

## **Hierarchical Maximum Likelihood Clustering Approach**

### Author

Sharma, Alok, Boroevich, Keith A, Shigemizu, Daichi, Kamatani, Yoichiro, Kubo, Michiaki, Tsunoda, Tatsuhiko

### Published

2017

### Journal Title

IEEE Transactions on Biomedical Engineering

### Version

Accepted Manuscript (AM)

### DOI

[10.1109/TBME.2016.2542212](https://doi.org/10.1109/TBME.2016.2542212)

### Rights statement

© 2017 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

### Downloaded from

<http://hdl.handle.net/10072/343356>

### Griffith Research Online

<https://research-repository.griffith.edu.au>

# Hierarchical Maximum Likelihood Clustering Approach

Alok Sharma\*, Keith A. Boroevich, Daichi Shigemizu, Yoichiro Kamatani, Michiaki Kubo and Tatsuhiko Tsunoda\*

**Abstract— Objective:** In this work, we focused on developing a clustering approach for biological data. In many biological analyses, such as multi-omics data analysis and genome-wide association studies (GWAS) analysis, it is crucial to find groups of data belonging to subtypes of diseases or tumors. **Methods:** Conventionally, the k-means clustering algorithm is overwhelmingly applied in many areas including biological sciences. There are, however, several alternative clustering algorithms that can be applied, including support vector clustering. In this paper, taking into consideration the nature of biological data, we propose a maximum likelihood clustering scheme based on a hierarchical framework. **Results:** This method can perform clustering even when the data belonging to different groups overlap. It can also perform clustering when the number of samples is lower than the data dimensionality. **Conclusion:** The proposed scheme is free from selecting initial settings to begin the search process. In addition, it does not require the computation of the first and second derivative of likelihood functions, as is required by many other maximum likelihood based methods. **Significance:** This algorithm uses distribution and centroid information to cluster a sample and was applied to biological data. A Matlab implementation of this method can be downloaded from the web-link [http://www.riken.jp/en/research/labs/ims/med\\_sci\\_math/](http://www.riken.jp/en/research/labs/ims/med_sci_math/).

**Index Terms—**Hierarchical clustering, maximum likelihood, biological data.

## I. INTRODUCTION

THE aim of unsupervised clustering algorithms is to partition the data into clusters. In this case, the class label information is unknown; i.e., the knowledge regarding the state of the nature of samples is not provided and clustering is performed by taking into account a similarity or distance measure, distribution information or by some objective functions. In biological data (e.g. genomic data, transcriptomic data) the number of clusters, as well as the location of clusters, are unknown. However, the distribution is assumed (generally

normal Gaussian) in some cases. Therefore, it would be beneficial to develop a scheme that takes into account the distribution information as well.

In the literature, the k-means clustering algorithm has taken a dominant place for biological applications. Recently, in multi-omics data analysis tools like iCluster and iClusterPlus [42], k-means was used as the primary clustering algorithm. In cancer research, analysis tools such as ConsensusCluster (CC) and CCPlus [43], [62] also use k-means as one of the common clustering algorithms. The k-means algorithm has been overwhelmingly applied [25], perhaps due to its simplicity and ability to achieve a reasonable level of accuracy. However, since it uses only the distance between samples to partition the data, it is unable to track clusters when samples of different groups overlap with each other, which commonly occurs in many biological data. Therefore, in such scenarios, k-means may not find accurate clusters, leading to erroneous biological findings, particularly in cancer subtype analysis, GWAS analysis and multi-omics data analysis. Though k-means has played an important role in clustering analysis over the years (including biological analyses), a growing amount of data quantity and complexity requires the development of methods that can perform clustering with a greater level of accuracy.

Apart from the k-means algorithm, several other clustering algorithms have also been developed. Some of the clustering techniques are briefly summarized here as follows: 1) clustering using criterion function, e.g. i) related minimum variance criterion, ii) sum-of-squared error criterion, iii) scattering criterion, iv) determinant criterion, v) trace criterion, vi) invariant criterion [12]; 2) clustering using iterative optimization techniques by employing various criteria functions [18], [11], [16]; [12]; 3) hierarchical clustering [22], [23], [15]; 4) clustering using Bayes classifier [36], [35], [38], [31], [5], [48]; 5) iterative maximum likelihood clustering [9], [41], [10]; 6) likelihood based hierarchical clustering [4], [15]; 7) support vector clustering (SVC) [2], [32], [33] and so on. Recently, SVC has gained widespread attention in clustering [6], [32], [33], [24], [28], [61]. However, for large datasets (e.g. biological data), many of these clustering methods sometimes fail to find meaningful clusters and are also very slow in processing time [30], [26]. For many applications, classifiers like maximum likelihood or Bayes classifier are a preferred choice. There are various ways to implement these clustering methods.

Since this paper concentrates on the maximum likelihood

Manuscript was submitted on 6-Aug-2015. This work was supported in part by the CREST, JST Grant.

Alok Sharma is with (1) RIKEN Center for Integrative Medical Sciences, Yokohama 230-0045, Japan; and, (2) CREST, JST, Yokohama 230-0045, Japan; and, (3) Institute of Integrated and Intelligent Systems, Griffith University, Australia ([alok.f@gmail.com](mailto:alok.f@gmail.com)). Keith A. Boroevich, Yoichiro Kamatani, and, Michiaki Kubo are with (1). Daichi Shigemizu is with (1) and (2), and, Tatsuhiko Tsunoda is with (1), (2) and (4) Medical Research Institute, Tokyo Medical and Dental University, Tokyo 113-8510, Japan ([tatsuhiko.tsunoda@riken.jp](mailto:tatsuhiko.tsunoda@riken.jp)).

method, we summarize some implementations of this method. The maximum likelihood can be computed in the following manners: i) analytical, ii) grid search, or iii) numerical analysis. In practical cases, numerical analysis is typically performed to find the maximum likelihood estimate. In this approach, an initial value parameter is used in a hill climbing algorithm or gradient ascent algorithm (e.g. Newton-Raphson, Berndt-Hall-Hall-Hausman (BHHH), Davidon-Fletcher-Powell (DFP)) to find the maxima. Maximum likelihood is also estimated via an EM algorithm [37], [9], [17], [27], [1], [3], [7], [19], [12]. In these schemes, the initial settings can be crucial, as a bad choice could lead to unreasonable outcomes.

Hierarchical approaches are very well-known clustering methods. These approaches can be subdivided into two categories: agglomerative procedure (bottom-up) and divisive procedure (top-down). An agglomerative procedure begins by considering each sample as a cluster and at each step, the two clusters which are closest to each other under some similarity measure are merged. This procedure continues until only one cluster exists. This gives a tree structure known as dendrogram. A divisive procedure performs clustering in a way inverse to the agglomerative procedure. It starts by considering one cluster (containing all the data samples) and splits the cluster into two clusters at each step until all the clusters contain only one sample [29], [12]. In this paper, we consider only the agglomerative procedure for hierarchical clustering. The hierarchical approach is independent of initial parameter settings. It can be carried out by linear or non-linear regression models [49], [45], [15]. Usually in these methods, a joint likelihood is computed which is a triple integral (of joint probability, normal and gamma density functions) and is computed by the fourth-order Gauss-Lobatto quadrature [15]. This makes the computation quite expensive. In some cases, to make computation simpler, a Markov Chain Monte Carlo approach is used to estimate the dendritic tree [4].

