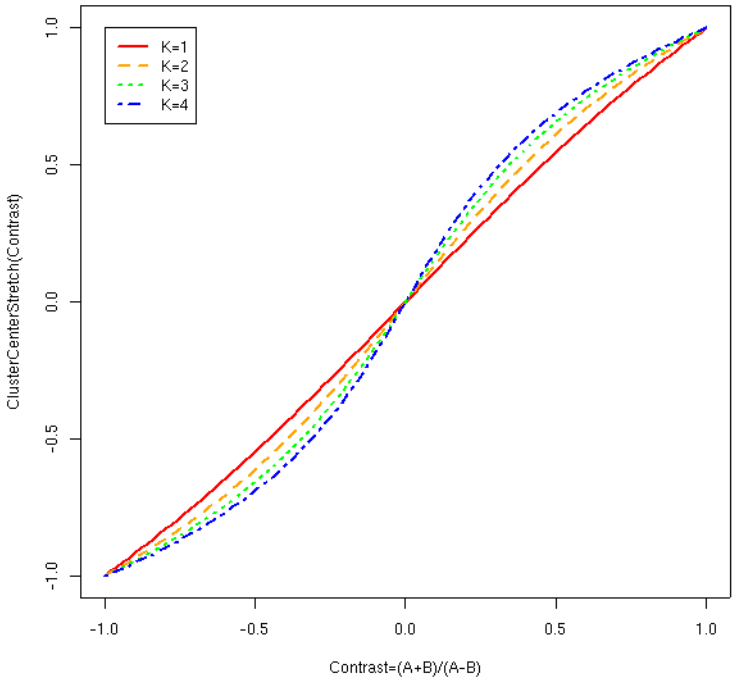

Figure 3: Examples of the Cluster Center Stretch (CCS) transformation. The CCS transformation is defined as asinh(K*Contrast)/asinh(K) where Contrast is defined as $(S_A-S_B)/(S_A+S_B)$. The effect of the transformation is to stretch contrast values near zero (corresponding to heterozygous genotypes) and to compress contrast values near -1 and +1 (corresponding to homozygous genotypes). Higher values of K apply a more extreme transformation, setting K to 1 yields effectively an identity transformation. The value of K can thus be tuned to alter the balance between performance on homozygotes and heterozygotes, with higher K values making het calls more likely.

Calling Genotypes

We call genotypes by a template-matching procedure comparing the transformed allele signal values observed in an experiment to the typical values (prototype) we expect for each genotype. The genotype that is closest in typical value is the one that is assigned (a minimum distance classifier). The approximate confidence we have in that call is based on the ratio of the nearest prototype to the second nearest prototype. This allows us to rank the genotype assignments by quality, and hence make the decision not to call in cases of ambiguity.

Every SNP is expected to have three genotypes, “AA”, “AB”, and “BB”. For each genotype for a given SNP, we expect to have a prototype (typical observed values for that genotype, or cluster center), with some scatter of values around the prototype. We approximate the scatter by a multidimensional normal distribution (and the careful choice of the CCS transformation ensures this is a good approximation). For clusters of this type, the standard method of evaluating the distance from the cluster center (prototype) to

a test point is to use the Mahalanobis distance. The Mahalanobis distance takes into account the variation (and covariation) in the cluster along each axis, and is defined by $\text{sqrt}[(x-\mu)^t \Sigma^{-1}(x-\mu)]$ where μ is the cluster center, x is the test value, and Σ is the variance-covariance matrix describing the multidimensional normal of the cluster.

So, within any experiment, we derive transformed values x for a SNP and compare to the three cluster centers μ_{AA}, μ_{AB}, and μ_{BB} with covariance matrices Σ_{AA}, Σ_{AB} and Σ_{BB}, obtaining distances d_{AA}, d_{AB} and d_{BB}. We call the genotype of the SNP as the genotype with the smallest such distance. In our clustering space, each prototype consists of two components – a center and a variance. The center component consists of a mean Contrast and Strength for the cluster, $\mu_G = (\text{Contrast}_G, \text{Strength}_G)$ where G denotes the genotype. The variance component is a 2x2 variance-covariance matrix $\Sigma_G = (\sigma_{1,1}, \sigma_{1,2}, \sigma_{2,1}, \sigma_{2,2})$, and is symmetric with $\sigma_{1,2} = \sigma_{2,1}$. The distance d_G is computed as defined above.

