Integrated Data Analysis for Genotyping microarrays

Kai Zhang
Department of Computer Science
New Jersey Institute of Technology
University Heights, Newark, NJ 07102

Marc Q. Ma
Department of Computer Science, and
Center of Applied Mathematics and Statistics
New Jersey Institute of Technology
University Heights, Newark, NJ 07102

Hui-Yun Wang
Bionomics Research & Technology
Environmental and Occupational Health Science Institute
Rutgers, The State University of New Jersey
170 Frelinghuysen Road, Piscataway, NJ 08854

Yu Wang
Department of Computer Science
New Jersey Institute of Technology
University Heights, Newark, NJ 07102

Frank Shih
Department of Computer Science
New Jersey Institute of Technology
University Heights, Newark, NJ 07102

CAMS Report 0405-33, Spring 2005
Center for Applied Mathematics and Statistics

Integrated Data Analysis for Genotyping Microarrays

Kai Zhang [1], Marc Ma[1, 2, *], Hui-Yun Wang[3], Yu Wang[1] and Frank Shih[1]
[1]Dept. of Computer Science, [2]Center of Applied Mathematics and Statistics, New Jersey Institute of Technology; [3]Bionomics Research & Technology, Environmental and Occupational Health Sciences Institute, Rutgers, The State University of New Jersey
Emails: Marc Ma - qma@njit.edu; * Corresponding author

Abstract

We present TIMDA (a Toolkit for Integrated Microarray Data Analysis), a Matlab-based software framework designed for spotted single nucleotide polymorphism (SNP) genotyping microarray data analysis. TIMDA features seamlessly integration of numerical computation, analysis, visualization and algorithms as well as excellent extensibility and maintainability. The framework consists of modules designed for image processing, intermediate data conversion, genotype calling, and loss-of-heterozygosity (LOH) study with text or graphics output. Each of these modules can work smoothly with others or independently from each other. Meanwhile, data from other software also can be integrated into TIMDA.

Keywords: gridding, gene expression, genotype-calling, support vector machines (SVM), artificial neural networks (ANN), pattern recognition.

1. Introduction

Functional genomics is a flourishing science enabled by recent technological breakthroughs in high-throughput instrumentation and data analysis for microarrays [1-5], in which functions and behaviors of thousands of genes in a genome are studied. Genomics study could be very useful in understanding the genetic basis for many biological phenomena. Using microarrays, biologists are able to monitor thousands of genes concurrently, which can be used to deduce gene functions and/or construct complex gene interaction networks.

A large-scale high-throughput genotyping system has been recently developed [3], which first uses one multiplex polymerase chain reaction (PCR) to amplify thousands of short DNA sequences, which contain single nucleotide polymorphisms (SNPs), then uses spotted microarrays to establish gene expression and/or genotypes of the SNP-containing DNA sequences. The amount of data generated from this type of microarray experiment is enormous and it is not feasible to use manual processing for accurate and reliable analysis in daily routine. Without accurate, reliable and fast data analysis yielding critical intermediate results on gene expression, statistics and/or genotypes, it is impossible to infer genetic changes due to drug action, genetic differences during development, gene interaction networks or biological pathways.

We developed TIMDA (a Toolkit for Integrated Microarray Data Analysis, URL: http://web.njit.edu/qma/QMa_software.html) software package for automatic, accurate and reliable data analysis for genotyping microarrays. TIMDA features not only seamlessly integration of numerical computation, analysis, visualization and algorithm, but also excellent extensibility and maintainability.

TIMDA consists of modules for image processing (IM), intermediate data conversion (IDC), genotype-calling (GC), and loss-of-heterozygosity (LOH) study. IM employs several novel techniques in gridding and segmentation, and supports the popular combined TIFF image format, which contains digitized images scanned from two channels, red and green, as shown in Figure 1, which is artificially rendered to show the intensities from two channels. IDC is used to collect useful information such as spot locations, IDs, estimated intensity levels and background intensity levels from text files with different formats and converts them into generalized Matlab's workspaces for subsequent analysis. GC is used to determine the genotypes using different approaches which include machine learning approaches, support vector machines (SVMs) [6] and artificial neural networks (ANNs) [7] with iterative adjustment. Furthermore, users are also allows to plug-in their own genotype-calling methods. LOH is used to establish the pattern of LOH using data from microarrays for normal cells and diseased cells. The OneInt module integrates IP, IDC and GC to help users enhance their efficiencies. In TIMDA, each of these modules can work seamlessly with others or independently from each other. Meanwhile, data from other software also can be integrated into TIMDA.

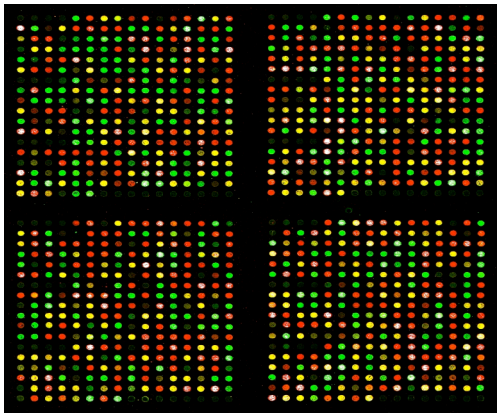


Figure 1. A sample microarray image.

