HICCUP: Hierarchical Clustering Based Value Imputation using Heterogeneous Gene Expression Microarray Datasets

Qiankun Zhao
AOL Labs China
qiankun.zhao@corp.aol.com

Prasenjit Mitra
Penn State University, USA
pmitra@ist.psu.edu

Dongwon Lee
Penn State University, USA
dongwon@psu.edu

Jaewoo Kang
Korea University, Korea
kangj@korea.ac.kr

***Abstract*—A novel microarray value imputation method, HICCUP[1], is presented. HICCUP improves upon existing value imputation methods in the several ways. (1) By judiciously integrating heterogeneous microarray datasets using hierarchical clustering, HICCUP overcomes the limitation of using only single dataset with limited number of samples; (2) Unlike local or global value imputation methods, by mining association rules, HICCUP selects appropriate subsets of the most relevant samples for better value imputation; and (3) by exploiting relationship among the sample space (e.g., cancer vs. non-cancer samples), HICCUP improves the accuracy of value imputation. Experiments with a real prostate cancer microarray dataset verify that HICCUP outperforms existing approaches.**

I. Introduction

Microarray technology allows us to monitor thousands of genes with different types of tissues, species, and experimental conditions. Microarray data has been widely used in many biological applications such as inferring gene function and gene network, drug discovery, and patient diagnosis [1], [4], [6]. In real experiments, often, microarray data contains considerable number of missing values due to various reasons such as insufficient resolution, image corruption, or dusts and scratches on the slides [20]. Because many microarray data analysis techniques require a complete matrix of gene expression values as input, the imputation of missing values in microarray data has attracted a lot of research [3], [20].

Formally, let $G = \mathbb{R}^{m \times n}$ be a gene expression data matrix, which denotes the microarray data of m genes and n samples/experiments, where $m \ll n$. Then, the *Value Imputation Problem* in the context of gene expression values is to estimate a missing value in the l-th location of gene i (g_i), denoted as $G_{i,l}$, using other expression values available in G.

A. Motivation

Existing value imputation approaches can be classified into two categories: *local* and *global*. In the local approach, such as *KNN*, the missing expression value in the l-th sample of gene i, is imputed using the expression values for the l-th sample in a set of similar genes $S = \{g_1, g_2, \cdots, g_j\}$ [20]. In the global approach, a set of basis *eigen-genes* are generated using techniques such as Singular Value Decomposition (SVD), and missing values are imputed using regression models on the eigen-genes [3]. Experimental results show that the performances of local and global approaches vary depending on types of datasets. Although useful as they are, however, existing approaches have the following limitations:

- Most existing methods impute missing values using a single dataset. Because a typical dataset contains a small number of samples with a large number of genes, this lack of samples makes value imputation difficult.
- Since the number of samples in each dataset is limited, it is hard to build a general imputation model that covers many possible *biological properties* (e.g., different stages of cancer and non-cancer samples)
- Although gene expression values consist of two dimensions, *sample* and *gene spaces*, existing approaches tend to focus only on the gene space. The imputation models that un-selectively use all the samples (e.g., cancer and non-cancer samples) may not reflect the different types of correlations within different types of samples.

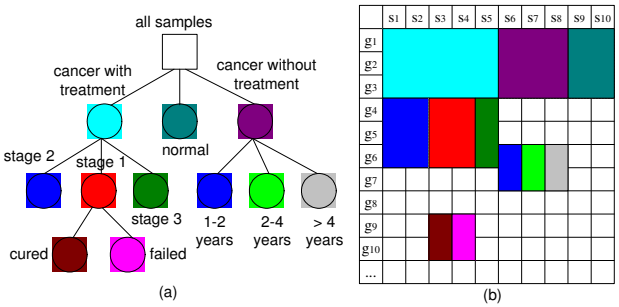

Fig. 1. Microarray and Sample Hierarchy

B. Our Ideas

To overcome the limitations of existing approaches, one may aggregate samples from multiple microarray datasets. However, since microarray datasets from different sources have different characteristics, their integration must be done carefully. For instance, for a gene in an early stage cancer tissue, it is not reasonable to impute missing values using late stage cancer tissues. Rather, we should use early stage cancer tissue samples and experiments conducted under similar conditions as the target sample.

[1]HICCUP stands for *HI*erar*C*hical *C*lustering based val*U*e im*P*utation.

We propose to build a unified imputation model by integrating heterogeneous microarray datasets to make the imputation model robust and accurate. In particular, we propose to utilize a *hierarchical structure* to reflect the correlations in the sample space among those heterogeneous datasets. Correlations in the gene space are modelled using *association rules* [18] within individual clusters of samples in the hierarchy. Given a target sample with missing expression values, the value imputation is conducted using association rules within the most relevant sample cluster. The framework of our HICCUP imputation method is presented in Figure 2.

First, by representing each sample as a vector of expression values, all samples can be clustered into a set of hierarchically organized subsets using the vector-based similarities among samples. Note that, rather than using the whole set of genes, in our approach, the similarities are calculated based on a subset of genes that is selected using an *entropy-based metric*, for each level of the cluster hierarchy.

Next, we extract a set of *discriminative* and *covering gene-values sets* for each cluster to capture the statistical properties. A *gene-values set* consists of a set of genes and its corresponding ranges of expression values. A gene-values set is *discriminative* in a sample cluster if the corresponding genes and value ranges occurred only in this sample cluster but not in any other sample clusters. A set of gene-values sets *covers* a sample cluster if any of the samples in the cluster can be modeled by at least one of the gene-values sets. To extract such discriminative and covering gene-values sets, a *class-dependent discretization* approach is applied to original expression values [7].