Over the years, several hierarchical approaches have been proposed. Here we summarize a few schemes. Single linkage or link (SLink) [57] merges two nearest-neighbor clusters at a time in an agglomerative hierarchical fashion. It uses the Euclidean distance to measure the closeness between two clusters (if it is less than an arbitrary threshold). This method is very sensitive to data position, sometimes creating issues by generating clusters composed of a long chain (known as chaining effect). The complete linkage (CLink) hierarchical approach [8] depends on the farthest-neighbor and reduces the chaining effect. This technique is also sensitive to outliers. The use of the average distance could be a way to overcome this sensitiveness. This was done in the average linkage (ALink) hierarchical approach [59], [34]. It computes the average distance between two clusters for linking. Similarly, the median linkage (MLink) hierarchical approach [14] uses the median distance for linking. In the weighted average distance linkage (WLink) hierarchical approach [46], [39], cluster sizes are disregarded when computing average distances. As a result, smaller clusters will get a larger weight in the clustering process [46]. Vaithyanathan and Dom [63] developed a model-based hierarchical clustering by utilizing

an objective function based on a Bayesian analysis. They used multinomial likelihood function and Dirichlet priors, and applied their strategy on document clustering. Similarly, hierarchical clustering of a mixture model was proposed by Goldberger and Roweis [20] and applied on scenery images and handwritten digits. Their method optimized the distance between two Gaussian mixture models. They have assumed that the desired number of clusters is predefined.

In this work, we developed a hierarchical maximum likelihood (HML) clustering algorithm. We derive the HML method, such that there is no need to compute triple integrals or to find first and second derivatives of likelihood functions. The proposed technique can also deal with small sample size cases, where data dimensionality is higher than the number of samples, by considering the range space of covariance matrices (of clusters) during the clustering process. Since the clustering equations are derived from Gaussian models, the algorithm will be more suitable for data that follows a Gaussian distribution. We provide mathematical derivation of the method. Experiments were conducted on both simulated and real data to exhibit the performance of the proposed method compared with other state-of-the-art methods.

## II. OVERVIEW OF MAXIMUM LIKELIHOOD CLUSTERING

In this section, we briefly describe the maximum likelihood method for clustering [12]. Let a  $d$ -dimensional sample set be  $\chi = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$  having  $n$  unlabelled samples. Let  $c$  be the number of clusters and  $\Omega = \{\omega_j\}$  be the state of the nature or class label for  $j$ th cluster  $\chi_j$  (for  $j = 1, 2, \dots, c$ ). Let  $\theta$  be any unknown parameter (having mean  $\mu$  and covariance  $\Sigma$ ). Then the mixture density is given by

$$p(\mathbf{x}|\theta) = \sum_{j=1}^c p(\mathbf{x}|\omega_j, \theta_j) P(\omega_j) \quad (1)$$

where  $p(\mathbf{x}|\omega_j, \theta_j)$  is the conditional density,  $\theta = \{\theta_j\}$  (for  $j = 1 \dots c$ ) and  $P(\omega_j)$  is the a priori probability. The log likelihood can be given by joint density

$$L = \log p(\chi|\theta) = \log \prod_{k=1}^n p(\mathbf{x}_k|\theta) = \sum_{k=1}^n \log p(\mathbf{x}_k|\theta) \quad (2)$$

If the joint density  $p(\chi|\theta)$  is differentiable with respect to  $\theta$  then from Equations 1 and 2

$$\nabla_{\theta_i} L = \sum_{k=1}^n \frac{1}{p(\mathbf{x}_k|\theta)} \nabla_{\theta_i} \left[ \sum_{j=1}^c p(\mathbf{x}_k|\omega_j, \theta_j) P(\omega_j) \right] \quad (3)$$

where  $\nabla_{\theta_i} L$  is the gradient of  $L$  with respect to  $\theta_i$ . If  $\theta_i$  and  $\theta_j$  are independent and suppose a posteriori probability is given as

$$P(\omega_i|\mathbf{x}_k, \theta) = \frac{p(\mathbf{x}_k|\omega_i, \theta_i) P(\omega_i)}{p(\mathbf{x}_k|\theta)} \quad (4)$$

then from Equation 4, we can see that  $\frac{1}{p(\mathbf{x}_k|\theta)} =$

$\frac{P(\omega_i|\mathbf{x}_k, \boldsymbol{\theta})}{p(\mathbf{x}_k|\omega_i, \boldsymbol{\theta})P(\omega_i)}$ . Substituting this value in Equation 3 and since for any function  $f(x)$  its derivative  $\partial \log f(x) / \partial x = 1/f(x) \cdot f'(x)$ . We have

$$\nabla_{\boldsymbol{\theta}_i} L = \sum_{k=1}^n P(\omega_i|\mathbf{x}_k, \boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}_i} \log p(\mathbf{x}_k|\omega_i, \boldsymbol{\theta}_i) \quad (5)$$

Equation 5 can be equated to zero ( $\nabla_{\boldsymbol{\theta}_i} L = 0$ ) to obtain maximum likelihood estimate  $\hat{\boldsymbol{\theta}}_i$ . The solution can therefore be obtained by

$$P(\omega_i) = \frac{1}{n} \sum_{k=1}^n P(\omega_i|\mathbf{x}_k, \hat{\boldsymbol{\theta}}) \quad (6)$$

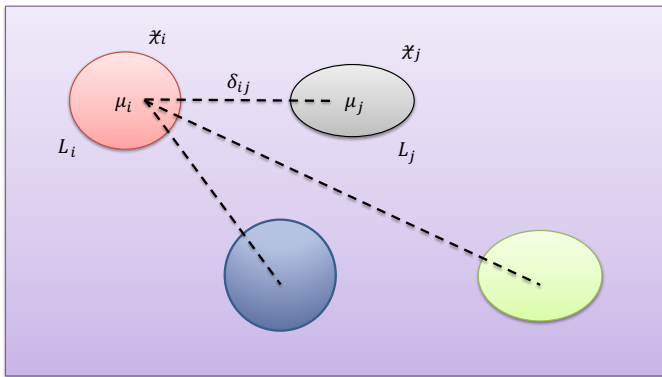
$$\sum_{k=1}^n P(\omega_i|\mathbf{x}_k, \hat{\boldsymbol{\theta}}) \nabla_{\boldsymbol{\theta}_i} \log p(\mathbf{x}_k|\omega_i, \hat{\boldsymbol{\theta}}_i) = 0 \quad (7)$$

$$P(\omega_i|\mathbf{x}_k, \hat{\boldsymbol{\theta}}) = \frac{p(\mathbf{x}_k|\omega_i, \hat{\boldsymbol{\theta}}_i)P(\omega_i)}{\sum_{j=1}^c p(\mathbf{x}_k|\omega_j, \hat{\boldsymbol{\theta}}_j)P(\omega_j)} \quad (8)$$

For a normal distribution case, the parameter  $\boldsymbol{\theta}$  is replaced by the unknown mean  $\boldsymbol{\mu}$  and covariance  $\Sigma$  parameters in the above equations to yield maximum likelihood estimates. In the literature, the parameter  $\boldsymbol{\theta}$  is iteratively updated to reach the final value  $\hat{\boldsymbol{\theta}}$  using the hill climbing algorithms.