A schematic representation of the modular design of TIMDA is shown in Figure 2(a). Users can select the individual module or several combined modules to analyze their microarray data. Figure 2(b) shows the OneInt module, which allow users to directly determine the genotypes of their microarray experiments using a set of modules selected (GUI, IM, IDC, GC and Output) by default.

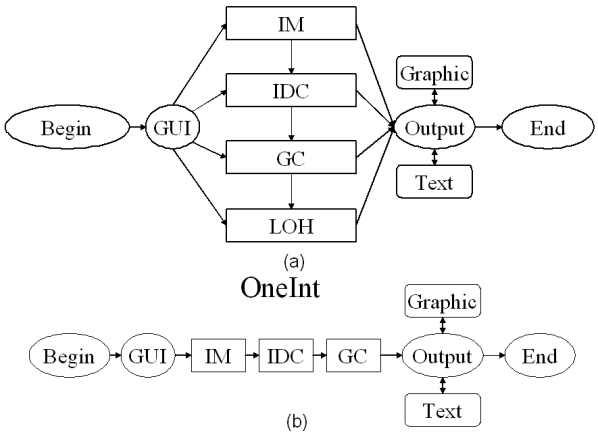


Figure 2. (a) TIMDA framework; (b) OneInt module.

2. Image Processing (IP)

The image processing module performs automatic gridding, segmentation of foreground/background pixels and quantification of the intensities for each spot. IP supports the popular combined TIFF image format, which contains digitized images scanned from two channels, red and green. Image processing has been a bottleneck for reproducibility, accuracy and efficiency of microarray data analysis due to the intrinsic difficulties such as bad grid layouts, contaminations, background estimations, noises, irregular spot shapes, dense layouts and etc. [8]. The design of IP module improves the quality of image processing due to some of its novel features.

2.1 Automatic gridding

The purpose of gridding is to index microarray layouts despite of possible rotations of the scanned images for the microarrays. TIMDA performs automatic image rotation detection and correction as an option when loading images. If an image has no rotation, its intensity projections onto horizontal and vertical axes should be regularly distributed and their variances should be maximized as the peak points. Figure 3 illustrates the variance changes in response to the varying rotation. It is observed that variances of intensity projections to the both axes can be used to indicate the rotation degree. A microarray image usually includes several sub-arrays. After the step of global rotation correction for the whole image, TIMDA applies local rotation correction for each sub-array if this option is selected, which is different from the method in Ref. [16].

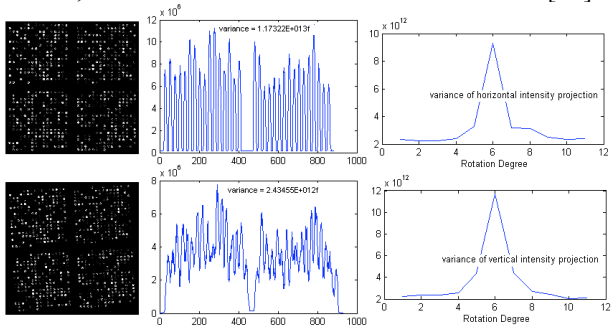


Figure 3. Global rotation detection.

In gridding, a threshold value is manually selected to convert the original image to a binary image, which is then used for constructing spot layout template. We developed a new non-parametric method for this purpose. First, the variances of horizontal and vertical intensity projections are measured as a function of threshold values, such as what is shown in Figure 4 (Right panel). Then we choose the value that maximizes the variance as the threshold for binary image construction. In Figure 4, a "bad" result of gridding is shown when a threshold is chosen arbitrarily (Left panel); a "good" result is shown when a threshold value that generates maximal variance (from Right panel) is used (Middle panel). With optimal threshold values, the converted binary image can show all spots in the most distinguishable way. TIMDA also allows manual threshold selection for both channels using sliding bars while the automatic gridding results are not satisfied. Figure 5 shows the interface of IM module. Based on the binary template, the spot layout is approximately determined using the intensity projections, in which the locations for peaks and gaps are measured. For real images, the actual gridding somehow may not be perfectly aligned, so the fine-tuning is carrying out to complete the gridding procedure. The

fine-tuning is based on the assumption that with more accurate gridding, the spot region covers more intensities than the other regions locally. The center for each spot can be determined by finding the local maximum.

