Engineering, **2013, 5, 502-508**
http://dx.doi.org/10.4236/eng.2013.510B103 Published Online October 2013 (http://www.scirp.org/journal/eng)

A Scalable Method for Cross-Platform Merging of SNP Array Datasets

Peikai Chen, Y. S. Hung
Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong, China
Email: pkchen@eee.hku.hk, yshung@eee.hku.hk

Received 2013

ABSTRACT

Single nucleotide polymorphism (SNP) array is a recently developed biotechnology that is extensively used in the study of cancer genomes. The various available platforms make cross-study validations/comparisons difficult. Meanwhile, sample sizes of the studies are fast increasing, which poses a heavy computational burden to even the fastest PC. Here, we describe a novel method that can generate a platform-independent dataset given SNP arrays from multiple platforms. It extracts the common probesets from individual platforms, and performs cross-platform normalizations and summarizations based on these probesets. Since different platforms may have different numbers of probes per probeset (PPP), the above steps produce preprocessed signals with different noise levels for the platforms. To handle this problem, we adopt a platform-dependent smoothing strategy, and produce a preprocessed dataset that demonstrates uniform noise levels for individual samples. To increase the scalability of the method to a large number of samples, we devised an algorithm that split the samples into multiple tasks, and probesets into multiple segments before submitting to a parallel computing facility. This scheme results in a drastically reduced computation time and increased ability to process ultra-large sample sizes and arrays.

Keywords: SNP Array; Scalable Processing; Cross-Platform

1. Introduction

SNP array is a recent advancement in the high throughput biomedical measurements at the molecular level [1]. Each array is a finger-tip sized and one-sample-only biochip, on which millions of grids can be found. Each grid contains millions of identical copies of short single-stranded DNA sequences that are to be bound with the DNAs from the measuring sample. Each grid is called a probe, and a number of probes (usually 6, 12 or 20), referred to as a probeset, may be needed to detect (in a process known as summarization) the signal of an allele (of a locus) in the DNA. In a study, often multiple samples are used on separate arrays. As a result, the levels of DNAs in each sample, though quantified before experiment, may vary and the probe signal benchmarks in each array vary as well. A process known as normalization is usually conducted to reduce this bias [2]. Ultimately, given N samples and N arrays, each with J bi-allelic (namely, A/B alleles) probesets, for a sample $i \in \{1,..,N\}$, there will be J ordered-pairs of signals, denoted as $(S_A^{i,j}, S_B^{i,j})$ $j \in \{1,..,J\}$, where $S_A^{i,j}$ and $S_B^{i,j}$ are the A and B allele signals, respectively. The summarized signals $(S_A^{i,j}, S_B^{i,j})$ can be used for a variety of purposes, such as calling the genotypes of a SNP locus [3], estimating the copy numbers of a gene [4-6], or predicting the LOH likelihood of a region [6]. This genetic information, especially for a disease sample, has strong biological meanings attached to them. Therefore, the proper preprocessing of SNP arrays from probe level signals to summarized signals are very important for correctly identifying the disease-susceptible loci.

Recent years have witnessed the rise of cross-institutional collaborations in biomedical studies, where large numbers of samples can be harvested and measured in different batches by different platforms. Furthermore, similar studies may be conducted using different platforms, making the cross-study comparison extremely difficult. For example, N_1 samples may be in Affymetrix Human Mapping 250 K arrays, while other N_2 samples are in Affymetrix SNP6.0. This presents a problem because the former array contains about a quarter of a million probesets, while the latter contains close to two million probesets (half of which are SNP/allelic and the other half are non-allelic probesets). Researchers may either choose to use samples of only one array, say the one with more probesets, or choose to use the probesets common to both arrays and have a larger sample size. The latter is often more desirable since it will increase the statistical power of a finding. Furthermore, SNP ar-

rays contain far more probesets than the number of genes in the genome, and hence using the common probesets are often good enough in precision. This is particularly true in copy number inference, where the width of copy number events is believed to be usually larger than that between two probesets. By compromising the precision by using the common probests, there can be a strong increase in sample size. This idea is illustrated by the two-platform example in **Figure 1**, where each entry $\mathcal{D}^{i,j}$ in $\mathcal{D}$ is the ordered pair $(S_A^{i,j}, S_B^{i,j})$ mentioned above.