Given a target sample with missing gene expression values, relevant clusters are selected from the cluster hierarchy. The intuition is that imputing values using only samples within a cluster that shares similar statistical properties should yield better results. Each sample cluster and the target sample are represented as a vector of discriminative and covering gene-values sets. The *matching score* between a cluster in the hierarchy and the target sample uses the cosine similarity between the two vectors. The sample cluster with the highest *matching score* is selected. Finally, to estimate the missing value, a set of gene-values *association rules* with the target genes in the right hand side (RHS) are extracted and used if there exist any in the relevant cluster.

C. Contributions

Our contributions are as follows:

- Complementing existing approaches using single microarray dataset, HICCUP integrates heterogeneous microarray datasets and chooses the most relevant samples using hierarchical clustering technique for better value imputation.
- Unlike existing approaches focusing only on the gene space, HICCUP explores correlations in both the *sample* space (e.g., using hierarchical clusters) and the *gene* space (e.g., using association rules) across different microarray datasets.
- Experimental results show that the hierarchical clustering based integration of heterogeneous datasets substantially improves imputation quality. In addition, our association rule-based imputation outperforms the local and global imputation approaches.

II. HICCUP: Hierarchical Clustering based Value Imputation

As shown in Figure 2, the first step is to integrate heterogeneous microarray datasets. In this paper, we assume that these datasets share the same gene identification and can be mapped into a single dataset. Given the integrated data collection, the HICCUP imputation consists of the following subtasks: (1) construction of cluster hierarchy, (2) discretization of gene expression value, (3) discriminative and covering pattern extraction, (4) relevant cluster selection, (5) association rule mining, and (6) imputation.

A. Construction of Cluster Hierarchy:

To cluster the samples in microarray datasets into meaningful cluster hierarchy, it is critical to select the appropriate subset of genes. In the literature, Existing work on *subspace clustering* showed how to cluster high dimensional data and partially solved the *curse of dimensionality* [14]. Also, there are works on *biclustering* to cluster gene expression data simultaneously [13]. However, subspace-based clustering and biclustering cannot fully represent the hidden biological properties in our cases as different clusters are partitioned based on different subsets of genes. In our approach, the variances for a single subset of genes are expected to reflect one of the statistical properties, and different subset of genes are selected and used for clustering such that different statistical properties can be represented. Our clustering is similar to the approach in [19], but we are using hierarchical clusters to reflect groups of samples using different subsets of genes at different levels.

Gene selection for sample clustering is the problem of feature selection [12]. Feature selection has been well studied for supervised learning such as classification, where a priori knowledge of the class label of each object is available [5]. Feature selection for unsupervised learning is relatively more difficult. There are two approaches in feature selection for clustering: the *wrapper* approach [8] and the *filter* approach [5]. In this paper, we adopt the filter approach to select different subsets of genes for each level of clusters in the cluster hierarchy. The filter feature selection is based on the observation that values of features that form clusters have very different point to point distance histograms than features that do not form obvious clusters. Consider the following example.

Example 1. Figure 3(a) and (b) show nine samples described by two different sets of gene expression values. For simpler illustration, we only use two genes. It can be observed that the samples in Figure 3(a), which are described by the expression values of $gene_1$ and $gene_2$, do not form any obvious cluster as the points are uniformly distributed. However, in Figure 3(b), samples clearly form three clusters with respect to the expression values of $gene_3$ and $gene_4$. In Figure 3

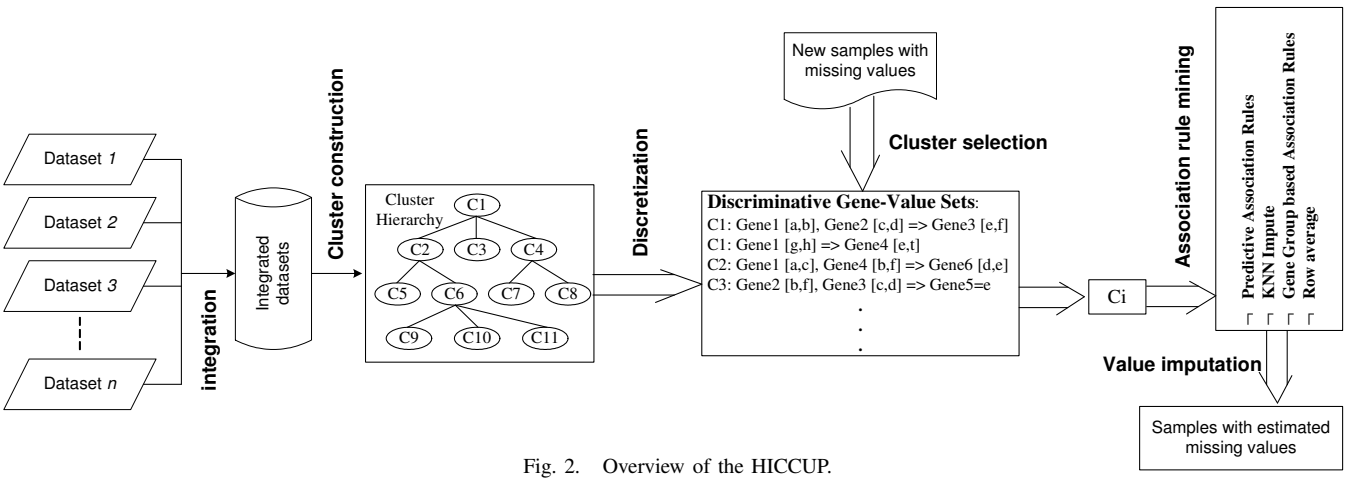

Fig. 2. Overview of the HICCUP.