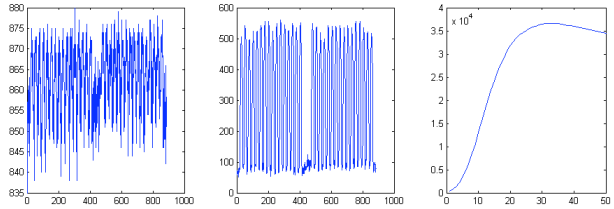


Figure 4. Automatic threshold determination. Left (a): The horizontal projection with a bad threshold. Middle (b): It shows such a projection with the best threshold. Right (c): The distribution of variances of vertical intensity projection based on different thresholds.

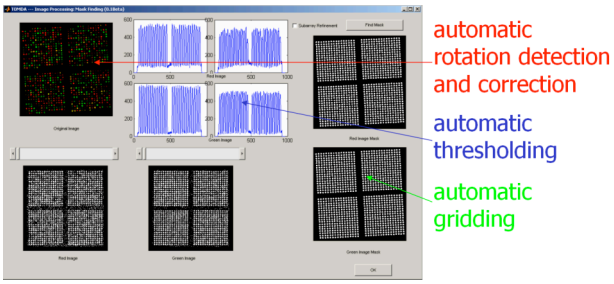


Figure 5. The gridding in the IM module.

2.2 Robust segmentation

After gridding, the segmentation step is carried out to estimate the background and foreground values for each spot. There are several methods for this task. The global background (GB) estimation method calculates the average intensity level of all pixels not belonging to signal regions. Thus, GB ignores the spatial background variation across the whole slide. Several other spatial and histogram-based techniques have been proposed for analyzing microarray images to overcome this limitation. Fixed circle (FC) segmentation fits circles with a constant diameter to all the spots in the image [9] to distinguish the foreground and background. Adaptive circle (AC) segmentation estimates the circle's diameter individually for each spot [10] to improve the performance of FC method, and may generate more reliable estimates. However, AC still does not handle irregular shapes such as donuts properly. Adaptive shape (AS) segmentation offers more flexible answers to irregular shapes, but it cannot give robust estimation for the foreground or background when a big local variation of intensities exists. BlueFuse™ [11, 12], which is one of the histogram-based methods, uses a Bayesian model to generate confidence measures for each spot. In comparison to the spatial-based approaches, histogram-based methods do not analyze spatial distribution for each spot, instead, they directly analyze its histogram distribution, in which the pixels are categorized into foreground and background based on some criteria. However, the quantization in histogram-based methods is unstable when a large target mask is set to compensate for spot size variation [13].

As technology advances, the density of spots on microarrays continues to increase. With fewer pixels used in local background estimation, the variance increases, while the traditional approaches cannot make significant statistical sense and result in unreliable estimates. To overcome this limitation, we focus on a new robust segmentation method based on a larger region to estimate the local background, which is termed as extended local background (ELB) and is essentially a histogram-based method. Before TIMDA processes ELB for each spot, it uses the spot gridding template constructed from the last step to the global background, in which it collects all pixels belonging to the background region over the whole chip, and then eliminate 20% high intensity pixels because of irregular spot shapes, saturated and contaminated spots, and then calculates the global background level $\overline{v}$ and global standard deviation σ_{GB}. The global cutoff threshold value v_{GB} is computed as

$$v_{GB} = \overline{v} + \sigma_{GB},$$

which will be further applied to the local background estimation. Next, TIMDA uses local histogram information on a suitable (larger) population to generate better and more robust estimates of local background noise. Figure 6 shows one allowable ELB configuration.

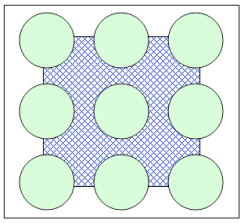


Figure 6. Extended local background.

Pixels in the shaded area with intensity below v_{GB} are initially treated as the candidate background pixels and their mean, median, standard deviation σ_{BG} and other statistics values are calculated. We choose the median intensity $\overline{v}_{local}$ as the background intensity associated with each individual spot. Then, a new local foreground cutoff value $v_{forebground\_cutoff}$ is calculated as

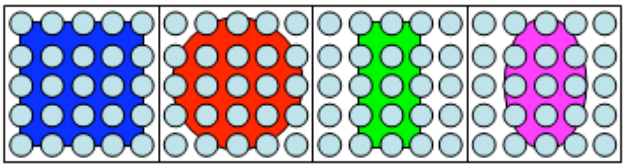
$$v_{forebground\_cutoff} = \overline{v}_{local} + 2\sigma_{BG},$$

which is used to the foreground estimation. In signal region, only pixels with the intensities above this threshold will be classified as foreground pixels from which we compute the median value, which is assumed to be the combination of true signal intensity and background

intensity, $v_{(signal+noise)}$. True signal intensity is revealed using the following in which $v_{ELB} \equiv \bar{v}_{local}$

$$v_{TrueSignal} = v_{(signal+noise)} - v_{ELB}.$$

TIMDA allows flexible ELB configurations in terms of different shapes and sizes as shown in Figure 7, which includes square, circle, rectangle and ellipse. The configurations can be defined by pixel-wise or spot-wise. For a pixel-wise definition, to set the width with 100 and the height with 100, users will get the similar configuration shown in Figure 7, panel (a); for a spot-wise definition, to set a rectangular ELB with size of 5 by 3 in terms number of spots, users will get the configuration shown in Figure 7, panel (c).