### III. HML METHOD

Here we describe the proposed HML method for clustering. For  $n$  samples, the search starts at level  $n$ , where two clusters are merged at a time such that the overall likelihood maximizes (an illustration is given in Fig. 1). In the hierarchical framework, there is no need for initial parameter settings and hence the solution is unique in contrast with iterative optimization techniques. In order to develop the maximum likelihood estimate in the hierarchical framework, we address two fundamental issues: 1) what is the criterion function; and, 2) what is the distance or similarity measure that satisfies the selected criterion function.



**Figure 1:** Illustration of the hierarchical maximum likelihood method. In this case, four clusters are given and two closest clusters are to be merged. A similarity measure  $\delta_{ij}$  is used to find the closeness of clusters. Two clusters  $\chi_i$  and  $\chi_j$  with likelihood functions  $L_i$  and  $L_j$  are merged such the total likelihood is maximized.

To investigate these two issues, we defined the class-based log-likelihood of two clusters  $\chi_i$  and  $\chi_j$  as

$$L_i = \sum_{\mathbf{x} \in \chi_i} \log[p(\mathbf{x}|\omega_i, \boldsymbol{\theta}_i)P(\omega_i)] \quad (9)$$

and similarly,  $L_j$  can be derived accordingly.

It is important to know how the class-based log likelihood functions (called as log-likelihood here after) change if two clusters are merged. For this, suppose mean and covariance of  $\chi_i$  and  $\chi_j$  are defined as  $\boldsymbol{\mu}_i, \Sigma_i$  and  $\boldsymbol{\mu}_j, \Sigma_j$ , respectively. The mean and covariance functions are expressed as follow:

$$\boldsymbol{\mu}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in \chi_i} \mathbf{x} \quad (10)$$

$$\Sigma_i = \frac{1}{n_i} \sum_{\mathbf{x} \in \chi_i} (\mathbf{x} - \boldsymbol{\mu}_i)(\mathbf{x} - \boldsymbol{\mu}_i)^T \quad (11)$$

where  $n_i$  is the number of samples in  $\chi_i$ . The expressions for  $\boldsymbol{\mu}_j$  and  $\Sigma_j$  can be derived accordingly. If the component density is normal and a priori probability is defined as  $P(\omega_i) = n_i/n$  (where  $n$  is the total number of samples) then Equation 9 can be written as

$$L_i = n_i \log P(\omega_i) + \sum_{\mathbf{x} \in \chi_i} \log \left[ \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_i|^{\frac{1}{2}}} \exp \left[ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_i)^T \Sigma_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i) \right] \right]$$

or

$$L_i = -\frac{1}{2} \text{tr} \left[ \Sigma_i^{-1} \sum_{\mathbf{x} \in \chi_i} (\mathbf{x} - \boldsymbol{\mu}_i)(\mathbf{x} - \boldsymbol{\mu}_i)^T \right] - \frac{n_i d}{2} \log 2\pi - \frac{n_i}{2} \log |\Sigma_i| + n_i \log \frac{n_i}{n}$$

where  $\text{tr}()$  is a trace function. Since  $\text{tr}[\Sigma_i^{-1} \sum_{\mathbf{x} \in \chi_i} (\mathbf{x} - \boldsymbol{\mu}_i)(\mathbf{x} - \boldsymbol{\mu}_i)^T] = \text{tr}(n_i I_{d \times d}) = n_i d$ , we can write  $L_i$  as

$$L_i = -\frac{1}{2} n_i d - \frac{n_i d}{2} \log 2\pi - \frac{n_i}{2} \log |\Sigma_i| + n_i \log \frac{n_i}{n} \quad (12)$$

Similarly,  $L_j$  can be formulated. The total log-likelihood for  $c$  clusters can be written as

$$L_{tot} = \sum_{k=1}^c L_k \quad (13)$$

where  $L_k$  is from Equation 12.

If clusters  $\chi_i$  and  $\chi_j$  are merged then the resultant mean and covariance can be given as

$$\boldsymbol{\mu}_i^* = \frac{1}{n_i + n_j} (n_i \boldsymbol{\mu}_i + n_j \boldsymbol{\mu}_j) \quad (14)$$

$$\Sigma_i^* = \frac{1}{n_i + n_j} \left[ (n_i \Sigma_i + n_j \Sigma_j) + \frac{n_i n_j}{n_i + n_j} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T \right] \quad (15)$$

The determinant of  $\Sigma_i^*$  can be written as

$$|\Sigma_i^*| = \frac{1}{(n_i + n_j)^d} |Q| \quad (16)$$

where

$$Q = (n_i \Sigma_i + n_j \Sigma_j) + \frac{n_i n_j}{n_i + n_j} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T \quad (17)$$

We can now obtain the change in  $L_i$  after merging two clusters  $\chi_i$  and  $\chi_j$  as

$$L_i^* = -\frac{1}{2}(n_i + n_j)d - \frac{(n_i + n_j)d}{2} \log 2\pi - \frac{(n_i + n_j)}{2} \log |\Sigma_i^*| + (n_i + n_j) \log \frac{(n_i + n_j)}{n} \quad (18)$$

After rearranging Equation 18 and from Equation 12, we get

$$L_i^* = L_i + L_j + (n_i + n_j) \log(n_i + n_j) - (n_i \log n_i + n_j \log n_j) - \frac{(n_i + n_j)}{2} \log |\Sigma_i^*| + \frac{n_i}{2} \log |\Sigma_i| + \frac{n_j}{2} \log |\Sigma_j| \quad (19)$$

The value of  $|\Sigma_i^*|$  from Equation 16 can be substituted in Equation 19, which will give  $L_i^*$  as

$$L_i^* = L_i + L_j + \delta_{ij} \quad (20)$$

Since  $\delta_{ij}$  is a similarity measure to compute the closeness between two clusters, it can be multiplied by a constant without affecting its decision. Here we multiply the similarity by 2 to take out the halves factor which appeared in Equation 19. We get the similarity measure as

$$\delta_{ij} = f_\lambda + f_N \quad (21)$$

where

$$f_\lambda = n_i \log |\Sigma_i| + n_j \log |\Sigma_j| - (n_i + n_j) \log |Q| \quad (22)$$

and

$$f_N = (d + 2)(n_i + n_j) \log(n_i + n_j) - 2n_i \log n_i - 2n_j \log n_j \quad (23)$$

So in summary, the two clusters should be merged if the similarity  $\delta_{ij}$  between the two is maximum compared to all the other cluster pairs as this would maximize the likelihood function  $L_{tot}$  (of Equation 13); in other words, choose cluster  $(i, j)$  such that the overall  $L_{tot}$  is maximized; i.e.,  $(i^*, j^*) = \arg \max_{i,j} \delta_{ij}$ .

The second concern for the algorithm is to find the number of clusters in the data. If the number of clusters ( $c$ ) is known, then the algorithm can be executed until the desired number  $c$  is obtained. If a rough estimate is given ( $a \leq c \leq b$ ) then the  $L_{tot}$  curve in the range  $[a, b]$  can be considered and  $c$  can be estimated for which  $L_{tot}$  is maximum. If no information about  $c$  is known, then the algorithm can be run for all clusters  $[1, n]$  and the best value can be obtained by using the  $L_{tot}$  curve.