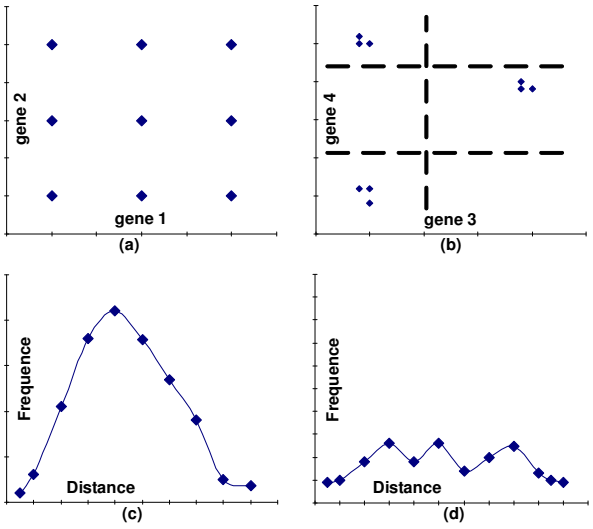

Fig. 3. An Example of Feature Selection

(c) and (d), *distance* refers to the Euclidian distance between any two samples with respect to the two-dimensional gene expression values; *frequency* refers to the number of times a distance value has occurred. From another point of view, the point to point distance distribution with respect to $gene_1$ and $gene_2$ as shown in Figure 3(c) has a predictable bell shape, while the distances with respect to $gene_3$ and $gene_4$ has a very different distribution as shown in Figure 3(d). □

a given subset of genes, denoted as $\mathcal{G}_m$, $A_i = A_{i1}, \cdots, A_{i|\mathcal{G}_m|}$ is the expression values for genes in G in sample *i*, where $|\mathcal{G}_m|$ is the cardinality of the subset of genes. For a set of *n* samples, let D_{ij} be the normalized distance between two samples i and j over the gene subset $\mathcal{G}_m$, the distance-based entropy of a gene subset $\mathcal{G}_m$ is defined as follows:

$$E_{\mathcal{G}_m} = -\sum_i \sum_j (D_{ij} \log D_{ij} + (1 - D_{ij}) \log(1 - D_{ij}))$$

Using the distance-based entropy measure, a subset of genes with the lowest entropy is selected to cluster a given set of samples into smaller clusters. For example, given the microarray data in Figure 1 (b), over the entire sample space, in the first level, we may find out that the subset of genes $\mathcal{G} = \{g_1, g_2, g_3\}$ has the lowest distance-based entropy. Then, the entire samples are clustered into three clusters (children of the root in the hierarchy) based on the similarity calculated over the selected subset of genes.

Algorithm 1 Construction of sample cluster hierarchy.

Input: The integrated microarray matrix *M*
Threshold for the number of sample in each cluster α
Output: A sample cluster hierarchy *H*

1: Initialize root of the cluster hierarchy as $M = \mathcal{C}_0^0$
2: Let $i \leftarrow i' \leftarrow 1$
3: **while** $i \leq i'$ **do**
4: **for all** $\mathcal{C}_i^j \in M$ **do**
5: Select the subset of genes with lowest entropy G_i
6: **while** $|\mathcal{C}_i^j| > \alpha$ **do**
7: Cluster $\mathcal{C}_i^j$ into clusters $\{\mathcal{C}_{i+1}^0, \cdots, \mathcal{C}_{i+1}^n\}$
8: Add $\{ \mathcal{C}_{i+1}^0, \cdots, \mathcal{C}_{i+1}^n \}$ to M
9: j ← j+1
10: $i' \leftarrow i + 1$
11: **end while**
12: **end for**
13: i = i+1
14: **end while**
15: Return H = M

The cluster hierarchy construction algorithm is shown in Algorithm 1. Recursively, each cluster in the *n*-th level of the hierarchy is further partitioned into smaller clusters using a new subset of genes with lowest entropy over samples in this cluster till the number of samples reaches a certain threshold (Line 5-10). Note that $\mathcal{C}_i^j$ refers to the *j*th cluster in the *i*th level of the cluster hierarchy. An illustrative example is shown in Figure 1(a), where Figure 1(a) shows the original dataset and subsets of genes with the lowest entropy over each cluster of samples. The cluster results are shown in Figure 1(a) with the same color scheme. Note that the choice of clustering algorithm is orthogonal to the features and similarity measures. In this paper, we use the graph cut clustering algorithm with Pearson correlation based similarity measure [16].

B. Discretization of Gene Expression Values

In the cluster hierarchy, to extract discriminative gene-values sets, for each group of sibling clusters, the gene expression values are discretized. In this case, each cluster is taken as a class and then a class dependent continuous value discretization method [7] is employed. The basic idea is to find some cut point(s) for a numeric feature such that the intervals in the result are as pure as possible within each cluster. For those features whose values are relatively randomly distributed between different classes of samples, the algorithm will not find any proper cut point, and we should thus discard them for the gene-values set extraction.