Though important and highly demanded, this idea to merge multiple platform SNP arrays is not well discussed in the literature, perhaps owing to the relatively new SNP array itself. Two recent works related to this topic by Bengtsson *et al.* [7,8] proposed to merge multiple-platform measurements on one sample to produce a full resolution copy number estimate for that particular sample only, as opposed to our proposed objective of merging smaller datasets for a larger one. Multiple-platform comparison and merging of the gene expression microarrays are discussed in [9-11]. Some of the techniques used in these literatures inspired our current approach.

A few literatures also explored the problem of preprocessing large datasets of SNP arrays. For example, Xiao *et al.* [12] explored the scalability of SNP arrays in genotyping. Bengtsson *et al.* [13] discussed the preprocessing of large SNP datasets in bounded memory.

This paper is organized as below. Section 2 will discuss the proposed approach and technical derivations. Section 3 will present the results. Section 4 will interpret the approaches and results.

2. Approach and Algorithms

To implement the idea as depicted in **Figure 1**, three key steps need to be performed:

- *Extracting the common probesets*. Let the set of probesets for Platform 1 (denoted as $\mathcal{P}_1$) be P_1, and those for $\mathcal{P}_2$ be P_2, take intersection of these sets, *i.e.*, $P = P_1 \cap P_2$. Then P is the set of common probesets, and also the set of probesets for resulting datasets $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}$. For individual platforms, this set P represents a truncation of its original probesets.
- *Cross-sample normalization*. Let N_1 and N_2 be the sample sizes of $\mathcal{P}_1$ and $\mathcal{P}_2$, respectively, and denote $(N_1 + N_2)$ as *N*. For each sample $i \in \{1,..,N\}$, obtain the perfect match (PM) probes that correspond to the set P. Let the set of probes in sample i be $\{y_k^i \mid k \in 1,..,K_i\} \triangleq \pi_i$, where K_i is the total number of probes for i. Update y_k^i such that all π_i follow the same empirical distribution.
- *Summarization for each probeset in* P. For a probeset j in sample i, suppose there are L_j^A probes for allele A, then the purpose is to regress $y_{i,\ell} = \mu + \alpha_i + \beta_\ell + \epsilon, \forall\, \ell \in \{1,..,L_j^A\}$ and $i \in \{1,..,N\}$. While α_i is regarded as the sample effect, β_ℓ is regarded as the probe-specific binding affinity. Thus the summarized signal $S_A^{i,j}$ can be obtained by $S_A^{i,j} = \mu + \alpha_i$. Similarly, $S_B^{i,j}$ can be obtained. Repeat this for all i and j, to produce the objective dataset, $\mathcal{D}$.

Since the above two-platform algorithm can be easily generalized to multiple-platform datasets, subsequent discussions in the paper will base on the two-platform example.

However, there are a few issues to be addressed:

- *The numbers of probes per probeset* (*PPP*) *vary from platform to platform*. For example, in SNP250 K, there are on average more than 12 probes per probeset, while in SNP6.0, where the total number probes in an array is roughly the same as SNP250 K but the number of probesets increases by 8 fold, resulting in the PPP being only about 6. This means that overlapping the probesets of two platforms does not automatically create a platform-independent matrix $\mathcal{D}$ that can be readily used for post-processing analyses.
- *Both the numbers of samples and the numbers of probes per array can be huge*. Since multiple-platforms are used, the total sample size is usually large. Further, even after overlapping the probesets, there may still be millions of probes to be handled. This may impose computational time and memory constraints that are beyond the capability a normal PC.

To handle these problems, the following strategies are adopted:

- Cross-platform normalization. Even though the platforms may have variable numbers of probes, it is assumed that this does not affect the overall binding affinity. A cross-platform normalization is key to make sure that the probe signals are calibrated to the same benchmarks.
- Division of samples into tasks, and probesets into

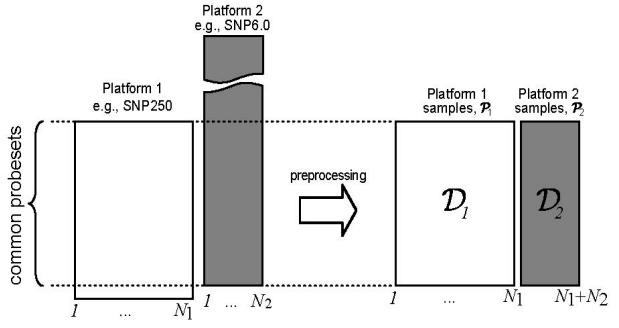

Figure 1. The overall objective is to produce a platform-independent dataset $\mathcal{D} = [\mathcal{D}_1\ \mathcal{D}_2]$, for other analyses, e.g., copy number analysis.

segments. This reduces the computation load for each individual task. The overall tasks can be fed to a parallel computing environment to increase the scalability of the algorithm.