The confidence we assign to this call is d_1/d_2, where d_1 is the smallest distance of the three and d_2 is the second-smallest distance. This confidence is always between zero and 1. It is a rough measure of the quality of the call (but is not a "p-value"). We set a threshold for quality of 0.5 for a call/no-call decision, based on the performance on several test data sets. This can be adjusted by the user to tune the tradeoff between call rate and accuracy – see the results section for a comparison of performance at various thresholds.

The next section describes how we learn the prototypes and their variation for each SNP from the data.

Estimating Cluster Centers and Variances

The above section dealt with how to call genotypes and ascribe confidence values to those calls given an appropriate prototype. This section deals with how to derive these prototypes.

This is achieved in an ad-hoc Bayesian procedure, where we start by deriving a generic prior describing genotype clusters and centers for the 'typical' SNP, and then visit each SNP in turn, combining the generic SNP prior with initial genotype estimates for the specific SNP to derive a posterior estimate of cluster centers and variances. This posterior estimate is what is then used in the manner described in the previous section to call genotypes. Figure 4 provides a couple of examples of SNPs to which this procedure has been applied.

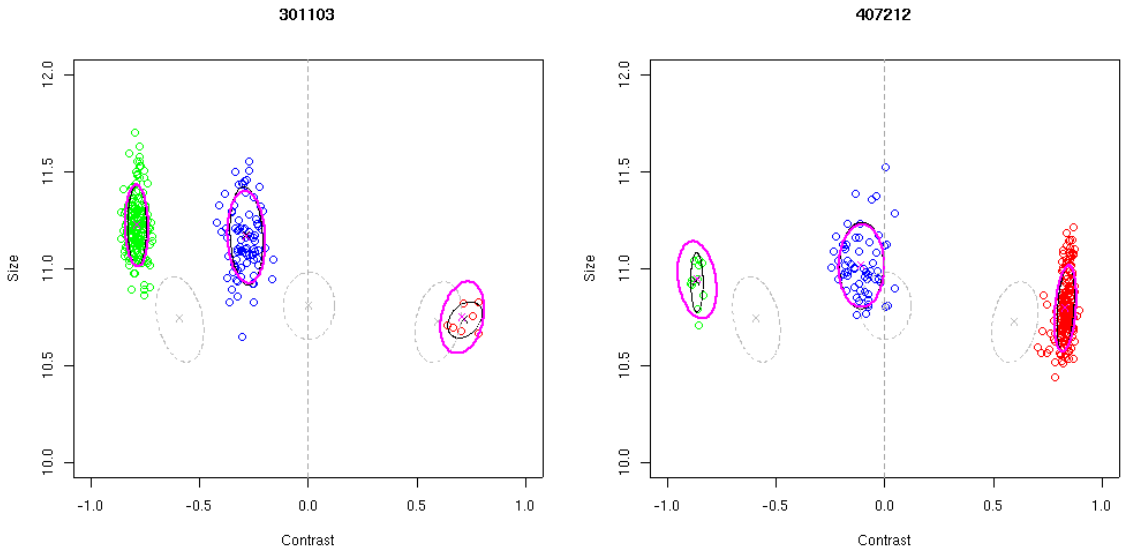

Figure 4: Examples of SNP clustering in action. The samples and SNPs here are part of HapMap and so we have an independent estimate of the genotype for each case, indicated by the color of each point (red=AA, blue=AB, green=BB). In each plot the prior estimates of cluster centers and variance are indicated in the light gray x's and ellipses. Note that because we are using here a generic prior for all SNPs the prior is the same for both examples. The cluster centers and variances from the observed data as estimated by using DM with a stringent 0.17 threshold for seeding are indicated by the black x's and the black ellipses. The posterior estimates of cluster centers and variances are indicated by the magenta x's and ellipses. In all cases the ellipses extend to a Mahalanobis distance of 2 from the cluster center. One thing to note is that in both SNPs the heterozygote and major homozygote clusters the posterior estimate is essentially the same as the estimate provided by the data, which is what we would want – there is sufficient data available for these clusters that the prior is overridden. For the minor homozygotes the posterior estimate is more strongly influenced by the prior as there are few data available for this rare genotype. A second thing to note is that this clustering approach is comfortably handling the phenomenon of unequal allele signals for A and B alleles – particularly in the case of the SNP on the left which is quite shifted from the ideal heterozygotes contrast value of zero.