(a) (b) (c) (d)

Figure 7. Different allowable ELB configurations.

The interface of the ELB sub-module shown in Figure 8 allows users to monitor each individual spot in the microarray image, *e.g.*, background/foreground intensity, standard deviation, local cutoff values, the number of pixels in the foreground region, *etc.*.

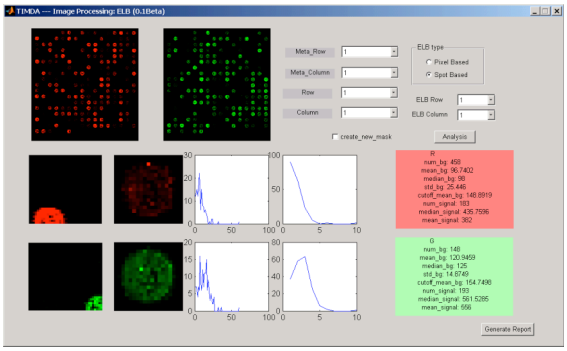


Figure 8. The interface of ELB sub-module.

Figure 9 shows the local background region definition [14] used in some popular microarray image processing software packages including ScanAlyze™, QuantArray™, ImaGene™ and GenePix™. To estimate local background noise, ScanAlyze™ allows user to define a size of a square region, which is similar to our ELB method, however it only allows square regions. QuantArray™, ImaGen™ and GenePix™ use fixed regions. One obvious disadvantage for fixed regions is that if the specific spot is defective for any reasons, the defect will significantly affect the background estimation. Involving more pixels for estimation, ELB is more robust and accurate and makes more accurate statistic estimation. On the other hand, ELB can also tolerate more errors in the gridding stage because of the pre-distinguishing global cutoff threshold value v_{global_cutoff} and local cutoff threshold value $v_{forebground\_cutoff}$ used. Since it is a histogram-based method, it is also not sensitive to the errors in spatial arrangement such as grid locations. ELB is a novel, viable and feasible method for robust microarray image segmentation and quantification.

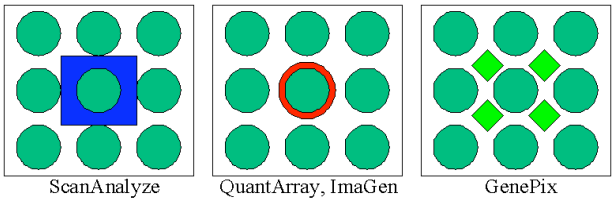


Figure 9. Background region definition used in other analysis tools.

Figure 10 shows the scatter plot of the log value of foreground intensity vs. the value of background intensity obtained using the ELB method in the IP module (left panel) and GenePix™ software (right panel) on the same microarray image. Intuitively, the variance of the background intensity values should be bounded, which is consistent with the random noise model [17]. Relatively large background values are likely the result of misclassification of some foreground pixels. The ELB model results in low variance of background values and thus more robust segmentation.

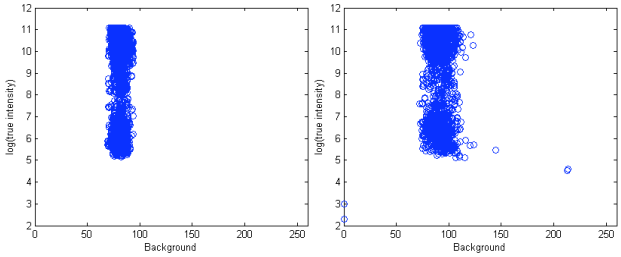


Figure 10. ELB/IP (left) vs GenePix™(right). Y-axis is the log value of foreground intensity. X-axis is the value of background intensity.

Figure 11 illustrates the background value of each spot in a microarray chip obtained using different ELB configurations. The distribution of background values becomes smoother as the ELB definition size gets larger.

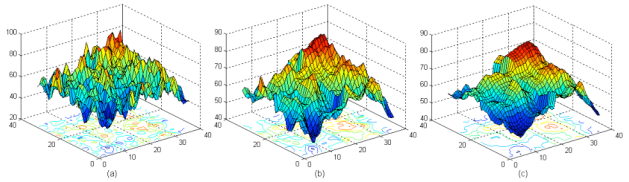


Figure 11. The background values calculated using different ELB configurations. Left, middle and right panels correspond to 3×3, 5×5 and 7×7 spot-based configurations, respectively.

3. Intermediate Data Conversion (IDC)

All text reports of TIMDA are highly structured and could be opened by any text editor or spreadsheet application such as Microsoft™ Exel. Since Excel is not a very strong programming tool, it is necessary to construct a middle layer which integrates different text outputs into Matlab for further analysis. IDC module helps users to convert data into identical format. Figure 12 shows the interface of IDC module. IDC collects data from loaded text files, which include block, column, row, ID, *x* and *y* coordinates, red and green intensities, red and green background intensities. Some known formats have been stored to increase the efficiency. A Matlab's workspace (.mat file) will be created at the IDC module.