Furthermore, some other functions related to  $L_{tot}$  can be developed to find the best value of  $c$ . The HML method is given in Table 1.

Table 1: Hierarchical Maximum Likelihood (HML) method

1.	Let $r = 1$ , $\chi_i = \{\mathbf{x}_i\}$ , $\Sigma_i = I_{d \times d}$ and $\boldsymbol{\mu}_i = \mathbf{x}_i$ , $i = 1, 2, \dots, n$ .
2.	While $r \leq n - c$ (if unknown $c$ then $c = 1$ ).
3.	Find pair $\chi_i$ and $\chi_j$ for which $\delta_{ij}$ is maximum.
4.	Merge two clusters $\chi_i \leftarrow \chi_i \cup \chi_j$ and delete $\chi_j$ . Compute $L_{tot}$ after the merger.
5.	Increment $r$ and go to step 2.

It can be observed from Table 1 that when  $r = 1$  we have assumed covariance of a sample to be an identity matrix as it is not possible to obtain a non-zero covariance of a cluster having only one sample. However, this would reduce  $f_\lambda$  to  $-2 \log |Q|$  and  $f_N$  to  $2(d + 2) \log 2$  (in Equations 22 and 23); i.e., the merger of clusters at  $r = 1$  mainly depend on  $f_\lambda$  as  $f_N$  is constant. Therefore, when  $r = 1$ , we can consider  $\delta_{ij} = f_\lambda$  (in Equation 21).

It is possible to have the number of samples in a cluster less than the data dimensionality  $d$ . This would lead to a small sample size (SSS) problem.

#### IV. SMALL SAMPLE SIZE CASE OF THE HML METHOD

As discussed earlier, if the dimensionality of samples is higher than the number of samples in a cluster, it creates an SSS problem. In this situation, the covariance matrices will become singular and their determinant will become zero [50], [51], [52], [53]. Thereby, no solution can be obtained. Moreover, if  $d$  is very large, the computation of the covariance matrix is expensive. In this case, the rectangular matrix can be computed as follows:

$$\Sigma_i = H_i H_i^T \quad (24)$$

$$\text{where } H_i = \frac{1}{\sqrt{n_i}} \hat{H}_i \in \mathbb{R}^{d \times n_i} \quad (25)$$

$$\text{and } \hat{H}_i = [\mathbf{x}_1 - \boldsymbol{\mu}_i, \mathbf{x}_2 - \boldsymbol{\mu}_i, \dots, \mathbf{x}_{n_i} - \boldsymbol{\mu}_i] \quad (26)$$

where  $\mathbf{x} \in \chi_i$ . The singular value decomposition (SVD) of  $H_i$  would give  $U_i D_i V_i^T$ . Let the rank of  $H_i$  be  $r_i$ . This will give  $r_i$  non-zero eigenvalues in  $D_i$ . Since  $\Sigma_i = U_i D_i^2 U_i^T$ , the eigenvalues of  $\Sigma_i$  will be squared of the eigenvalues of  $H_i$ . Let  $\lambda_i^k > 0$  be the  $k^{\text{th}}$  eigenvalue of  $\Sigma_i$  (where  $k = 1, 2, \dots, r_i$ ). Since  $|\Sigma_i|$  is same as  $|U_i D_i^2 U_i^T|$  or  $|D_i^2| |U_i^T U_i|$  and  $U_i$  is an orthogonal matrix, we can write  $|\Sigma_i| = |D_i^2| = \prod_{k=1}^{r_i} \lambda_i^k$ . Now computation of  $\delta_{ij}$  (Equation 21) can be done by using non-zero eigenvalues. This, in turn, requires us to solve Equation 22 as

$$f'_\lambda = n_i \sum_{k=1}^{r_i} \log(\lambda_i^k) + n_j \sum_{k=1}^{r_j} \log(\lambda_j^k) - (n_i + n_j) \sum_{k=1}^q \log(\lambda_q^k) \quad (27)$$

where  $\lambda_j^k$  is the  $k^{\text{th}}$  eigenvalue and  $r_j$  is the rank of  $\Sigma_j$ . Similarly,  $\lambda_q^k$  is the  $k^{\text{th}}$  eigenvalue and  $r_q$  is the rank of  $Q$  (Equation 17). Since  $Q$  is a symmetric matrix, it can be written as  $Q = H_q H_q^T$ . Rectangular matrix  $H_q$  can be computed as (from Equation 17)

$$H_q = \left[ \sqrt{n_i} H_i, \sqrt{n_j} H_j, \sqrt{\frac{n_i n_j}{n_i + n_j}} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) \right] \in \mathbb{R}^{d \times (n_i + n_j + 1)} \quad (28)$$

From Equations 25 and 26, we can write Equation 28 as

$$H_q = \left[ \hat{H}_i, \hat{H}_j, \sqrt{\frac{n_i n_j}{n_i + n_j}} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) \right] \quad (29)$$

Similarly, when dimensionality  $d$  is very large compared to the number of samples per cluster then we have to approximate  $f_N$  as the ranks of covariance matrices are no longer  $d$ . To approximate  $f_N$ , we assume if  $d > n/4$  then the rank of covariance (or some confidence limit for eigenvalues of covariance) of data could be used instead of  $d$ . We call  $d_{eff}$  the rank of covariance of data (or effective dimension). Therefore, in Equation 23 we use  $d_{eff}$  in place of  $d$  when the dimensionality is large (as described before). This will approximate  $f_N$  as  $f'_N$ .

Therefore, rather than computing similarity  $\delta_{ij}$  from Equation 22, we can compute from Equation 27 and  $f'_N$  as

$$\delta_{ij} = f'_\lambda + f'_N \quad (30)$$

As discussed earlier, at the start of the algorithm, when  $r = 1$  (Table 1), all clusters will have 1 sample each and covariance for each cluster is assumed to be identity. In this case (when  $r = 1$ ), we can use  $\delta_{ij} = f'_\lambda$  which is basically  $-2 \sum_{k=1}^{r_q} \log \lambda_q^k$ .

To verify if similarity  $\delta_{ij}$  (of Equation 30) can work well on high dimensional case, we created two random clusters having  $n_1 = 100$  samples in cluster 1 and  $n_2 = 50$  samples in cluster 2. The dimensionality was varied as  $d = 2, 10$  and  $2000$ . Cluster 2 is moved from location 1 to location 10 as depicted in Fig. 2. At each location, the similarity  $\delta_{ij}$  is measured. It is expected that as cluster 2 reaches close to cluster 1, the similarity  $\delta_{ij}$  increases. If the dimensionality  $d$  is high ( $d \gg n$ ), the same characteristics should be observed.