Given a subset of samples S', a gene g', and a partition boundary B', the class information entropy of the partition induced by T', denoted as $E(g', T', S')$, is defined as:

$$E(g', T', S') = \frac{S_1}{S'} Ent(S_1) + \frac{S_2}{S'} Ent(S_2)$$

$$Ent(S) = -\sum_{i}^{k} p(C_i, S) \log(p(C_i, S))$$

where k is the number classes and $p(C_i, S)$ is the portion of sample in S that are in class C_i, S_1 and S_2 are the two partitions generated with B'. The partition with minimum entropy is selected and the bi-partition iterates till the number is below the threshold. Essentially, for each group of sibling clusters in the cluster hierarchy, a different discretization scheme is used such that the intervals can effectively reflect the difference among siblings.

C. Discriminative&Covering Pattern Extraction

For each cluster, we propose to extract a set of discriminative and covering gene-values sets. First, we define the *gene-values set* and *frequent gene-values set*. Then, we introduce the concept of *discriminative power* and *coverage*.

Given a cluster of samples $\mathcal{S}_i$ in cluster *i*, each sample $S_{ij} \in \mathcal{S}_i$ consists of a it gene-values set.

Definition 1 (Gene-values Set) *A gene-values set GVS_m is defined to be a set of genes and their corresponding ranges of expression values. A gene-values set can be represented as GVS_i = $\{g_1\ [t_1, t'_1],\ g_2\ [t_2, t'_2],\ \cdots,\ g_n\ [t_n, t'_n]\}$, where g_m $(i \leq m \leq n)$ is a gene and $[t_m, t'_m]$ is a real expression value range for the corresponding gene lying between [0, 1].* □

A sample s_j *supports* a gene-values set GVS_m, denoted as $s_j \prec GVS_m$, if all genes g_i in s_j occur in GVS_m and the expression value of g_i lies in the value range of g_i in GVS_m. The *support* of a gene-values set GVS_i in a sample cluster C_l, *support* (GVS_m, C_l), is the total number of samples in C_l that support GVS_m. Based on the support value, we define the *frequent gene-values set* for a given cluster of samples.

Definition 2 (Frequent Gene-values Set) *Given a cluster of samples C_j, a gene-values set GVS_i = $\{g_1\ [t_1, t'_1],\ g_2\ [t_2, t'_2],\ \cdots,\ g_n\ [t_n, t'_n]\}$ is defined as a frequent gene-values set iff support(GVS_i, C_j) $\geq \theta * |C_i|$, where θ is the user-defined minimal support threshold and $|C_i|$ is the number of samples in the cluster.* □

Given a cluster of samples, there will be a very large number of frequent gene-values sets. We define the relations between different frequent gene-values sets as follows:

Definition 3 (Generality vs. Specificity) *Given two frequent gene-values sets GVS_i and GVS_j, GVS_j is more general than GVS_i iff: $\forall\ g_m \in GVS_j\ :\ \exists\ g_{m'} = g_m \in GVS_i$ such that $t_{m'} \geq t_m$ and $t'_{m'} < t\prime_m$ or $t_{m'} > t_m$ and $t'_{m'} \leq t'_m$. Also, we say GVS_i is more specific than GVS_j. Note that genes exist in GVS_i may not necessarily exist in GVS_j.* □

Given the cluster hierarchy, for each cluster, there will be a set of frequent gene-values sets. We define *discriminative frequent gene-values sets* as follows:

Definition 4 (Discriminative Frequent Gene-values Set) *A discriminative frequent gene-values set X_m for cluster C_m is defined as: X_m = $\{GVS_1, GVS_2, \cdots, GVS_k\}$, where $\forall\ GVS_i \in X_m : \nexists\ any\ GVS_{i'} \in X_p(m) \cup X_s(m)$ such that GVS_i is more general than $GVS_{i'}$, where $X_p(m)$ and $X_s(m)$ are the discriminative frequent gene-values sets for cluster C_m's parent cluster and sibling clusters, respectively.* □

Discriminative frequent gene-values sets can characterize samples in a specific cluster and can distinguish samples in this cluster as much as possible from samples in its sibling clusters. The discriminative frequent gene-values set should have no overlap with its parent sample cluster as well. However, different discriminative frequent gene-values sets have different supports and their discriminative contributions are different. To differentiate the discriminative power of different frequent discriminative gene-values sets, we propose the concept of *discriminative power*.

Definition 5 (Discriminative Power) *Given a cluster C_i, its sibling clusters in the hierarchy is represented as a set of clusters C^s = $\{\ C_1^s, C_2^s, \cdots, C_n^s\}$, the discriminative power of $GVS_j \in X_i$ for cluster C_i is defined as:*
$P(GVS_j, C_i) = \frac{support(GVS_j, C_i)}{|C_i|} \cdot \log \frac{\sum_{m=1}^{n} |C_m^s|}{\sum_{m=1}^{n} support(GVS_j, C_m^s)}$. □

The components of the formula above to determine the discriminative power are similar to *term frequency* and *inverted document frequency* in information retrieval. The importance of a gene-values set increases proportionally to the number of times it appears in the cluster but is offset by how common this gene-values set is in all of the clusters in the entire dataset.

Based on the discriminative power, the set of frequent gene-values sets can be ranked accordingly. Rather than using all the frequent discriminative gene-values sets, we take the top-*k* gene-values sets that cover all the samples in the cluster as the final discriminative and covering gene-values sets. The top-*k* discriminative and covering gene-values sets for a cluster C_i is denoted as $\mathbb{D}(C_i, k)$.