To start the process we need to seed with some initial genotype estimates from which to build the generic prior. We have an excellent candidate in the existing DM approach which we use with a highly-stringent confidence threshold of 0.17 to determine initial genotype calls. Note that in this use of DM calls for a starting point there is still an indirect reliance on the MM probes, however we have demonstrated that it is possible to get sufficiently good initial estimates without requiring MM probes so it is feasible to make new chip designs with at least half the number of probes. With these initial calls in hand we then take a random sample of 10,000 SNPs and scan through them to identify SNPs which each have at least 2 initial DM calls (the minimum requirement to have a variance estimate for each genotype). Note this places a requirement of an absolute minimum of 6 samples to run together, though in practice it is generally better to have more (discussed in more detail below). The use of a random sample of SNPs allows for faster and more memory efficient processing – only a small subset of the probe intensities needs to be loaded and analyzed. The random sampling is formally a simple random sample from all SNPs on the chip and is implemented in a deterministic fashion so that

re-analyzing the same data at a different time or on a different operating system will yield the same results. This result of this step is typically ~5,000 SNPs (depending on sample size and genetic diversity) which are then used to derive the generic SNP prior.

What we loosely term the generic prior consists of a 4 components:

- m The 6-dimensional vector of the average cluster center coordinates across SNPs (6 free parameters)
- M The 6x6 variance-covariance matrix of cluster center coordinates across SNPs with entries $m_{i,j}$ (21 free parameters)
- S The block-diagonal 6-dimensional variance-covariance matrix of within-genotype transformed allele signal estimates with entries $s_{i,j}$ (9 free parameters: 3 genotypes each with a variance term in each of the two transformed allele signal dimensions and one variance-covariance term)
- p The 'effective number of observations' or 'pseudo-count' associated with the variance estimate S. We supply this as a predetermined value (default is 40) rather than estimating it from the data. Results are quite insensitive to the setting of this parameter.

In total, the generic prior consists of 36 parameters estimated from thousands of SNPs, and one parameter p, which is set up-front. This generic prior can be derived on-the-fly within the dataset being analyzed or can be derived up-front from a previous dataset and applied to a new one. We have tried out both ways and found that there is generally little difference between either approach, provided the dataset used to generate the prior is similar in terms of overall call rates to the dataset to which it is applied, however if the datasets have overall very different levels of performance applying the prior generated from one dataset to the other can lead to sub-optimal (though still very good) performance. The recommended default is to estimate the generic prior on-the-fly, though it is quite likely that future work will find improved modes of running which involve a workflow where parameters are estimated in a large training dataset up-front.

Having estimated the generic prior, the next step is to visit each SNP in turn to combine the prior with whatever DM initial estimates may be available for the particular SNP to come up with a posterior estimate for cluster centers and variances. To set up some notation, we have the following SNP-specific quantities:

Observed data for the given SNP

- v The 6-dimensional vector of the cluster center coordinates, estimated as the average transformed intensity value within each genotype. Some or all of these entries may be null if there are no DM initial estimates of one or more of the 3 genotypes.
- W The 6x6 block-diagonal variance-covariance matrix of within-genotpe variances and covariances. The entries of the matrix are $w_{i,j}$ with $w_{i,j} = 0$ for $|i-j| > 1$. Some or all of these entries may be null if there are not at least 2 DM initial estimates of one or more of the 3 genotypes.

N A 6x6 diagonal matrix with entries (n_{AA}, n_{AA}, n_{AB}, n_{AB}, n_{BB}, n_{BB}), the number of DM initial estimates for each of the 3 genotypes.

SNP-specific posterior estimates

μ The 6-dimensional posterior estimate of cluster center coordinates to be used in the prototype for calling genotype estimates.

Σ The 6x6 block-diagonal estimate of within-genotype variances of transformed allele signal estimates. The entries of the matrix are $\sigma_{i,j}$ with $\sigma_{i,j} = 0$ for $|i-j| > 1$.