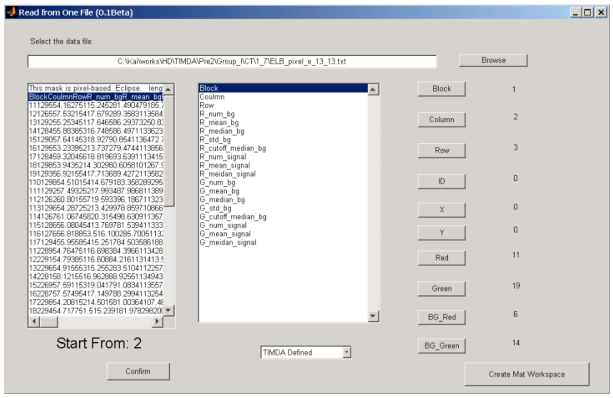


Figure 12. The interface of IDC module.

4. Genotype-Calling (GC)

In two-color SNP genotyping microarray experiments on haploid samples, there are three possible genotypes for each SNP locus: homozygotes "C/C", "T/T" and heterozygote "C/T" on one strand. The core requirement of all genotyping microarrays is to generate accurate genotype calls. GC module allows users to set different preprocessing (background subtraction and normalization) and genotype calling options to generate accurate genotype calls.

4.1 Preprocessing

For background subtraction, GC offers a special global background subtraction method for comparison purpose, which only use spot intensity readings (before background subtraction) to determine a global background intensity threshold. It sorts the spots in descending order in terms of the ratio of intensities, *r/g*, and then from the leading spots: $(P_1,P_2,P_3,...P_n)$, it extracts the mean green intensity being the global red background intensity, meanwhile, from the trailing spots: $(P_{m+1},P_{m+2},P_{m+3},...P_{m+n})$, it extracts the mean red intensity being the global red background intensity. In other words, it finds the red background from the brightest green spots and finds the green background from the brightest red spots. This method is feasible to eliminate non-specific hybridization, but the global background method not considering the variance of the whole chip is its weakness. Users are encouraged to use the true signal readings from subtracting ELB background intensities.

Red and green channels may be imbalanced in digitization of the fluorescent signals, *e.g.,* red signals are systematically stronger than green signals [15], which needs to be corrected using normalization techniques. GC has a set of normalization options. One method is that the spots are sorted in descending order in terms of the red to green intensity ratio *r/g*. The leading $(P_1,P_2,P_3,...P_n)$ and trailing spots $(P_{m+1},P_{m+2},P_{m+3},...P_{m+n})$ are tentatively classified as homozygotes "C/C" and "T/T". Then TIMDA computes the ratio $\overline{r}/\overline{g}$, in which $\overline{r}$ and $\overline{g}$ are the mean intensities of the leading and trailing spots. All intensity readings are then normalized in respect to this ratio. In other words, we use the bright red and green signals to determine the normalization factor. Another method is that heterozygotes are first roughly found using any genotype-calling methods, then $\sum r / \sum g$ is calculated as the channel imbalance factor. This idea is illustrated in Figure 13 in which the circle-shaped data points denote tentative heterozygotes whose distribution should be balanced around the line $y = 0$ if channels are perfectly balanced.

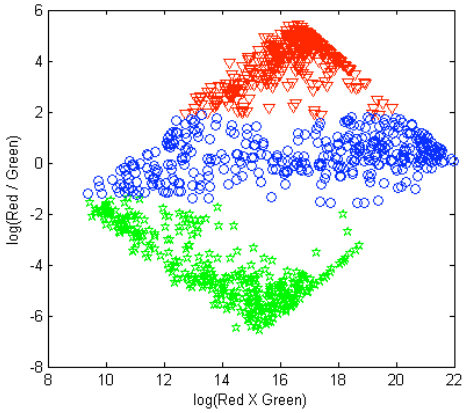


Figure 13. MA plot for a microarray image.

4.2 Auto-Calling

GC integrated a set of algorithms for genotype-calling: the linear cutoff method, logarithm ratio cutoff method, SVM based method and ANN based method, which are very commonly used machine learning approaches. The essence of both ANN and SVM based methods is that through machine learning using a set of training data, we obtain a set of "optimal" classifiers without any human

intervention and thus no human errors. For linear cutoff method, GC calculates $(\log r, \log g)$ and projects to the coordinate system, using $y_1 = x + 2$ and $y_2 = x - 2$ to be the hyperplane to separate three classes, which is similar to Figure 14. The logarithm ratio cutoff method is using $(\log r/g, \log r \times g)$ projection (MA plot). $y_1 = 2$ and $y_2 = -2$ to be the hyper-planes.