Definition 6 (Top-k Discriminative & Covering GVS) *Given a cluster of samples, C_i, and the corresponding discriminative frequent gene-values sets X_i = $\{GVS_1, GVS_2,$*

$\cdots$, $GVS_n\}$ *in descending order of discriminative power,* $\mathbb{D}(C_i, k) = \{GVS'_1, GVS'_2, \cdots, GVS'_k\}$, *where* $\forall 1 \leq j \leq k$, $GVS'_j = GVS_j$ *and* $\forall s_l \in C_i$, $\exists$ *at least one* $GVS_m \in \mathbb{D}(C_i, k)$ *such that* $s_i \prec GVS_m$. □

The value of *k* is based on the *coverage* of the frequent discriminative gene-values sets. The coverage of a gene-values set with support value of *n* is the set of *n* samples in the cluster that support this set. The top-k discriminative and covering gene-values sets are these *k* discriminative and covering gene-values sets with the largest discriminative powers and they should cover all the samples in the clusters. Note that, to handle the missing expression values, we allow certain redundancy in the frequent discriminative gene-values set. That is, there coverage of some top-k discriminative and covering gene-values sets may have overlaps.

Algorithm 2 Extraction of $\mathbb{D}(C_i, k)$.

Input: A sample cluster hierarchy *H*
Output: $\mathbb{D}(C_i)$ for each C_i

Let C_i represents the sample cluster in hierarchy *H*
Let $G^f_{C_i}$ be the frequent gene-values sets in cluster C_i
for all $C_i \in H$ **do**
  for all $GVS_j \in G^f_{C_i}$ **do**
    Calculate the discriminative power of $P(GVS_j)$ with respect to the sibling clusters of C_i
  end for
  Rank $GVS_j \in G^f_{C_i}$ in descending order of $P(GVS_j)$
  Let k=1
  while $C_i \neq \varnothing$ **do**
    $C_i = C_i - \{s_l | s_l \prec GVS_k\}$
    $\mathbb{D}(C_i) = \mathbb{D}(C_i) \cup GVS_k$
    k = k+1
  end while
end for
Return $\mathbb{D}(C_i, k)$

In this paper, we use the state-of-the-art frequent itemset mining approach, FP-Tree algorithm [9], to extract the sets of frequent gene-values sets for each cluster in the hierarchy. Given, the sets of frequent gene-values sets for each cluster, the algorithm extracts the top-k discriminative frequent gene-values sets (Algorithm 2). The first step is to determine the discriminative power of each gene-values set in a cluster by testing on the sibling clusters. Then, the gene-values sets are ranked in descending order of their discriminative power. *k* top gene-values sets are selected till these sets cover all the samples in the corresponding cluster.

D. Cluster Selection

Given a target sample with missing expression values, the first step is to select the cluster of relevant samples. As the classification power of every individual discriminative and covering gene-values set is limited by its coverage, which is usually not enough, we propose to use the aggregated information of the entire set of discriminative and covering gene-values sets. The cluster selection is performed in a top-down manner. That is, firstly, one of the cluster in the first level of the cluster hierarchy that is most relevant is selected. Then, one of the children of this cluster is selected recursively if the target sample matches any of the clusters. To measure how good the target sample matches a cluster in the hierarchy, we propose the *matching score*.

Definition 7 (Matching Score) *Given the target sample* s_i, *for any cluster* C_j *in the hierarchy, the matching score between the target sample and the cluster is denoted as:* $Score\ (s_i, C_j) = \frac{\vec{s_i} \cdot \vec{C_j}}{|\vec{s_i}||\vec{C_j}|}$, *where* $\vec{s_i}$ *is the vector representation of the target sample with respect to the gene-values set based vector representation* $\vec{C_j}$ *of* C_j. □

HICCUP aggregate the discriminative power of all the gene-values sets in a cluster to represent the sample cluster and the similarity between the target sample with the corresponding cluster is used as the score. Given the matching score for each cluster, the cluster with the largest matching score is chosen as the best matching in this level of the hierarchy. This matching process iterates till no better cluster can be selected.

E. Association Rules Mining & Value Imputation

Given a cluster of samples, without the target sample, we cannot specify the RHS of the association rule because we observed that the number of association rules can be huge even for a single cluster in the hierarchy. As a result, the association rules are generated on the fly with respect to a specific target sample and its missing values.

Suppose we have a target sample s_i with *k* missing expression values for $\mathcal{G}' = \{g'_1, g'_2, \cdots, g'_k\}$ and *n* known expression values for $\mathcal{G} = \{g_1, g_2, \cdots, g_n\}$, for each gene g'_j a set of discriminative frequent gene-values sets that contain any subset of $\mathcal{G}$ and g'_j is selected. The gene-values sets are ranked according to their discriminative power and the corresponding association rules with g'_j in the RHS is generated and ranked according to the *confidence* value. Generally, the confidence of an association rule $X \Rightarrow Y$ is defined as $conf(X \Rightarrow Y) = support(X, Y)\ / support(X)$. For each association rule, r_l, extracted from gene-values set GVS_i in sample cluster C_j related to a target gene g'_j, a priority score $\mathcal{D}(r_l)$ is assigned.

$$\mathcal{D}(r_l) = P(GVS_i, C_j) \cdot Conf(\{GVS_i - g'_j\} \Rightarrow g'_j) \cdot (|GVS_i| - 1)$$

The association rule with the highest priority score is used to estimate the missing value. HICCUP breaks ties by using the overlaps of the estimations of multiple association rules. All the association rules are generated based on the frequent discriminative gene-values sets extracted in the previous phase. However, in some cases, we cannot find any discriminative gene-values sets for a specific sample with certain missing expression values. In such cases, we propose to use the KNNImpute or the row average imputation within this specific cluster of samples. Note that we are not directly using the KNNImpute and row average imputation method on the integrated dataset. Those methods are applied only to samples in the most relevant cluster. As our experimental results shall show in the next section, even by applying KNNImpute and SVDImpute within the relevant sample cluster, the imputation quality can be improved.