- A platform-dependent summarization scheme. For example, for samples in Platform 1 (*i.e.*, $\mathcal{P}_1$), the summarization by regression is limited to the samples in $\mathcal{P}_1$ only. This ensures that all samples in the regression have the same number of probes per probeset.
- Post-summarization smoothing. Since the platforms have different PPPs. The summarization regression will create inter-platform variability in noise levels. Specifically, in arrays with few PPP, such as SNP6.0, the noise levels tend to be higher. A smoothing scheme at the probeset level can be adopted to adjacent probesets signals, using platform-dependent parameters.

The overall strategy is illustrated by the five-step approach shown in **Figure 2**. The following sub-sections will elaborate on the technical issues of the key steps.

2.1. Cross-Platform Quantile Normalization

Given a sample $i \in \{1,..,N_1\}$ and its probe-level signals $\{y_k^i\}$, single-platform quantile normalization works as follows:

- The set $\{y_k^i\}$ is sorted in increasing order, yielding the sorted vector $\{\hat{y}_k^i\}$ and indices $\{I_k^i\}$, such that $y_k^i = \hat{y}_{I_k^i}^i$.
- Replace $\{\hat{y}_k^i\}$ with a new value $\{\bar{y}_k\}$, which is independent of sample *i*. Usually, $\{\bar{y}_k\}$ is the average of all $\hat{y}_k^i$-s, *i.e.*, $\bar{y}_k = \sum_i \hat{y}_k^i / N_1$.
- Each y_k^i is updated with $\tilde{y}_k^i = \bar{y}_{I_k^i}^i$ before the updated $\{\tilde{y}_k^i\}$ are used for summarization.
- In cross-platform normalization, the platforms have variable numbers of probes. As a result, $\bar{y}_k$ will have to be estimated differently.

To handle this, for a platform p, an auxiliary matrix $A_p \in \Re^{K_p \times N}$ is created, where K_p is the number of probes of p after truncation. Each entry of A_p is given by:

$$A_p^{k,i} = \begin{cases} \hat{y}_k^i, i \in \mathcal{P}_p \\ \hat{y}_\tau^i, i \in \mathcal{P}_o \neq \mathcal{P}_p \end{cases} \quad (2.1)$$

where τ is the index with closest quantile to k in the platform $o \neq p$. That is,

$$\tau = \arg\min(|\, k / K_p - \tau / K_o \,|) \; \forall \tau \in \{1,..,K_o\} \quad (2.2)$$

where $o \in \{1,2 \setminus p\}$ is the platform of sample i, and K_o the number of probes in o.

The matrix A_p contains the platform-independent quantile values. The average quantile values for p can be obtained by $\bar{y}_k^p = \sum_{i=1}^{N} A_p^{k,i} / N$. The remaining procedures in quantile normalization can be done as described above.