Having set up this notation, the posterior estimates are derived as a two-step process. Firstly we obtain a posterior estimate of the 9 non-zero parameters in S by doing point-wise shrinkage towards the prior estimate, using an effective number of observations p which is chosen up-front:

$$\sigma_{i,j} = \left((n_{i,j}-1)\, w_{i,j} + p\, s_{i,j}\right) / \left((n_{i,j}-1) + p\right)$$

This Bayesian update has the intuitively sensible property that when there is little or no data available for a genotype within a SNP ($n_{i,j}$ small) the variance-covariance matrix to be used for the genotype will be predominantly driven by the typical variance-covariance that we see across most SNPs. Conversely, if there is abundant information for a particular genotype of a SNP ($n_{i,j}$ large) there will be little reliance on the prior and the estimate will be primarily based on data specific to the SNP. The point of transition between the reliance on prior and observed is tuned by the number of pseudo-observations, p. We have found that overall performance is insensitive to the setting of p, though it is possible that it may have a larger effect on certain genotypes, such as rare genotypes. The recommended default value for p is 40.

The final step is to come up with a posterior estimate of the cluster centers, μ. We make the assumption that cluster variances are independent of the centers. The update rule is

$$\mu = \left(M^{-1} + (NS)^{-1}\right)^{-1} \left(M^{-1}m + (NS)^{-1}v\right)$$

Again, this has the intuitively sensible property that when there is little or no labeled data available the estimate of cluster centers will be driven mainly by the prior estimate m, and when there is a lot of data available for a given genotype the estimate will be driven by v. Loosely speaking, this update rule has the form of a weighted average of the prior and observed data, with the prior having weight inversely proportional to M, the variance-covariance matrix of cluster centers in the SNPs used to build the prior, and the observed data for the particular SNP having weight inversely proportional to NS, a product of the number of observations and the variance in the observed data.

With these posterior estimates of center and spread for each cluster, genotypes and confidences are then determined as outlined in the previous section.

Special Cases

The preceding algorithm assumes that the observations for each SNP are well described by prototypes for each genotype. However, for SNPs on the X chromosome, there are distinct clusters for each gender due to males having one fewer copy of the X chromosome. This not only changes the location of the cluster centers for XY individuals, but the SNPs located on chrX may end up being called as heterozygote. We therefore treat the chrX SNPs differently for XX individuals than for XY individuals. Note that the special treatment of chrX SNPs described here is only applied to SNPs on chrX in the nono-pseudo-autosomal region, and for the rest of this section when we talk about chrX it is to be interpreted as chrX excluding the pseudo-autosomal region

We detect the difference between XY and XX individuals by the seed calls from DM. XY individuals are estimated as those having heterozygosity less than 7.5% on chrX. The remaining individuals are classified as XX. For each chrX SNP, we treat XX individuals and XY individuals as separate data sets.

XX individuals are handled using the standard BRLMM methodology for all chrX SNPs, that is, three cluster centers are learned from the data along with covariance matrices and used to classify observations. However, no data from XY individuals is used in this calculation.

XY individuals are handled using a modification of the BRLMM methodology for all chrX SNPs. Only two cluster centers can be learned from the data (AA and BB), and only the data for the XY individuals are used. Therefore the following modifications are performed. First, only DM homozygous calls are used to seed the learning procedure that estimates cluster centers. This provides approximate locations for the homozygous prototypes for the SNP-specific clustering.

Second, we modify the heterozygous cluster for the generic prior to remove it from the range of typical data (this is to avoid having special-purpose code for two or three prototypes). This surgery removes any reasonable possibility of making a heterozygous call in an XY individual. The modification moves the heterozygous prototype to (Contrast=0, Size = -Infinity), and modifies the heterozygous prototype covariance matrix to be (0.01, 0, 0, 0.01). The covariance with the other cluster prototypes in the prior is set to be zero. This removes any influence the heterozygous cluster has on the homozygous clusters, and vice versa. Thus, for XY individuals, only “AA” and “BB” genotypes are fit, and for any real observed data, “AB” will never be called.

Fitting of XX and XY individuals separately improves the genotyping performance within each group. Modifying the prior for XY individuals to avoid heterozygous calls improves the genotyping performance for XY individuals. This is the justification for having a special purpose modification for chrX SNPs within BRLMM.