III. Experimental Validation

A. Datasets and Evaluation Metric

TABLE I
PROSTATE CANCER DATASETS.

Dataset Name	# Probes	# Normal Samples	# Tumor Samples	Total # Samples
Singh	12600	50	52	102
Welsh	12626	9	24	33
LaTulippe	12626	3	23	26

In the following experiments, we use three prostate cancer microarray datasets, on which the experiments were conducted by different research groups. Details of the three datasets, *Singh* [17], *Welsh* [22], and *LaTulippe* [11], are presented in Table I. As the datasets are obtained possibly with different experimental parameters, we integrate the three datasets by normalizing the expression values to remove the effects arising from variations in the technology and individual variations rather than from the biological differences between samples.

In the above microarray datasets, there are no missing expression values. Our strategy is to randomly pick and drop some of the expression values. As a result, these samples with missing values will be used as testing data, while the rest of the samples are used to construct our imputation model. In our experiments, we shall vary both the percentage of samples with missing values and the percentage of missing values within these samples.

To evaluate the quality of the imputed missing values, in the literature, the *normalized mean squared error*(NRMSE) metric was proposed as:

$$NRMSE = \sqrt{\frac{mean[(y_{impute} - y)^2]}{std[y]}}$$

where y_{impute} is the imputed values of the missing values in the target sample and y is the actual value of the expression value. In existing missing value imputation approaches, real values are the output of the imputation, whereas in our proposed approach, the output is an interval that the missing value belongs to. Note that, the intervals are not as specific as these real values. However, the intervals are semantically as meaningful as the real values as they are obtained from the cluster hierarchy we constructed. In this paper, we extend the NRMSE into the context of intervals. Specifically:

$$y_{impute} - y = \begin{cases} 0, \ if \ y \ is \ within \ the \ interval \ y_{impute} \\ Avg[y_{impute}] - y, \ otherwise \end{cases}$$

B. Performance Comparison

To compare the performance of our proposed approach with existing KNNImpute, SVDImpute, and their variant approaches, the following six different imputation approaches are implemented:

- *KNN-G* denotes the global KNNImpute that uses the whole set of integrated microarray dataset and the gene similarities are calculated using the whole set of genes.
- *KNN-L* denotes the local KNNImpute approach that uses only relevant clusters of samples, and gene similarities are calculated only using the discriminative gene-values sets of these clusters.
- *KNN-S* is similar to KNN-G but uses only single dataset for samples and the gene similarities are calculated using the whole set of genes.
- *SVD-G* represents the global SVD approach that uses the eigen-genes extracted from integrated microarray dataset.
- *SVD-L* is the local SVD approach that uses eigen-genes extracted from the corresponding single dataset.
- *HICCUP* refers to our proposed imputation method, where only samples in the relevant cluster are used and the imputations are obtained by association rule, KNNImpute, and Row Average.

Figure 4 (a) shows the best performance of the six imputation approaches, where *140* samples are used in the model construction process and *21* samples are randomly selected for testing. Note that this figure only compares the best performance of the six imputation approaches. The effects of different parameters will be discussed later. Specifically, for *KNN-G*: *K=13*, for *KNN-L*: *K=6*, for *KNN-S*: *K=11*, for *SVD-G*: *23%* of the eigen-genes are used, for *SVD-L*: *19%* of the eigen-genes are used, for *HICUP*: $\theta = 0.1$, $\alpha = 6$, the confidence threshold is *0.65*.

KNN-L and *KNN-G* outperform *KNN-S* in terms of the imputation quality measured by NRMSE. The reason is that *KNN-S* only uses samples in a single dataset, which may contain very few samples that have similar biological properties to the target samples with missing values, while the *KNN-G* approaches is based on a larger number of samples. Moreover, both the *KNN-S* and *KNN-G* are using similarities calculated using the entire set of genes, while *KNN-L* only uses the subset of discriminative genes in the cluster hierarchy for similarity calculation. As a result, the *KNN-S* and *KNN-G* approaches may not be able to extract the set of really similar genes for the imputation. Similarly, *SVD-G* outperforms *SVD-L*. The reason is that the eigen-genes extracted from the integrated dataset is more accurate to represent all the biological properties of the samples. Overall, our *HICCUP* approach outperforms all of the five other approaches and *KNN*s outperform the *SVD* imputations.

Figure 4 (b) shows another set of experiments that compare the quality of imputation while integrating heterogeneous datasets. Seven different dataset combinations are used: three single datasets, three combinations of two datasets, and one that combines the three together. Note that for each experiment 10% of the samples in the datasets are randomly selected as test samples and 5% of the entries are deleted as missing values. The first character of each dataset is used to represent individual and combined datasets. We see that when more and more samples are combined together the quality of imputation increases.