SVM is a widely used supervised machine learning method [6] in pattern recognition. It was first introduced to perform binary classification, *i.e.,* determining which class a given data point belongs to. Recent advances have rendered SVMs capable of performing multi-class classification. An SVM determines the parameters of a single learning unit through a virtual linear or nonlinear projection of the input data into a feature space with higher dimension. From our experience, applying the linear SVM can generate the best result, so only the linear SVM is preserved in GC module. SVM guarantees to find the optimal hyper-plane that is least likely to overfit, which makes SVM advantageous in many applications. The one-against-one method is a fairly good extension for multi-class classification from performance and complexity aspects, in which $k(k-1)/2$ SVMs are constructed in a tree structure. Each node represents a distinguishable pair-wise classifier from different classes. The determined class label will be popped up to the root from leaves. During the learning step, the SVM learns from the preprocessed data, which are more easily separable since background noise is subtracted and the channel imbalance are corrected, and then the map is obtained in respect to $(\log r, \log g)$ plot. The testing process just projects the data points into this map. Figure 14 (left panel) shows the linear SVM classifier learned from the training data, in which two straight hyper-planes are not necessarily parallel to each other.

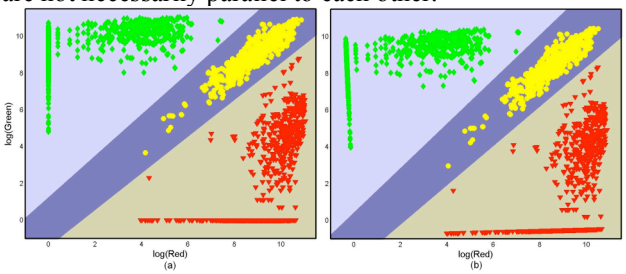


Figure 14. SVM classifiers using the linear kernel (left) and those with iterative adjustment (right).

Classification accuracy could be improved by introducing iterative refinement in the learning and testing stages. The resulting algorithm is termed as GenoIterSVM [18], which consists of a basic SVM as the core and an iterative procedure for improving classification accuracy. With this algorithm, the data are iteratively re-normalized during the learning and the blinded testing stages. Iterative leaning is aim at eliminating variations among different experiments. In SNP genotyping microarrays, usually there are plenty of data points representing heterozygous alleles at DNA loci. Ideally, these heterozygous data should be balanced along the bisector of the first quadrant of the coordinate system, while this is not always true. We can impose an artificial regularity constraint: the optimal curve fitting for these heterozygote data using linear regression analysis must be collinear with $y = x$. If this regularity constraint is not met, we consider that a systematic bias is inherent and must be corrected. Figure 14 (left panel) shows that the collinear constraint is not met for the SVM classifiers obtained after the initial learning stage. We correct this systematic error iteratively by shifting and rotating the coordinate system, until this constraint is met. This iterative learning process results in a system of canonical classifiers, as shown in Figure 14 (right panel). We use a similar iterative approach in the blinded testing step. We found that in most cases it is enough to achieve convergence after one iteration. Using such an procedure, we are able to eliminate the human intervention, such as setting any empirical thresholds, and obtain high accuracy, reliability and efficiency [18].

The use of feed-forward artificial neural networks (FF-ANNs) [7] as classifiers has also become common practice in pattern recognition. FF-ANN is a system of interconnected layers of linear or nonlinear computational units as shown in Figure 15. These FF-ANNs can be described mathematically as a weighted, directed graph in which vertices represent the computational units and the weighted edges the connections among these units. The networks take inputs as samples of measurements from the domain of interest and produce the desired network outputs.

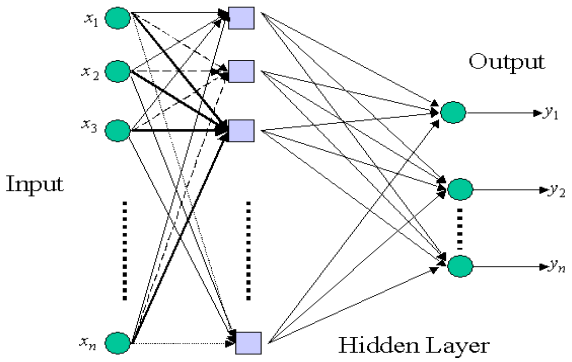


Figure 15. Architecture of a neural network.

Let us consider a FF-ANN with d input nodes, one hidden layer with m nodes, and one output node. The fundamental computing unit of such networks relates a single output, y, to multiple inputs represented by a vector $\vec{x} = \{x_1, ..., x_d\}$ connected by edges represented by the weight matrix $W_i^j = \{w_i^1, ..., w_i^j\}$ in which w_i^j denotes the

weight the edge connecting input node i to hidden node j. The value for the output node is computed using

$$y = f(\sum_{i=1}^{d}\sum_{j=1}^{m} w_i^j x_i - \tau),$$

in which f is the activation function and τ is the threshold value for tolerance control. We replace SVM using FF-ANN as the classifier. Like GenoIterSVM, iterative procedure is also applied to generate the canonical map for FF-ANN, which is termed as GenoIterANN as shown in Figure 16.