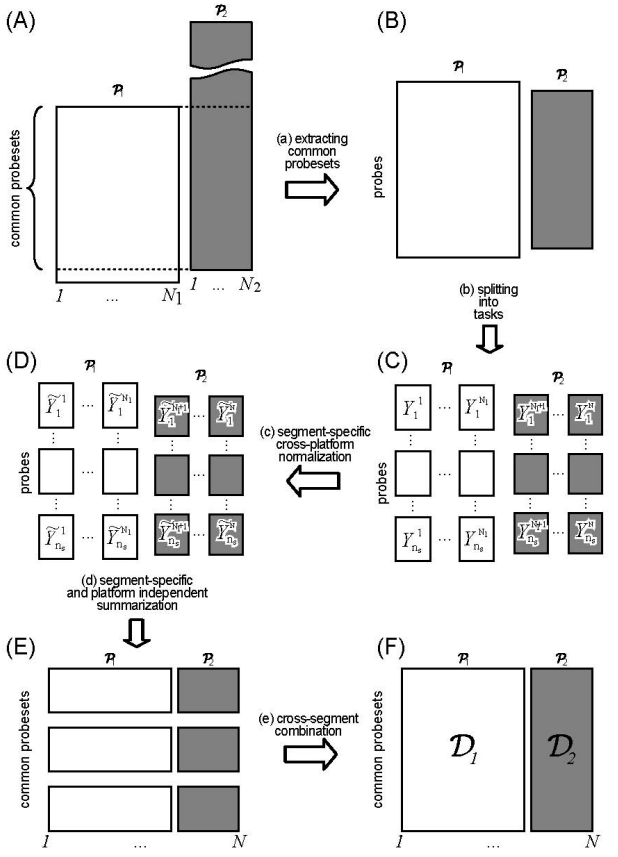

Figure 2. Approach overview. (a) Extracting the common probesets. Note that in the resulting datasets, since SNP6.0 has fewer probes per probeset, it is purposely made to have shorter vertical axis; (b) Splitting the computational task into smaller units, *i.e.*, dividing the samples into tasks, and the probes into segments; (c) Task-specific normalization; (d) Summarization in a segment-specific manner, after which the intermediate results are combined in (e) to produce the dataset $\mathcal{D}$. Note that the preprocessing steps from (a) to (d) are based on probe-level signals.

2.2. Parallelism for Scalability

The above step confers heavy computational burdens, especially when hundreds of samples or more are to be processed. To solve this, a parallel computing scheme is used in the normalization step. It is divided into two steps. First the samples are separately processed. Second, the segments are separately normalized.

2.3. Parallelism in the Samples

The N_1 samples in $\mathcal{P}_1$ are divided into n_1 tasks, and those in $\mathcal{P}_2$ divided into n_2 tasks. In a parallel com-

puting environment, each of these $(n_1 + n_2)$ tasks can be conducted as an individual process, which requires less memory than to process all samples together. The fol- lowing describes the works to be done within each task, *i.e.*, process.

The common probesets P are divided into n_s segments. The corresponding probes are absorbed into each segment. As a result, the probe-level signals $\{y_k^i\}$ for i will also be divided into n_s sets, *i.e.*, $\{y_k^i\} = \{Y_1^i, .., Y_{n_s}^i\}$ (**Figure 2(C)**), where Y_s^i is the set of probes in segment s. Similarly, the sorted probes $\{\hat{y}_k^i\}$ and the indices $\{I_k^i\}$ are divided into n_s sets, *i.e.*, $\{\hat{y}_k^i\} = \{\hat{\hat{Y}}_1^i, .., Y_{n_s}^i\}$ and $\{I_k^i\} = \{\mathcal{I}_1^i, .., \mathcal{I}_{n_s}^i\}$. In a parallel environment, the above segment-specific information needs to be stored in a file-system accessible to all processes.

2.4. Parallelism in the Segments

To complete quantile normalization, a new parallel environment can be created to handle each of the segments by the above step. The following describes the cross-platform normalization within each segment s.

For each platform p, a segment-specific auxiliary matrix $A_{p,s}$ is created and constructed as described in Equation (2.1), such that $A_p = [A_{p,1}, .., A_{p,n_s}]^T$. Next, use the segment-specific information in above step to create the mean quantile values $\bar{Y}_s^p$, such that $\{\bar{y}_k^p\} = \{\bar{Y}_1^p, .., \bar{Y}_{n_s}^p\}$. Finally, update the probe-values of each sample, to obtain $\tilde{Y}_s^i$, such that $\{\tilde{y}_k^i\} = \{\tilde{Y}_1^i, .., \tilde{Y}_{n_s}^i\}$ (**Figure 2(D)**).

This step avoids the overwhelming memory constraints when loading information of all probesets.

2.5. Summarization via Median Polish

Upon the completion of last step, probe-level signals in each segment have been updated with cross-platform quantile values. These probe level signals will be summarized to generate the outputs. To avoid the overwhelming memory requirements for loading these probes, a new parallel environment can be created to handle the probesets in each segment. Median polish regression as described above can be used to perform the summarizations. The intermediate outputs from individual segments can be combined to form an overall dataset $\mathcal{D}$.

2.6. Platform-Dependent Smoothing

The previous step generates a platform-independent output dataset $\mathcal{D} = [\mathcal{D}_1 \ \mathcal{D}_2]$. But as mentioned above, the different platforms have different PPPs, resulting in the variability of noise levels in the summarized signals for samples with different sources of platforms. This is particularly serious in copy number estimations, where the raw total copy number signals (in log-scale), $TCN^{i,j} = S_A^{i,j} + S_B^{i,j}$, are used in subsequent analyses such as GLAD [14] and GISTIC [15].