In summary, the results in Figure 4 (a) and (b) show that: (1) by exploring the correlations between samples, the quality of missing values imputation can be improved even if we use KNNImpute and SVDImpute; (2) the quality of imputation is improved when more and more heterogeneous samples are integrated together, and (3) samples selected using the subset of discriminative-genes-based similarity are more effective for

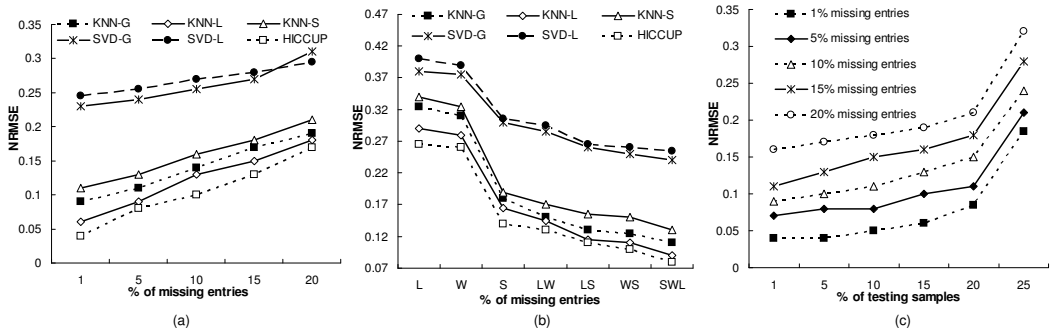

Fig. 4. Experimental Results (1)

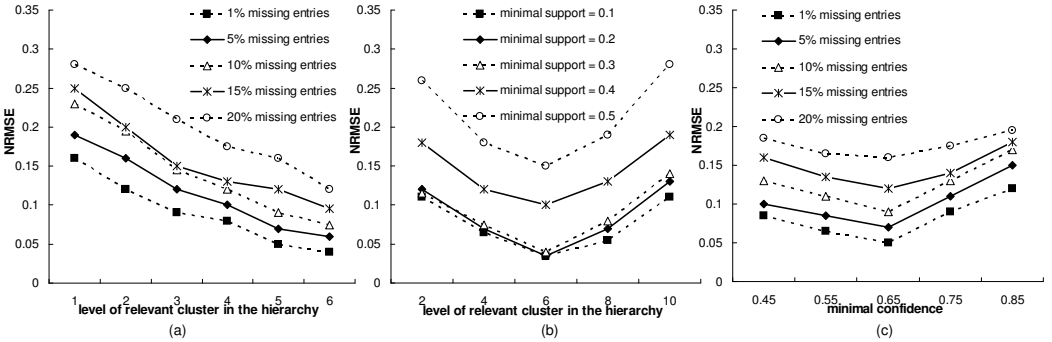

Fig. 5. Experimental Results (2)

missing imputation than the all-genes-based similarity.

C. Performance Evaluation Over Parameters

In this section, we evaluate the robustness of our *HICUP* imputation approach with respect to different parameters: *percentage of samples with missing values*, *percentage of missing entries in the sample*, *the minimal support threshold*, and *threshold for the number of samples in a cluster*.

Figure 4 (c) shows the quality of the imputation with respect to different percentages of samples with missing values. The experiments are conducted with the *161* samples in the integrated dataset. Samples are randomly selected and entries in these samples are randomly deleted as missing values. The minimal support threshold is set to *0.1*, the threshold for the number of samples in a cluster is set to *6*, and the confidence threshold is set to *0.65*. It can be observed that the quality of imputation decreases as the percentage of samples with missing values increases. However, our approach can produce satisfactory imputation with even upto 20% of the samples having missing values.

To evaluate the usefulness of the cluster hierarchy, which consists of *6* levels in all of the above experiments, the quality of imputation is evaluated with respect to the *level* of relevant clusters that are used. Figure 5 (a) shows the statistics of imputation quality using all the above experimental results. It can be observed that: as the *level* of the relevant cluster increases, the quality of imputation increases as well. This is due to the fact that a target sample has to satisfy the discriminative rules for all its parent clusters. As a result, to select the larger *level* of clusters in the hierarchy, the target sample is expected to be more similar biologically to the cluster.

Figure 5 (b) shows the quality of imputation with respect to the threshold for minimal number of samples in a cluster and the minimal support threshold. It can be observed when the minimal support threshold is less than *0.3*, the quality of imputation does not change much, whereas when it increases beyond *0.3*, the quality of imputation decreases quickly. By looking into the gene-values sets, we found that most of the discriminative gene-values sets have a minimal support less than *0.3*. As a result, based on the experimental results, we propose to use *0.1* as the minimal support threshold in the imputation scenario. At the same, this figure shows that *6* is the best threshold for the number of samples in a cluster.

Figure 5 (c) shows the quality of imputation with respect to the threshold for minimal confidence in the association based imputation. We see that when the minimal confidence value is *0.65*, the imputation quality is the best even with different number of missing entries in the samples. The reason is that when the confidence is too small, some of the imputations are not based on high correlations among genes and the quality of imputation decreases. On the other hand, when the minimal confidence is too large, some of the very useful associations are filtered out and the imputation quality decreases as well.

IV. Related Work

Existing missing gene expression value imputation methods can be classified into two categories: *local based approaches*, where genes that are similar to the target genes over the

sample space are selected for the imputation; *global based approaches*, where a set of eigen-genes are selected such that every gene can be representation as the linear combination of these eigen-genes [20]. Experiments with various datasets show that *KNNimpute* appears to provide a more robust and sensitive method for missing value estimation than *SVDimpute*, and both *SVDimpute* and *KNNimpute* surpass the commonly used row average method.

Also there are other imputation methods proposed based on the correlations [3], [15]. Bo, Dysvik, and Jonassen, [3], present a method named *LSimpute* based on the least squares principle. They utilize correlations between both genes and arrays. Experimental results show that *LSimpute* methods produce estimates that consistently are more accurate than those obtained using *KNNimpute*. Also, they compared the *LSimpute* with the expectation maximization (EM) based imputation and show that performance of the *LSimpute* method is at least as accurate as those from the best *EMimpute* algorithm. Sehgal, Gondal, and Dooley proposed a missing value imputation algorithm called collateral missing value estimation (CMVE) using multiple covariance-based imputation matrices for the final prediction of missing values [15]. The matrices are computed and optimized using least square regression and linear programming methods. More recently, Tuikkala, et al., investigated whether semantic similarity originating from gene ontology (GO) annotations could improve the selection of relevant genes for missing value estimation [21]. The results indicated that GO information can enhance the performance of the k-nearest neighbor algorithm when the number of experimental conditions is small and the percentage of missing values is high.