Results

The ideal way to assess performance would be to evaluate the tradeoff between accuracy and call rate in data generated from a collection of samples for which the true reference genotypes are available for all SNPs on the Mapping 500K set. Fortunately something closely approximating this has been made possible by the International HapMap Consortium – the phase 2 release provides reference calls on a collection of 270 samples for approximately 70% of the SNPs on the Mapping 500K set. This constitutes an excellent resource for the performance evaluation; though it is important to bear in mind the caveat that the genotype calls in HapMap themselves do have some small but non-zero error rate. Additionally, the HapMap samples consist of some trios, enabling the evaluation of Mendelian inheritance error rates. Finally, we also look at reproducibility of genotype calls on sample replicates.

For evaluation of call rates, accuracy and Mendelian inheritance error rate we use a collection of HapMap samples generated by a customer of the Mapping 500K product. This dataset consists of 66 HapMap samples – 33 CEPH Caucasian samples and 33 African (Yoruban) samples. All 66 of the samples meet the Mapping 500K product specification of 93% call rate with DM at a confidence threshold of 0.33. We use as the gold standard calls from HapMap release 20 after excluding any calls submitted to HapMap by Affymetrix (to reduce risk of positive bias in performance estimates). We also swapped the A↔B naming convention for 70 HapMap SNPs for which the allele names were clearly swapped (SNPs for which the per-SNP accuracy jumped from below 10% to above 90% when alleles were swapped). To account for the fact that one can adjust the confidence threshold to trade off between call rate and accuracy we look at performance at all possible thresholds and plot the relationship between HapMap concordance and no-call rate, as shown in Figure 5. The figure demonstrates the significant improvement in both call rate and accuracy comparing BRLMM with DM. Moreover, looking at the performance curves broken out by the type of the reference call (homozygous or heterozygous) we also see that BRLMM makes a large reduction in the performance differential between the two classes. Table 1 presents performance for DM and BRLMM at various thresholds combining performance across the Nsp and Sty chips. We have set the default BRLMM threshold to 0.5, it can be tuned for higher call rates or higher accuracy according to what will better suit the requirements of downstream analysis.

The performance improvement seen on the dataset as a whole is also seen consistently across all samples, as can be seen in figure 6.

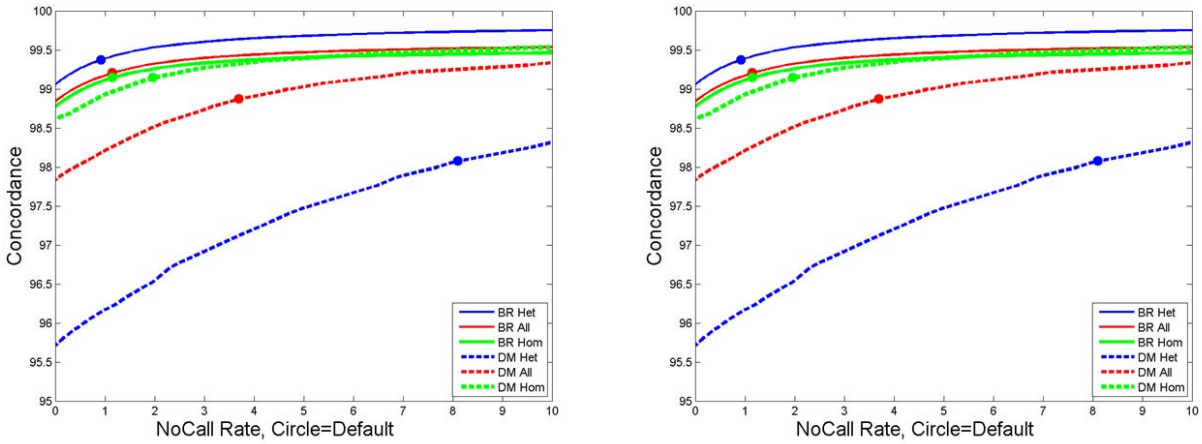

Figure 5: Performance of BRLMM and DM on HapMap samples for Nsp (left) and Sty (right) chips. Concordance with HapMap is assessed for all possible confidence thresholds for each genotype calling method and the resulting concordance with HapMap reference calls is plotted against the no-call rate. The ideal method and data would consist of a curve that reaches the top-left of each plot, corresponding to 100% accuracy at 100% call rate. Results for BRLMM and DM are presented in solid and dashed lines respectively. The dot on each curve indicates the performance at the default confidence threshold (0.5 for BRLMM, 0.33 for DM). There are two main points to make about the results – firstly, BRLMM provides a significant improvement in overall call rate and overall accuracy compared to DM (red curves). Secondly, BRLMM has markedly more even performance on homozygote (green curve) and heterozygote (blue curve) genotypes than DM, which has notably lower performance on heterozygotes..