Since SNP probesets are supposed to measure allele signals in the genome, most physically adjacent probesets are assumed to have the same copy numbers. Therefore, a location-based denoising scheme with source-platform-dependent parameters can be used to benchmark the noise levels of the samples. Specifically, the Nadaraya-Watson kernel estimate [16] can be used.

3. Results

To test the proposed algorithm, we performed a cross-study comparison of copy number aberrations studies on acute myeloid leukemia (AML). In all, two AML datasets were obtained, namely, 176 AML SNP arrays in SNP250K-sty (Affymetrix, CA) from the gene expression omnibus (GEO) (Accession No.: GSE15731), and another 226 AML Affymetrix SNP6.0 arrays from GSE23452. Ninety normal blood samples of European origins (CEU) from the HapMap project in SNP250K were downloaded from the HapMap website (hapmap.org), and will be used for reference. For convenience, the three datasets are referred to as $\mathcal{A}_{250}$, $\mathcal{A}_6$ and $\mathcal{H}$, respectively.

A total number of 222,673 common probesets were extracted, accounting for $\sim 95\%$ and $\sim 25\%$ of SNP probesets in the SNP250 and SNP6.0 platforms, respectively. A 128-process parallel environment was set up to perform the computing task. The 492 SNP arrays were divided into 124 tasks, and the 222,673 probests were divided into 16 segments. It took about 30 minutes to run the whole procedure. **Figure 3** shows the plots before and after quantile normalization of the three datasets. It can be seen that without normalization, the probe-level signals of three datasets are at different benchmarking levels (**Figure 3(A)**), which is especially undesirable in copy number estimates. The normalization pulls the arrays to the same level (**Figure 3(B)**), facilitating the cross-study comparison.

At the end of summarization as described in Section 2, a 222,673-by-402 AML dataset $\mathcal{D}$ were obtained, with each entry being the two-element summarized signals $(S_A^{i,j}, S_B^{i,j})$ as described in Section 1; as well as a 222,673-by-90 HapMap dataset $\mathcal{D}_H$, with each entry denoted as $(H_A^{i,j}, H_B^{i,j})$. To estimate the copy numbers, we use the $\mathcal{D}_H$ dataset as a reference. Specifically, for each sample in $\mathcal{A}_{250}$ and $\mathcal{A}_6$, the copy number signal of probeset j is obtained as: $C^{i,j} = TCN^{i,j} - R^j = S_A^{i,j} + S_B^{i,j} - R^j$, where R^j is the average signal of the j-th probeset in $\mathcal{H}$, *i.e.*, $\$R^j = \sum_{i \in \mathcal{H}} (H_A^{i,j} + H_B^{i,j}) / 90$.

Figure 4 shows the result of 5 samples from each datasets in copy number landscapes. Both datasets are characterized with Chromosome 7 deletion (Chr7d) and