In most of the existing approaches, correlations between genes are considered in the imputation process while the correlations between samples are not fully considered. In our HICCUP approach, both correlations will be considered by utilizing heterogeneous microarray datasets. Moreover, rather than only using the local or global correlations, we use the association rules between different genes over similar samples. We propose a systematic imputation method that integrates existing *KNNimpute* and row average methods with simple and group gene association rules.

More recently, Yoon, Lee and Park proposed to build sample classifier by integrating heterogeneous microarray datasets [23]. However, this approach did not consider missing expression values and human labels of each sample have been used as supervision.

V. Conclusions

In this paper, we proposed the first missing gene expression imputation approach that integrates heterogeneous microarray datasets. Specifically, we construct a cluster hierarchy to model the hidden biological parameters of different clusters of samples. Our association-rule based imputation algorithm achieves a balance between local and global gene value imputation approaches. Experiments conducted with real prostate cancer microarray datasets show that our proposed imputation approach outperformed the existing approaches.

References

[1] D. Allocco, I. Kohane, and A. Butte. Quantifying the relationship between co-expression, co-regulation and gene function. *BMC Bioinformatics*, 5(18), 2004.

[2] N. Altman. Replication, variation and normalisation in microarray experiments. *Applied Bioinformatics*, 4(1):33–44, 2005.

[3] T. Bo, B. Dysvik, and I. Jonassen. Lsimpute: accurate estimation of missing values in microarray data with least squares methods. *Nucleic Acids Research*, 32(3):34, 2004.

[4] P. Clarke, R. Poele, R. Wooster, and P. Workman. Gene expression microarray analysis in cancer biology, pharmacology, and drug development: progress and potential. *Biochem Pharmacol*, 62(10):1131–1136, 2001.

[5] M. Dash, K. Choi, P. Scheuermann, and H. Liu. Feature selection for clustering - a filter solution. In *Proceedings of ICDM*, pages 115–124, 2002.

[6] P. Dhaeseleer, S. Liang, and R. Somogyi. Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics*, 16(8):707–726, 2000.

[7] J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. In *International Conference on Machine Learning*, pages 194–202, 1995.

[8] J. Dy and C. Brodley. Visualization and interactive feature selection for unsupervised data. In *Proceedings of ACM SIGKDD*, pages 360–364, 2000.

[9] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. *2000 ACM SIGMOD*, pages 1–12. ACM Press, 05 2000.

[10] J. Kang, J. Yang, W. Xu, and P. Chopra. Integrating heterogeneous microarray data sources using correlation signatures. In *Data Integration in the Life Sciences Workshop*, pages 105–120, 2005.

[11] E. LaTulippe, J. Satagopan, A. Smith, H. Scher, P. Scardino, V. Reuter, and W. Gerald. Comprehensive gene expression analysis of prostate cancer reveals distinct transcriptional programs associated with metastatic disease. *Cancer Research*, 15(62):4499, Aug 2002.

[12] H. Liu, J. Li, and L. Wong. A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns. *Genome Informatics*, 13(1):51–60, 2001.

[13] S. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 1(1):24–45, 2004.

[14] L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data: a review. *SIGKDD Explor. Newsl.*, 6(1):90–105, 2004.

[15] M. S. Sehgal, I. Gondal, and S. Dooley. Collateral missing value imputation: a new robust missing value estimation algorithm for microarray data. *Bioinformatics*, 21(10):2417–2423, 2005.

[16] J. Shi, and J. Malik. Normalized Cuts and Image Segmentation. *PAMI*, 22(8):888–905, 2000.

[17] D Singh, P G Febbo, K Ross, D G Jackson, J Manola, C Ladd, P Tamayo, A A Renshaw, A V D'Amico, J P Richie, E S Lander, M Loda, P W Kantoff, T R Golub, and W R Sellers. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell.*, 1(2):203, Mar 2002.

[18] R. Srikant and R. Agrawal. Mining quantitative association rules in large relational tables. In H. V. Jagadish and Inderpal Singh Mumick, editors, *Proceedings of the 1996 ACM SIGMOD* , pages 1–12, 4–6 1996.

[19] C. Tang, A. Zhang, and J. Pei. Mining phenotypes and informative genes from gene expression data. In *Proceedings of ACM SIGKDD*, pages 655–660, 2003.

[20] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17(6):520–528, 2001.

[21] J. Tuikkala, L. Elo, O S. Nevalainen, and T. Aittokallio. Improving missing value estimation in microarray data with gene ontology. *Bioinformatics*, 22(5):566–572, 2006.

[22] J B. Welsh, L M. Sapinoso, A I. Su, S G. Kern, J W. Rodriguez, C A. Moskaluk, H F. Frierson, and G M. Hampton. Analysis of gene expression identifies candidate markers and pharmacological targets in prostate cancer. *Cancer Research*, 15(61):5974, Aug 2001.

[23] Y. Yoon, J. Lee, and S. Park. Building a classifier for integrated microarray datasets through two-stage approach. In *IEEE International Symposium on Bioinformatic and Bioengineering*, 2006.