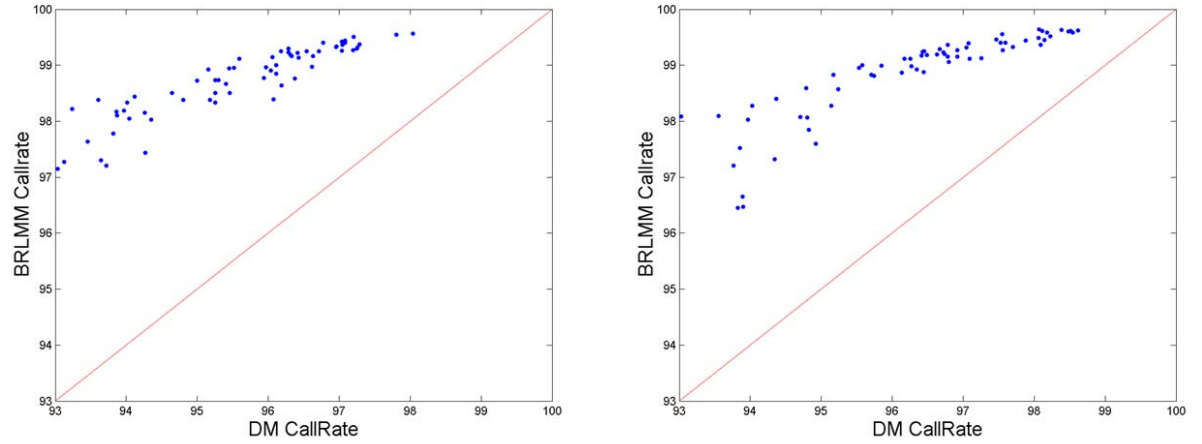

Figure 6: Comparison of per-sample call rates between BRLMM (at 0.5) and DM (at 0.33). Results on the left are for Nsp and on the right are for Sty. BRLMM improves call rates in all cases. In absolute terms the performance improvement is larger for lower call rate samples though when viewed as a fold-reduction in no-call rate we see a fairly consistent 2/3 reduction in no-call rate relative to DM at 0.33. The call rates are very correlated whichever method is used, which indicates that while BRLMM improves performance overall there is an intrinsic ranking to the samples whereby the results for some samples are better than others regardless of how they are analyzed.

Performance with respect to HapMap reference samples has been evaluated in this manner on other datasets of varying degrees of quality and we find that the call rate improvement can be characterized as a consistent 2/3 reduction in no-call rate going from DM at a threshold of 0.33 to BRLMM at a threshold of 0.5. At the same time we consistently find about a 0.1% increase in HapMap concordance going from DM at its more stringent threshold of 0.26 to BRLMM at a threshold of 0.5.

Method	Confidence Threshold	Overall Call Rate	Hom Call Rate	Het Call Rate	Overall Concordance	Hom Concordance	Het Concordance
DM	0.26	**94.16%**	97.24%	86.32%	**99.15%**	99.39%	98.38%
DM	0.33	**95.96%**	98.24%	90.16%	**98.94%**	99.27%	97.93%
BRLMM	0.3	**97.40%**	97.40%	97.75%	**99.40%**	99.34%	99.55%
BRLMM	0.4	**98.27%**	98.30%	98.48%	**99.31%**	99.25%	99.47%
BRLMM	0.5	**98.79%**	98.82%	98.93%	**99.26%**	99.20%	99.40%
BRLMM	0.6	**99.15%**	99.18%	99.25%	**99.17%**	99.11%	99.33%

Table 1: Performance on HapMap dataset for DM and BRLMM at various fixed thresholds. Results are based on combining the Nsp and Sty chips. At the default confidence of 0.5, BRLMM achieves approximately a 2/3 reduction in no-call rate over DM at a threshold at 0.33, a finding that has been consistently observed on a variety of datasets. At the same time it achieves an improvement of about 0.1 % in HapMap concordance compared to DM at a more stringent threshold of 0.26, or about 0.2% compared with DM with threshold at 0.33. The other big improvement of BRLMM is the large reduction in call rate and accuracy differences between homozygotes and heterozygotes. In the evaluation of concordance on homozygotes we found on the order of 25 SNP which exhibited what appeared to be 'allele swap' where the concordance with HapMap genotypes was extremely low (on the order of 20% or less) but which jumped to very high (90% or more) when the A and B alleles were swapped – we attribute such SNPs to errors in HapMap and swap the alleles before reporting final accuracy.