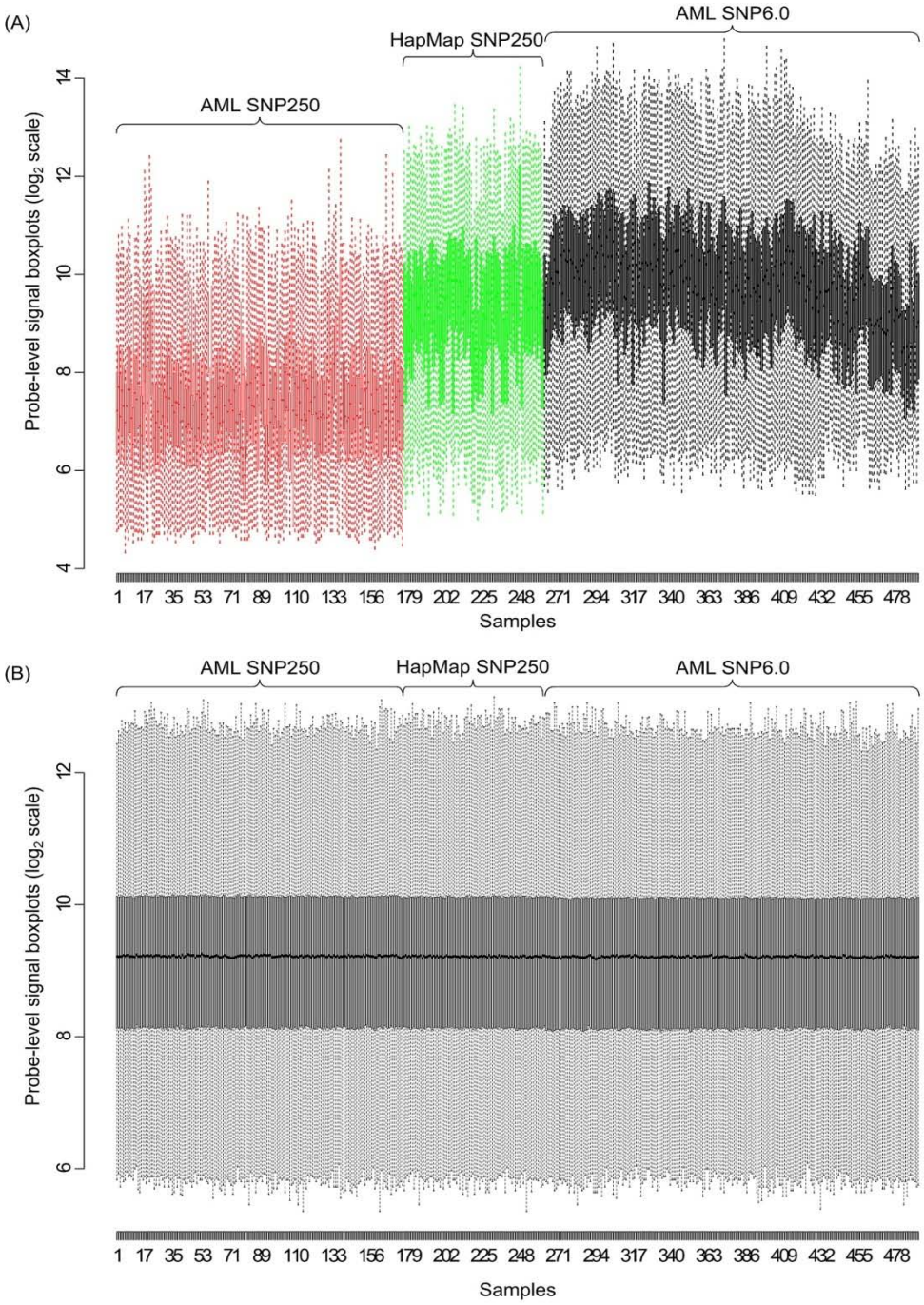

Figure 3. Before and after cross-platform quantile normalization. (A) Probe-level boxplots of the three datasets before normalization. Red, AML SNP250 ($\mathcal{A}_{250}$); Green, HapMap CEU90 SNP250 ($\mathcal{H}$); Black, AML SNP6.0 ($\mathcal{A}_6$); (B) Boxplots for the same samples after cross-platform quantile normalization.

Chromosome 8 amplifications (Chr8a), which is consistent with a recent finding [17]. In **Figure 4(A)**, where the TCNs are not smoothed, it can be seen that the SNP6 samples tend to have higher noise levels. This may cause serious problems in copy number analysis such as GISTIC [15]. A Gaussian window filtering with platform-dependent parameters, *i.e.*, the width of the window, yields noise of the samples that are in the same levels (**Figure 4(B)**).

4. Conclusions

In here we describe a scalable algorithm for merging cross-platform datasets of SNP arrays. It has major