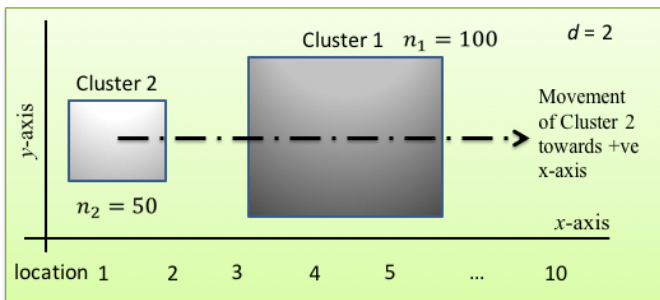
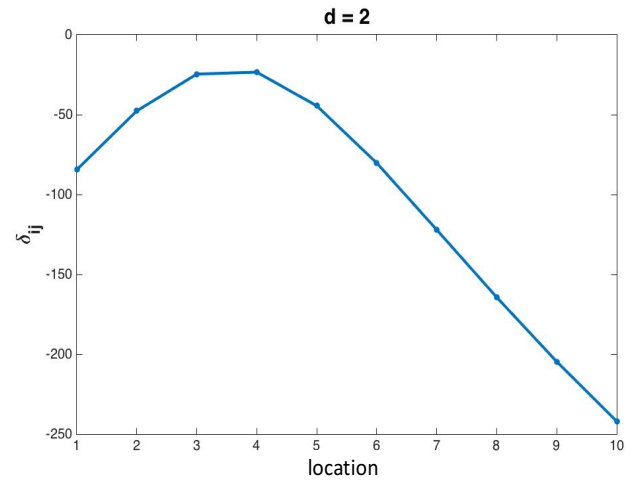
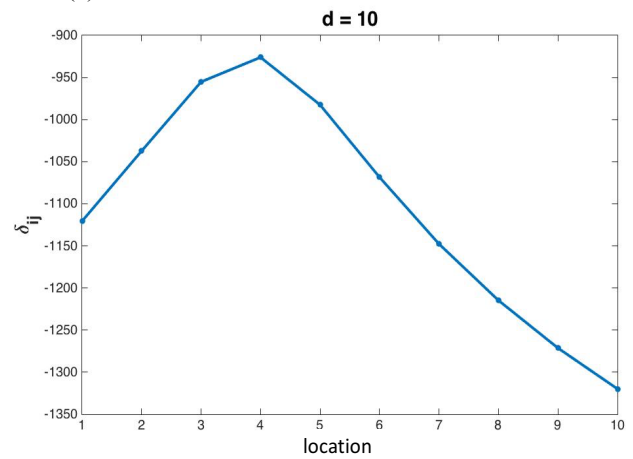


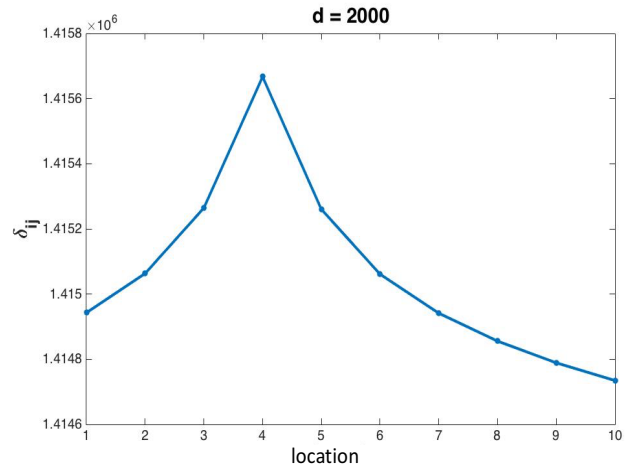
Figure 2: An illustration to verify similarity measurement (using  $d = 2$ ).



(a)



(b)



(c)

Figure 3: Behavior of similarity measure at different location with varying dimensions

It can be seen from Fig. 3a ( $d = 2$ ), that the similarity  $\delta_{ij}$  is maximized around location 4. A similar performance is observed when  $d = 10$  (Fig. 3b). If we set  $d$  to 2000, we observe similar characteristics (Fig. 3c) as of  $d = 2$  and  $d = 10$ . This shows that the similarity measure  $\delta_{ij}$  can work effectively when the dimensionality is high by providing the

same closeness information as when the dimensionality is low.

### V. SEARCH COMPLEXITY OF HML METHOD

In this section, we briefly describe the number of searches required by the agglomerative hierarchical clustering method. Since hierarchical clustering is based on the greedy algorithm, the search is generally quite expensive, of the order  $O(n^3)$ . However, here we tried to improve the search by efficiently handling the similarity matrix, reducing the HML search to  $O(n^2)$ .

Fig. 4 illustrates the HML method using 4 samples. At level  $n = 4$ , each sample is a cluster and hence there are 4 clusters. The nearest clusters using similarity  $\delta_{ij}$  are merged (in Fig. 4a, clusters 1 and 4 are merged). At the next level ( $n - 1 = 3$ ), the nearest clusters are merged again. This process is continued. It can be observed that at level  $n$ , distance or similarity is measured from a cluster to all other clusters giving  $\frac{1}{2}n(n - 1)$  search (Fig. 4b). At any level  $n - k$  the search would be  $\frac{1}{2}(n - k)(n - k - 1)$ . Therefore, the total search can be given as

$$S = \frac{1}{2} \sum_{k=0}^{n-2} (n - k)(n - k - 1) = \frac{1}{6} (n - 1)n(n + 1) = O(n^3) \quad (31)$$

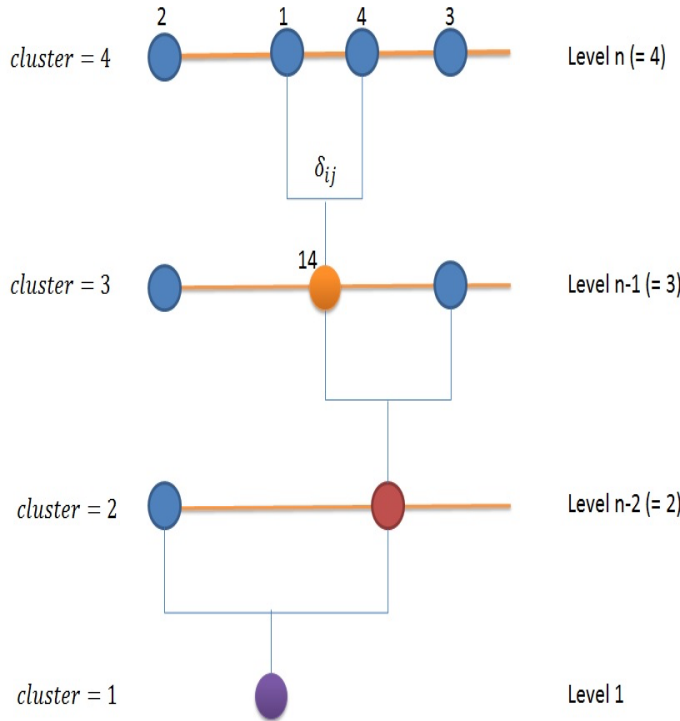


Figure 4a): A dendrogram for HML.

If the two clusters 1 and 4 are merged, we do not need to compute  $\frac{1}{2}(n - k)(n - k - 1)$  distances or similarities (where  $k = 1$  at level 3) in the next level. From Fig. 4c, we can observe that from the merged cluster 14, two new distances or similarities ( $d_{12}^*$  and  $d_{34}^*$ ) are calculated. However, the distance or similarity  $d_{23}$  is the same as before. Therefore, the

search can be reduced.