One caveat about evaluating concordance with HapMap is that to some extent it provides only a lower bound estimate for accuracy, since HapMap itself does have a certain error rate. With this in mind, it is useful to look at alternative measures of performance. The dataset used here contains (father,mother,child) trios of samples which can be assessed for Mendelian consistency. The Mendelian accuracy is estimated looking only at informative trios (those in which we have a call for all three samples where the parents are not both called heterozygous), call this number T. If the number of such trios which exhibit a Mendelian inconsistency is E then the Mendelian accuracy is estimated as (T-E)/3T, which is based on the assumption that when there is an inconsistency in a trio that it is only one of the three calls which is in error the vast majority of the time. Measured in this manner we find that the Mendelian accuracy using DM at a confidence threshold of 0.33 is 99.83%, increasing to 99.93% using BRLMM at a confidence threshold of 0.5 (approximately a 2-fold reduction in Mendelian error rate).

The final metric of performance we evaluate is reproducibility on sample replicates. Arguably this metric is less useful than those above since it only reports on the consistency of calls made but not on whether or not those calls are actually correct. Nevertheless, other things being equal a reproducible method will generally be preferable to one that isn't. We evaluate reproducibility on a different set of HapMap samples which includes 7 samples each replicated twice. Within this dataset we find that the reproducibility of DM calls at a threshold of 0.33 is 99.77%, which improves to 99.85% when calls are made with BRLMM at the default threshold of 0.5.

Discussion

BRLMM provides a significant improvement over DM method, raising call rates, accuracy and just as importantly, establishing balanced performance between homozygotes and heterozygotes. As a multiple-chip method it has some extra considerations which need to be taken into account in practice.

One matter to consider is the batch size in which to apply BRLMM. While more samples will generally lead to better performance, we have found that for good datasets performance reaches a plateau with as few as 50 samples, whereas for lower-quality datasets it can take as many as 100. The working definition of 'good' used here is a dataset with an average call rate of 95% using DM at a threshold of 0.33, and for 'lower-quality' we mean a dataset with an average call rate of 93% using DM at a threshold of 0.33 (of course these call rates increase when called with BRLMM). When we talk about numbers of samples we really mean number of distinct DNAs analyzed – a dataset consisting of many replicates of the same sample may not have sufficient genetic diversity to build the prior. Note that with the default settings BRLMM requires at least two observations of each genotype to build the prior, so the absolute minimum number of samples required is 6, though running with this small a number is not advised.

Another consideration is the extent to which datasets can be combined. On the one hand this should help in terms of increasing the number of observations, particularly for rare genotypes, thereby improving the performance on rarer genotypes. On the other hand, the validity of combination of datasets will depend on the degree to which the combined datasets have the same underlying probe intensity distribution, probe effects, cluster centers and cluster variances. We have found that combination of datasets from different labs can change performance slightly in either direction, and understanding the criteria under which it will and won't succeed remains an area of future work.

The main differences between BRLMM and the previously-developed RLMM [2,3,4] method lie in the clustering space transformation and in the estimation of cluster centers and variances. There are a number of other potential improvements which have been or are in the process of being evaluated. Though some of them are enabled in the software implementing the BRLMM method these features are not part of the default recommended workflow as they have not been as thoroughly tested, but it is conceivable

that these approaches or some modification of them may lead to further improvement in future. These are:

Robustness in estimation of cluster properties: The presence of an outlier value can have a large effect on the center and variance estimated for a given genotype cluster and it is possible that introduction of robust estimates of center and spread may improve performance in such circumstances, however we have so far found that robust estimates have little effect on overall performance and can have a large effect on the balance of performance between homozygotes and heterozygotes.

SNP fragment normalization: It has been shown for the Mapping 100K product that there can be notable biases in fragment amplification in the WGSA assay and that they can be successfully normalized, leading to a large reduction in noise [5,6]. We have found that similar approaches can benefit in the evolution of the WGSA assay used for the Mapping 500K product and it is possible that this will tighten the clusters for some datasets.