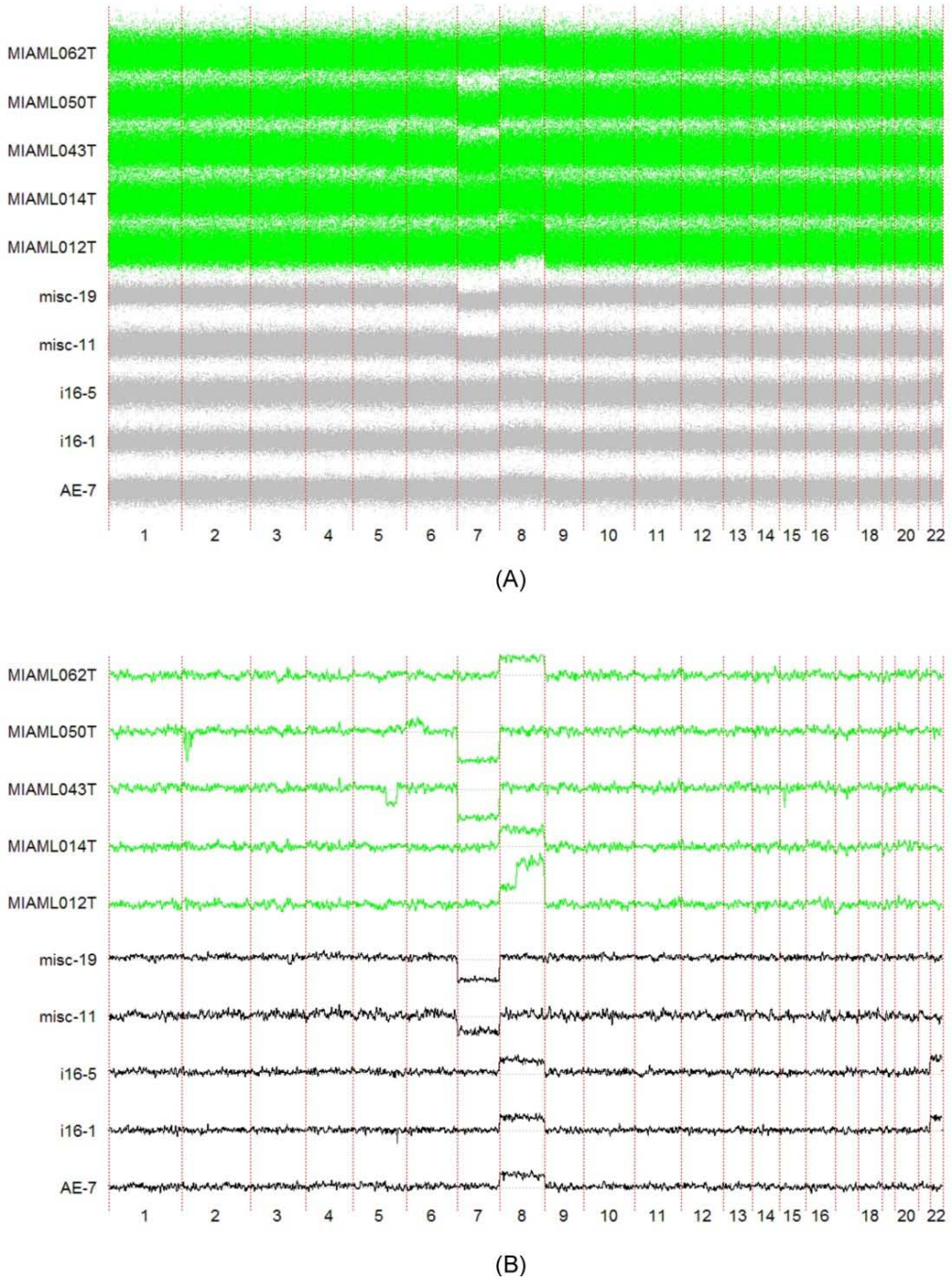

Figure 4. Copy number landscapes of selected samples of the two AML datasets, before and after smoothing. (A) Before smoothing; (B) After smoothing. Horizontal axis is chromosomes. Vertical axis is the copy number signal values. The lower 5 samples in black are from the AML SNP250 dataset ($\mathcal{A}_{250}$), while the 5 green samples are from the AML SNP6.0 dataset ($\mathcal{A}_{6}$).

advantages that may help the biomedical community perform cross-platform study and cross-study comparisons.

First, the cross-platform normalization and source-dependent smoothing make it possible to have a platform-independent dataset $\mathcal{D}$ for probeset-level analyses such as GLAD and GISTIC.

Second, the algorithm is carefully designed to avoid computational time and memory constraints. The breakdown of the original computing task into smaller units vastly increases the scalability of the algorithm so that hundreds of arrays or more can be processed simulta-

neously and completed within a reasonable time frame.

5. Acknowledgements

The work described in this paper is partially supported by the Hong Kong SAR RGC GRF (Project No HKU_762111M) and CRCG of the University of Hong Kong.