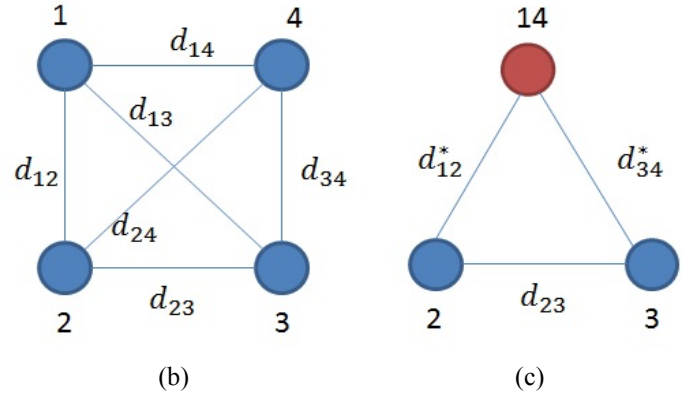


Figure 4: b) Distance or similarity computation at level  $n = 4$ ; c) Distance or similarity computation after a merger of two clusters for HML.

Consider the computation of the distance or similarity matrix when 6 samples are given in a dataset (Fig. 5a).

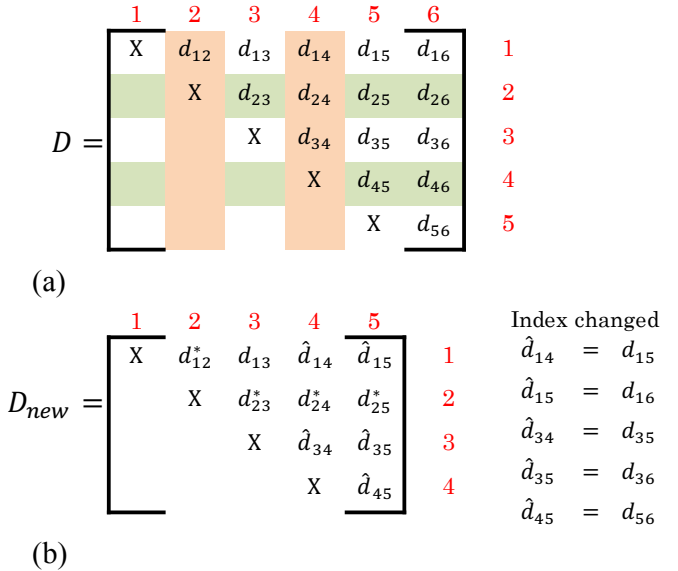


Figure 5: Distance or similarity matrix computation in HML.

At level 6, there are six clusters which would give 15 distances in a distance matrix  $D$ . Suppose clusters 2 and 4 are merged at this level. Then rows 2 and 4, and columns 2 and 4 will be deleted from  $D$ . In the next subsequent level, there will be 5 clusters. Distances between the merged cluster and all the remaining 4 clusters will be computed which will give  $d_{12}^*$ ,  $d_{23}^*$ ,  $d_{24}^*$  and  $d_{25}^*$  (Fig. 5b). For all the remaining distances those were not deleted at level 6, will have new indices (as shown in Fig. 5b) at level 5. This would give a new distance matrix  $D_{New}$  with 4 computed distances and 6 remaining distances (some with changed indices). Therefore, at level  $n - k$ , the required search is  $n - k - 1$ . The total search can now be given as follows:

$$S^* = \frac{1}{2}n(n - 1) + \sum_{k=1}^{n-2} (n - k - 1) = (n - 1)^2 = O(n^2) \quad (32)$$

## VI. EXPERIMENTS AND RESULTS

We carry out analysis on artificial (normal Gaussian) data as well as on biological data to evaluate the performance of HML. We divide this section into 3 subsections. Subsection A shows the performance of hierarchical methods using Gaussian data and microarray data. Subsection B describes the  $L_{tot}$  related curves to estimate number of clusters; and, in subsection C we describe the HML clustering method on genomic data. We have also given an illustration using four clusters (including SVC algorithm) in Supplement 1.

### A. Clustering on Gaussian data and gene expression data

In this section, we use Gaussian data of dimensionality  $d$  (similar topology as shown in Suppl. 1, Fig. S1a having 4 clusters with a total of 400 samples). We generated the data 20 times (using a different random seed), and for each time, we computed clustering accuracy. In order to get a statistically stable value, we computed average (mean) clustering accuracy over 20 attempts. We carried out this exercise for dimensionality  $d = 2 \dots 500$  (2, 3, ..., 19, 20, 25, 30, ..., 500). For comparison purposes, we used various hierarchical based clustering methods like SLink, CLink, ALink, WLink and MLink. The average clustering accuracies for various methods over dimensionality  $d$  are depicted in Fig. 6. It can be observed from Fig. 6 that when the dimensionality is relatively low the performance of HML is quite promising over the other hierarchical based clustering methods. However, as the dimensionality increases, the performance of various methods does not improve. For the HML method, the data distribution information is captured using covariance matrices of clusters. However, when the dimensionality is very large compared to the number of samples per cluster then covariance matrix will become singular and its determinant will become zero. In this case, we need to approximate the covariance matrix to overcome the ill-posed matrix issue. Furthermore, in this case it is difficult to get distribution information. Therefore, it is expected that performance will deteriorate if the dimensionality is very large. We can also observe from the figure that when the dimensionality is high ( $d \geq 100$ ), many clustering methods appear to converge. This is because these methods tend to accumulate most of the samples in a small number of dominant clusters, missing the other remaining clusters. In the case of HML, it estimates the covariance matrix of a cluster by considering the eigenvectors corresponding to the leading eigenvalues (basically a few non-zero eigenvalues). Since these few eigenvalues represent the dominant orientation of the data distribution, the estimated model becomes sensitive towards leading direction. Nonetheless, the HML method is able to produce a reasonable level of performance compared to other hierarchical based clustering methods.

Next, we generated another set of artificial (normal Gaussian) data 50 times (by changing the random seed), and produced boxplots for various hierarchical methods over selected data dimensionalities. The results are depicted in Supplement 2.

Thereafter, we utilized microarray gene-expression datasets, namely acute leukemia [21] and prostate tumor [58] data to measure the performance (in terms of clustering accuracy) of various clustering methods. The details of these datasets are as follows:

Acute leukemia dataset – this dataset consists of DNA microarray gene expression data of human acute leukemias for cancer classification. Two types of acute leukemia data are provided for classification, namely acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML). The dataset consists of 72 bone marrow samples (47 ALL and 25 AML) and over 7129 probes. All the samples have 7129 dimensions and all are numeric.

Prostate tumor dataset – this is a 2-class problem addressing tumor class versus normal class. It contains 77 prostate tumor samples and 59 non-tumor (or normal) samples. Each sample is described by the expression of 12,600 genes.

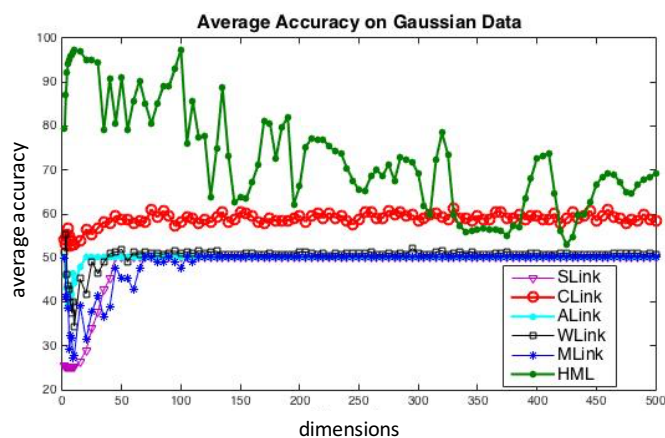


Figure 6: Average clustering accuracy of various hierarchical based clustering methods on Gaussian data.