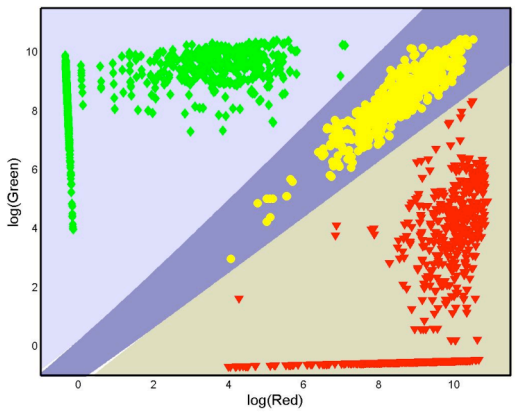


Figure 16. GenoIterANN canonical classifiers .

Figure 17 shows the interface of GC module, in which users can customize the background subtraction, normalization and genotype calling configurations freely. The two plots can be used to monitor the quality of the data set and judge the fitness of configurations. From our experience, different data sets might generate different results based on the different configuration options. Users could configure TIMDA according to their own demands and their experiment environments. Figure 18 shows the graphical output from GC module, in which different shapes represent different genotypes.

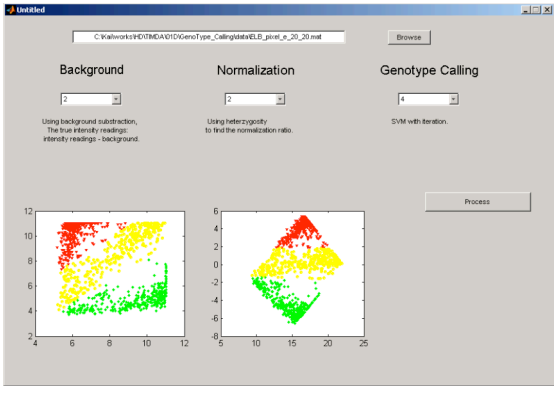


Figure 17. The interface of Genotype-calling.

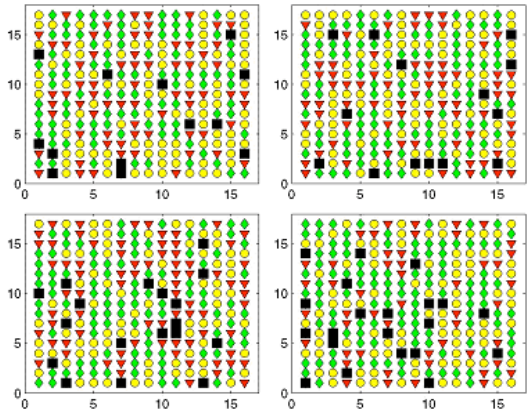


Figure 18. The graphical output of GC module.

Making genotype calls from input images is the main task of routine laboratory workload. To make it easier, TIMDA integrates the IM, IDC and GC module into OneInt module, which minimizes users' intervention during the processing so that users can setup parameters, like the ELB configuration and normalization method, in the very beginning. The interface of OneInt is shown in Figure 19.

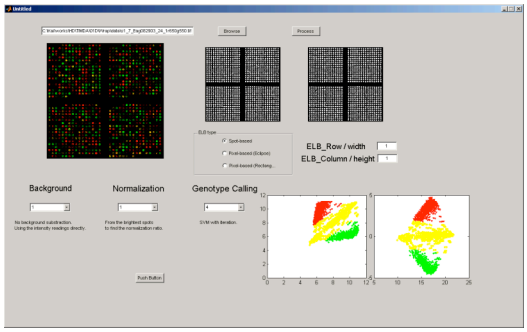


Figure 19. The interface of OneInt.

5. Loss-of-Heterozygosity (LOH)

Loss-of-heterozygosity (LOH) is a good way to monitor the genetic off/on regulation in disease samples in comparison with the normal samples. TIMDA's LOH module is designed to help researchers to analyze the results between normal and disease samples.

The LOH module offers two approaches to analyze LOH: index-based and ID-based. The index-based approach is based on the indexed spot layout, while ID-based approach is based on ID associations. Our LOH experiments are designed to determine genotypes from both sense and anti-sense directions, which are also commonly used in other labs. If ID-based approach is used, before processing LOH comparison, the associations for both directions are determined. Furthermore, replications are also supported to minimize errors. Replication analysis is also supported in the ID-based approach. For example,

for double-replication, ID associations are determined for replicate genes first. If both genes are shown the same genotypes, that genotype will be preserved under the same ID. If they are not the same, that ID will be discarded. In comparison step, two properties are concerned. One is the real loss of heterozygosity, *i.e.,* the heterozygotes change to homozygotes after genotype calling; the other is that the change of the ratio of the log ratios of green to red intensities for heterozygotes in both samples is beyond a threshold. Figure 20 shows the graphical output of LOH module using index-based comparison. The spots shown are heterozygosities in the normal samples. The solid stars mean, these genes are still heterozygosities and the logarithm ratio changes are below the certain threshold. The hollow stars mean they are still heterozygosities, but their logarithm ratio changes are above the threshold. The other hollow shapes indicate genotypes have changed.