REFERENCES

[1] N. Rabbee and T. P. Speed, "A Genotype Calling Algorithm for Affymetrix SNP Arrays," *Bioinformatics*, Vol. 22, No. 1, 2006, pp. 7-12. http://dx.doi.org/10.1093/bioinformatics/bti741

[2] B. Carvalho, H. Bengtsson, *et al*., "Exploration, Normalization, and Genotype Calls of High-Density Oligonucleotide SNP Array Data," *Biostatistics*, Vol. 8, No. 2, 2007, pp. 485-499. http://dx.doi.org/10.1093/biostatistics/kxl042

[3] Affymetrix, "BRLMM: An Improved Genotype Calling Method for the Genechip Human Mapping 500k Array Set," Affymetrix Inc., Tech. Rep., 2006.

[4] Y. Nannya, M. Sanada, *et al*., "A Robust Algorithm for Copy Number Detection Using High-Density Oligonucleotide Single Nucleotide Polymorphism Genotyping Arrays," *Cancer Research*, Vol. 65, No. 14, 2005, pp. 6071-6079. http://dx.doi.org/10.1158/0008-5472.CAN-05-0465

[5] G. Yamamoto, Y. Nannya, *et al*., "Highly Sensitive Method for Genomewide Detection of Allelic Composition in Non-Paired, Primary Tumor Specimens by Use of Affymetrix Single-Nucleotide-Polymorphism Genotyping Microarrays," *American Journal of Human Genetics*, Vol. 81, No. 1, 2007, pp. 114-126. http://dx.doi.org/10.1086/518809

[6] Affymetrix, "Cnat4.0: Copy Numbers and Loss of Heterozygosity Estimation Algorithms for the Genechip Human Mapping 10/50/100/250/500k Array Set," Affymetrix Inc., Tech. Rep., 2007.

[7] H. Bengtsson, P. Wirapati and T. P. Speed, "A Single-Array Preprocessing Method for Estimating Full-Resolution Raw Copy Numbers from All Affymetrix Genotyping Arrays Including Genome-Wide Snp5&6," *Bioinformatics*, Vol. 25, No. 17, 2009, pp. 2149-2156. http://dx.doi.org/10.1093/bioinformatics/btp371

[8] H. Bengtsson, A. Ray, *et al*., "A Single-Sample Method for Normalizing and Combining Full-Resolution Copy Numbers from Multiple Platforms, Labs and Analysis Methods," *Bioinformatics*, Vol. 25, No. 7, 2009, pp. 861-867.

[9] R. Bosotti, G. Locatelli, *et al*., "Cross Platform Microarray Analysis for Robust Identification of Differentially Expressed Genes," *BMC Bioinformatics*, Vol. 8, Supplement 1, 2007, p. S5. http://dx.doi.org/10.1186/1471-2105-8-S1-S5

[10] A. A. Shabalin, H. Tjelmeland, *et al*., "Merging Two Gene-Expression Studies via Cross-Platform Normalization," *Bioinformatics*, Vol. 24, No. 9, 2008, pp. 1154¨C60.

[11] F. Klinglmueller, T. Tuechler and M. Posch, "Cross-Platform Comparison of Microarray Data Using Order Restricted Inference," *Bioinformatics*, Vol. 27, No. 7, 2011, pp. 953-960.

[12] Y. Xiao, M. R. Segal, *et al*., "A Multi-Array Multi-SNP Genotyping Algorithm for Affymetrix SNP Microarrays," *Bioinformatics*, Vol. 23, No. 12, 2007, pp. 1459-1467.

[13] H. Bengtsson, K. Simpson, *et al*., "aroma.affymetrix: A Generic Framework in R for Analyzing Small to Very Large Affymetrix Data Sets in Bounded Memory," Tech. Rep., February 2008.

[14] P. Hupe, N. Stransky, *et al*., "Analysis of Array CGH Data: From Signal Ratio to Gain and Loss of DNA Regions," *Bioinformatics*, Vol. 20, No. 18, 2004, pp. 3413-3422.

[15] R. Beroukhim, G. Getz, *et al*., "Assessing the Significance of Chromosomal Aberrations in Cancer: Methodology and Application to Glioma," *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 104, No. 50, 2007, pp. 20007-20012.

[16] M. G. Schimek, "Smoothing and Regression: Approaches, Computation, and Application," Wiley Series in Probability and Statistics Applied Probability and Statistics Section, Wiley, New York, 2000. http://dx.doi.org/10.1002/9781118150658

[17] M. J. Walter, J. E. Payton, *et al*., "Acquired Copy Number Alterations in Adult Acute Myeloid Leukemia Genomes," *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 106, No. 31, 2009, pp. 12950-12955.