The expression data need not be Gaussian. In order to vary the data dimensionality (number of genes), we utilized Chi-squared feature selection method to rank the genes. We then performed cluster analysis (to evaluate clustering accuracy) on dimensionality  $d = 2, 5, 10, 20, 100, 200$  and 1000. The clustering accuracies on acute leukemia and prostate tumor are reported in Tables 2 and 3, respectively. It can be seen from Table 2 that CLink, ALink, MLink, WLink and HML provided reasonable performance. HML lead when  $d \leq 20$  and when  $d = 1000$ . It was able to reach 95.8%. For prostate tumor (Table 3), HML was able to achieve 75.7% clustering accuracy. It can also be observed that when the dimensionality is large, many methods tend to accumulate most of the samples in a small number of (in this case one) dominant clusters. For example, in the case of acute leukemia dataset (Table 2), out of total of 72 samples, most of the methods clustered 71 samples to a class and clustered only one sample to another class. Consequently, most of the methods showed a clustering accuracy of around 66.7%. It appeared to converge but in fact it was accumulating most of the samples in the wrong cluster. Therefore, increasing the dimension further doesn't produce better results for most of the methods and thus we stopped the evaluation at this point.

Furthermore, we can see that until  $d = 20$  the clustering accuracy on prostate tumor dataset (Table 3) by HML was around 55%. But when dimensionality increased further ( $d \geq 100$ ), the clustering accuracy reached 75.7%. The reason for this could be that the gene ranking method (Chi-squared method which is a filter-based feature selection scheme) and clustering methods are mutually independent techniques. Therefore, the genes are ranked independent of the clustering method used. For higher dimensionality, HML tries to estimate the covariance matrix using the leading eigenvalues of the data distribution. It is not necessary that these leading eigenvalues correspond to the highest ranked genes (obtained by the Chi-squared method). Therefore, increasing the number of genes gives new possibility of improving or deteriorating the performance of the classifier. This phenomenon can be observed in other methods too. In Table 3, CLink produced 58.1% clustering accuracy when  $d = 2$  and when the dimension was increased to  $d = 5$ , it gave 50.7%. However, going further up to  $d = 10$  gave 61.8% but dropped down after  $d = 20$ . In ALink, higher clustering accuracy is observed when  $d = 5$  and  $d = 10$ , but lower for  $d = 2$  and  $d \geq 20$ . In WLink, it is higher for  $d = 5$  and  $d = 200$ , but lower for  $d = 2$  and the remaining dimensions. Also in MLink, clustering accuracy is higher for  $d = 5$  but lower for  $d = 2$  and  $d \geq 10$ .

**Table 2:** Clustering accuracy on acute leukemia dataset.

Dim	SLink	CLink	ALink	WLink	MLink	HML
2	66.7%	84.7%	76.4%	94.4%	94.4%	95.8%
5	66.7%	81.9%	84.7%	81.9%	81.9%	95.8%
10	66.7%	81.9%	81.9%	73.6%	73.6%	93.1%
20	66.7%	73.6%	76.4%	76.4%	66.7%	95.8%
100	66.7%	68.1%	70.8%	76.4%	81.9%	70.8%
200	66.7%	66.7%	66.7%	66.7%	66.7%	63.9%
1000	66.7%	66.7%	66.7%	66.7%	66.7%	76.4%

**Table 3:** Clustering accuracy on prostate tumor dataset.

Dim	SLink	CLink	ALink	WLink	MLink	HML
2	57.4%	58.1%	58.1%	58.8%	58.1%	54.4%
5	55.2%	50.7%	61.8%	61.8%	61.8%	55.2%
10	55.2%	61.8%	61.8%	51.5%	54.4%	55.2%
20	55.2%	61.8%	55.2%	55.2%	55.2%	53.7%
100	55.2%	61.0%	55.2%	55.2%	55.2%	75.7%
200	55.2%	50.0%	55.2%	61.0%	55.2%	75.7%
1000	55.2%	58.8%	55.2%	55.8%	55.8%	71.2%

**B. Estimation of the number of clusters**

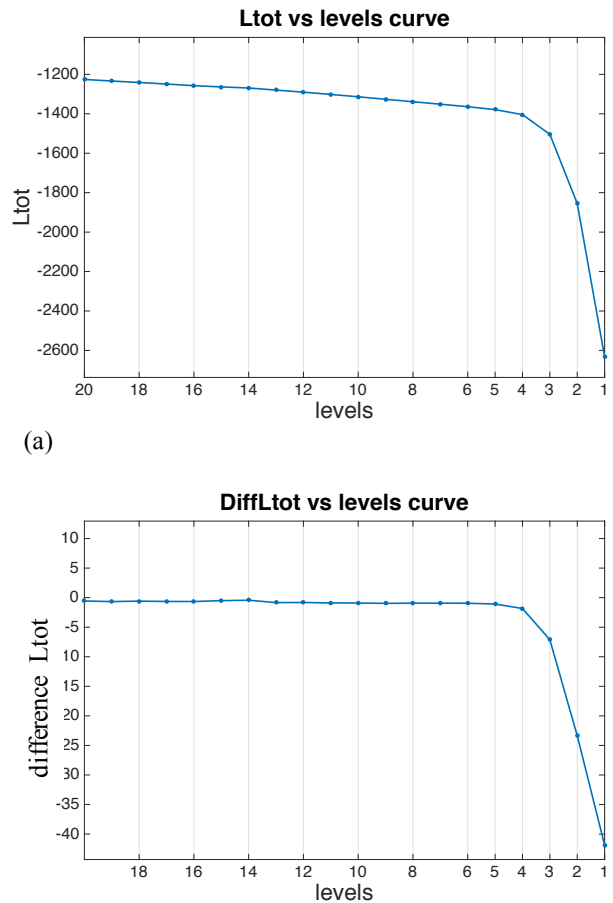
It is also crucial to estimate number of clusters  $c$  present in the given data. If some prior information (e.g. range of  $c$ ) about clusters is known then one can estimate  $c$  close to its true value. In some cases, this information is unknown, in that situation it is required to investigate all possible levels (in the hierarchical framework), so that the samples can be thoroughly investigated to estimate  $c$ . In this paper, we propose two curves to estimate  $c$ . The first curve is  $L_{tot}$  versus the levels curve and the second is the difference of  $L_{tot}$  ( $dL_{tot}$ ) versus the levels curve. As an illustration, we used a 4 cluster case (as in Suppl. 1, Fig. S1a). The  $L_{tot}$  and  $dL_{tot}$

curves are shown in Figures 7a and 7b. These curves are given between levels 1 and 20. At level  $l$  there are  $l$  clusters present. From Fig. 7a, the  $L_{tot}$  curve changes significantly between levels 1 and 4, and from  $l = 4$  onwards the rate of change in  $L_{tot}$  is low. Therefore, increasing the level further would not change the partitioning of data significantly. Thus,  $c$  can be estimated to be 4. However, if finer clusters (i.e., clusters having fewer samples) are required then one can consider having the level value for which  $L_{tot}$  is maximum.