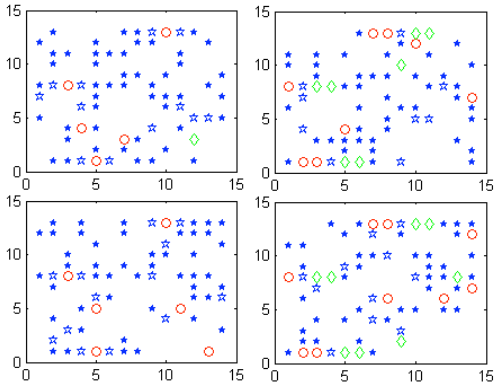


Figure 20. The graphical output of LOH module.

6. Summary

We have shown the design of TIMDA, whose modular design makes the extension and maintenance very convenient. Novel features include the automatic rotation detection and correction, non-parametric thresholding and the ELB local noise model for microarary image processing, and GenoIterSVM and GenoIterANN for auto-calling, which make TIMDA a powerful package with quality, reproducibility and ease-of-use. Using TIMDA, routine microarray data processing can be done more user-friendly and more robustly.

References

[1] D. Amararunga and J. Cabrera, Exploration and analysis of DNA microarray and protein array data, Wiley-Interscience-John Wiley and Sons, Inc., 2004.

[2] S. Dudoit, Y. H. Yang, M. J. Callow and T. P. Speed, "Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments," *Statistica Sinica* 12: 111-139, 2002.

[3] H.-Y. Wang, M. Luo, I. V. Tereshchenko, D. M. Frikker, X. Cui, J. Y. Li, G. Hu, Y. Chu, M. A. Azaro, Y. Lin, L. Shen, Q. Yang, M. E. Kambouris, R. Gao, W. Shih, and H. Li, "A genotyping system capable of simultaneously analyzing >1,000 single nucleotide polymorphisms in a haploid genome," *Genome Research* 15:276-283, 2005.

[4] M. Schena, D. Shalon, R.W. Davis, and P. O. Brown, "Quantitative monitoring of gene expression patterns with a cDNA microarray," *Science* 270: 467–470, 1995.

[5] J. L. DeRisi, V. R. Lyer, and P. O. Brown, "Exploring the metabolic and genetic control of gene expression on a genomic scale," *Science* 278: 680–686, 1997.

[6] C. Cortes and V. Vapnik, "Support vector network", *Machine Learning*, vol. 20, pp. 273–297, 1995.

[7] T. L. Fine, Feedforward neural network methodology, Springer, New York, 1999.

[8] N. Haan and G. Snudden, "Microarrays in the real world: Image analysis," *European Biopharmaceutical Review*, pp. 44–47, 2004.

[9] Y.H. Yang, M.J. Buckley, S. Dudoit and T.P. Speed. "Comparison of methods for image analysis on cDNA microarray data," *Journal of Computational and Graphical Statistics* 11:108-136, 2002.

[10] L. Stefano and Y. Lu, "Gridding and compression of microarray images," *IEEE Computational Systems Bioinformatics Conference*, 2004.

[11] N. Haan and G. Snudden, "Microarrays in the real world: Image analysis," *European Biopharmaceutical Review*, pp. 44–47, 2004.

[12] http://www.cambridgebluegnome.co.uk/

[13] X. Wang, R. Istepanian and Y. Song, "Application of wavelet modulus maxima in microarray spots recognition," *IEEE Trans. Nanobios* 2(4): 190-192, 2003.

[14] http://www.stat.berkeley.edu/~sandrine.

[15] G. C. Tseng, M-K. Oh, L. Rohlin, J. C. Liao and W. H Wong, "Issues in cDNA microarray analysis: quality filtering, channel normalization, models of variations and assessment of gene effects," *Nucleic Acids Res*, 29(12): 2549-2557, 2001.

[16] Y. Wang, M. Ma and F. Shih, "Precise gridding of microarray images by detecting and correcting rotations in subarrays ," JCIS/CBGI, Salt Lake City, Utah, 2005, accepted.

[17] K. Zhang, M. Ma, H, Wang, Y. Wang, T. Wang, F. Shih and P. Soteropoulos, "Robust Microarray Image Segmentation and quantification using extended local background," *JCIS/CVPRIP*, Salt Lake City, Utah, 2005, accepted.

[18] K. Zhang, M. Ma, H, Wang, Y. Wang, M. Banerjee, A. Karmaker, F. Shih, J. Wang, H. Li, "SNP auto-calling using support vector machines," *JCIS/CBGI*, Salt Lake City, Utah, 2005, accepted.