Alternative metrics of quality: We have found that a genotype call threshold based on a method suggested in [7] can lead to a slightly better tradeoff between call rate and accuracy – the idea is that instead of using the ratio of smallest to second-smallest Mahalanobis distances one uses $(\text{Mahalanobis distance})^2 + \log(|\Sigma|) + \log(\text{prior probability of being in cluster})$. Given more testing this metric may turn out to do even better.

Analysis of a single sample at a time: It would be a great practical convenience to be able to attain the same performance improvement running only a single sample at a time. Single-sample analysis is typically logistically and computationally more convenient, especially when used in a high-throughput environment. This is possible if one can safely make assumptions about the applicability of previously-computed quantities to each newly-generated chip: the probe intensity distribution to which intensities should be normalized, the estimated probe effects, and the SNP-specific cluster centers and variances. We have found that these assumptions do hold approximately and some of this functionality is enabled in the software implementing BRLMM but the area needs more work.

Treatment of strand information: The current approach ignores the available information about the strand from which each probe is selected, relying on the assumption that beyond the usual probe-specific effects there is no overall difference between the two strands. The majority of the time this appears to be a reasonable assumption but there are some cases where it is clear that the probes for one of the strands are providing limited or even conflicting information. It is likely that an approach that allows for different treatment of the two strands may extract a little more performance out of such SNPs.

Adaptive homozygous/heterozygous balancing. We have chosen a default value of K, the tuning parameter in the allele signal transformation, which achieves an optimal hom/het performance balance in various datasets used for training but have found that in test datasets although it always yields a large reduction in the hom/het discrepancy as compared to DM, it isn't always optimal. An area for future work is to adaptively estimate K or to use some suitably parameterized alternative transformation to further minimize any hom/het discrepancy.

Finally, though the BRLMM method on the Mapping 500K set provides a very significant performance improvement, the existing DM method is still an important part of the workflow. BRLMM can only be run in multiple-chip mode (at least for now), and in a typical high-throughput environment one needs the instant performance metric provided by the DM call rate that can be applied to each chip in turn to decide in real-time if a sample needs to be re-hybridized or re-done.

References

1) Xiaojun Di, Hajime Matsuzaki , Teresa A. Webster , Earl Hubbell , Guoying Liu , Shoulian Dong , Dan Bartell , Jing Huang , Richard Chiles , Geoffrey Yang , Mei-Mei Shen , David Kulp , Giulia C. Kennedy , Rui Mei , Keith W. Jones and Simon Cawley. *"Dynamic model based algorithms for screening and genotyping over 100K SNPs on oligonucleotide microarrays"*. Bioinformatics 2005 21(9):1958-1963
2) Nusrat Rabbee and Terence P. Speed, *"A genotype calling algorithm for Affymetrix SNP arrays"* UC Berkeley Statistics Online Tech Reports, August 2005. http://www.stat.berkeley.edu/users/nrabbee/693.pdf
3) Nusrat Rabbee and Terence P. Speed. *"A genotype calling algorithm for Affymetrix SNP arrays"* Bioinformatics Advance Access published online on November 2, 2005 http://bioinformatics.oxfordjournals.org/cgi/content/abstract/bti741v1
4) RLMM: http://www.stat.berkeley.edu/users/nrabbee/RLMM/
5) Yasuhito Nannya, Masashi Sanada, Kumi Nakazaki, Noriko Hosoya, Lili Wang, Akira Hangaishi, Mineo Kurokawa, Shigeru Chiba, Dione K. Bailey, Giulia C. Kennedy and Seishi Ogawa. *"A Robust Algorithm for Copy Number Detection Using High-Density Oligonucleotide Single Nucleotide Polymorphism Genotyping Arrays"* Cancer Research 2005 65:6071-6079
6) Shumpei Ishikawa, Daisuke Komura, Shingo Tsuji, Kunihiro Nishimura, Shogo Yamamoto, Binaya Panda, Jing Huang, Masashi Fukayama, Keith W. Jones, Hiroyuki Aburatani. *"Allelic dosage analysis with genotyping microarrays"* Biochemical and Biophysical Research Communications 2005 333:1309–1314\
7) Richard O. Duda, Peter E. Hart and David G. Stork *"Pattern Classification (2nd ed)"* Wiley Interscience, 2000.