We have also presented the  $dL_{tot}$  curve (Fig. 7b). At level  $l$ , the value of  $dL_{tot}$  can be given as

$$dL_{tot}(l) = \frac{L_{tot}(l+1) - L_{tot}(l)}{L_{tot}(l+1)} \times 100 \quad (33)$$

The multiplication by 100 in Equation 33 can be dropped (it is given here just for presentation purposes of the plot). The  $dL_{tot}$  curve basically measures the rate of change of  $L_{tot}$  curve. It can be seen from Fig. 7b that after level 4 ( $l > 4$ ) the curve is not changing much. Therefore, we can estimate  $c = 4$  using  $dL_{tot}$  curve.



**Figure 7:** a) likelihood  $L_{tot}$  plot; b)  $dL_{tot}$  curve

**C. Clustering on genomic data**

In this part, we analyze the HML method on a set of genomic data. As discussed before, there are two main concerns in clustering: 1) how many clusters are present; and, 2) what are the locations of these clusters? It is also interesting to identify

or remove some sub-population from the data in order to solve the issue of population stratification, because the existence of unbalanced population stratification between cases and controls may produce false positives and negatives in GWAS [60], [47] [40], [13]. Here we employ data from a collection of 7,001 individuals from the BioBank Japan (BBJ) project and 45 Japanese HapMap (JPT) samples [60]<sup>1</sup>. The total number of SNPs was 140,387, genotyped via the Perlegen platform. We also incorporated 45 Han Chinese HapMap (CHB) samples and merged these data using PLINK v1.9 (<https://www.cog-genomics.org/plink2>) on 140,367 common SNPs. Prior to PCA, we performed filtering using similar criteria as of that used by Yamaguchi et al. [60]. We removed SNPs with a call rate < 99%, a MAF < 0.01, and a Hardy-Weinberg equilibrium (HWE) exact test p-value > 10<sup>-6</sup>. Individuals with missing calls for > 5% of SNPs were also removed. After filtering, 6,998 BBJ, 44 JPT and 45 CHB samples sharing 117,758 SNPs remained. Consequently, the population consists of mainland Japanese (Hondo) having 6,891 samples, 45 CHB samples and 151 Okinawa samples, referred as the Ryukyu (RYU) cluster. Hondo consists of 628 Kyushu, 908 Kinki, 358 Tokai-Hokoriku, 3,975 Kanto-Koshinetsu, 466 Tohoku, 512 Hokkaido and 44 JPT samples. In this section, the goal is to identify RYU and CHB from Hondo so that the Hondo data can be explored for further analysis. We first performed PCA on the filtered data using the R package SNPRelate [64] to reduce the data dimensionality and conduct analysis on 5 dimensional data. Linkage disequilibrium (LD) pruning with a threshold of 0.2 was used to define a representative set of 32,090 SNPs for PCA.

There are three main clusters on this five dimensional data, namely, Hondo, RYU and CHB. We employed this data to first carry out clustering analysis to find correctly labelled samples of the Hondo, RYU and CHB clusters using various clustering methods; i.e., we evaluated the number of true positives. All the methods were executed to provide 3 clusters only. The true positive number and its corresponding percentage achieved by different methods are depicted in Table 4.

**Table 4:** Correctly clustered Hondo, RYU and CHB samples (true positive) using various clustering methods on BBJ and HapMap data.

Methods	Hondo (6891 samples)	RYU (151 samples)	CHB (45 samples)
K-means	5460 (79.2%)	93 (61.4%)	29 (65.0%)
SLink	6889 (99.9%)	2 (1.3%)	0 (0.0%)
CLink	6875 (99.8%)	2 (1.3%)	0 (0.0%)
ALink	6889 (99.9%)	2 (1.3%)	0 (0.0%)
WLink	6881 (99.9%)	2 (1.3%)	0 (0.0%)
MLink	6881 (99.9%)	2 (1.3%)	0 (0.0%)
HML	6655 (96.6%)	144 (95.4%)	45 (100.0%)

It can be observed from Table 4 that most of the methods achieve high true positives for the Hondo cluster, however, many fail to obtain similar performance for the RYU and CHB

<sup>1</sup> Here we did not employ European and African SNPs as they are quite well separated on leading two PCA components which will make clustering problem very easy. This analysis has shown on European SNPs by Novembre et al. [44].

clusters. One reason could be the imbalanced size of the subgroups. It can be noted that 6891 out of 7087 samples belong to the Hondo cluster; i.e., almost 97% of samples belong to the Hondo cluster leaving only 3% to the RYU and CHB clusters. This imbalance creates problems for many methods and consequently the majority of samples accumulated in one cluster and the methods failed to track other clusters objectively. Therefore, even the data appears to be separable (as in Fig. 8b), the detection of the RYU and CHB clusters are difficult due to the limited number of samples. Furthermore, in this imbalanced situation, the overall accuracy measure is not very meaningful (since all the samples grouped in only one cluster, i.e., the Hondo cluster, would show high overall clustering accuracy) and therefore we reported true positives for all the clusters. From the results, HML shows better detection for the RYU and CHB clusters. For CHB, the HML method clustered all the samples correctly.

In the previous analysis, we provided the number of cluster information to all the methods and obtained results. In the subsequent analysis, we do not provide this information and study the characteristics of the HML method. For this, we perform clustering on 5-dimensional BBJ and HapMap data and plot the transformed 5-dimensional data on 3-dimensional plane using the linear discriminant analysis (LDA) method [12], [54], [55], [56]. It can be observed from the  $L_{tot}$  plot (Fig. 8a) that after  $level = 3$  the  $L_{tot}$  curve does not change significantly. However, at  $level = 7$  it reaches its peak value. Therefore, one interpretation could be to consider 3 clusters as this would give the most significant partition of the data. This would provide the same results as obtained in Table 4. However, if some finer clusters (clusters with fewer samples) are required then maximum value of  $L_{tot}$  can be considered which would give 7 clusters. In Fig. 8b, we illustrated partition of data using 7 clusters. However, as mentioned, 3 clusters are dominant. The leftmost cluster (Cluster 1 in the figure) encompasses of Chinese samples, the center cluster (Cluster 2) is mostly Hondo samples and the rightmost cluster (Cluster 3) includes RYU samples. There are 6662 samples in Cluster 2 (Hondo). All CHB is clustered in Cluster 1 giving false negative (FN) error 0 (0.0%). Around 7 RYU samples are misclassified as the Hondo cluster, giving FN = 7 (4.6%). There are four other clusters as well (containing very few samples) which are not labelled in Fig. 8b. These are basically outliers representing noise. Thus after clustering, outliers can be removed and further analysis can be conducted on a particular region of interest. Therefore, HML can be applied to clustering problems to provide reasonable information about the cluster location and cluster numbers.

## VII. CONCLUSION

In this study, we proposed a hierarchical maximum likelihood (HML) method by considering the topologies of genomic data. It was shown that the HML method can perform clustering when the clusters appeared in an overlapping form. This method was also useful when the number of samples is lower than the data dimensionality. HML is free from initial parameter settings, and, it does not require computation of first

and second derivative of likelihood functions as required by many other maximum likelihood based methods. The HML method was tested both on artificial and real data and was able to deliver promising results over many existing clustering techniques. It was also illustrated that HML can estimate the number of clusters reasonably well. A Matlab package of our HML method is available from our webpage.

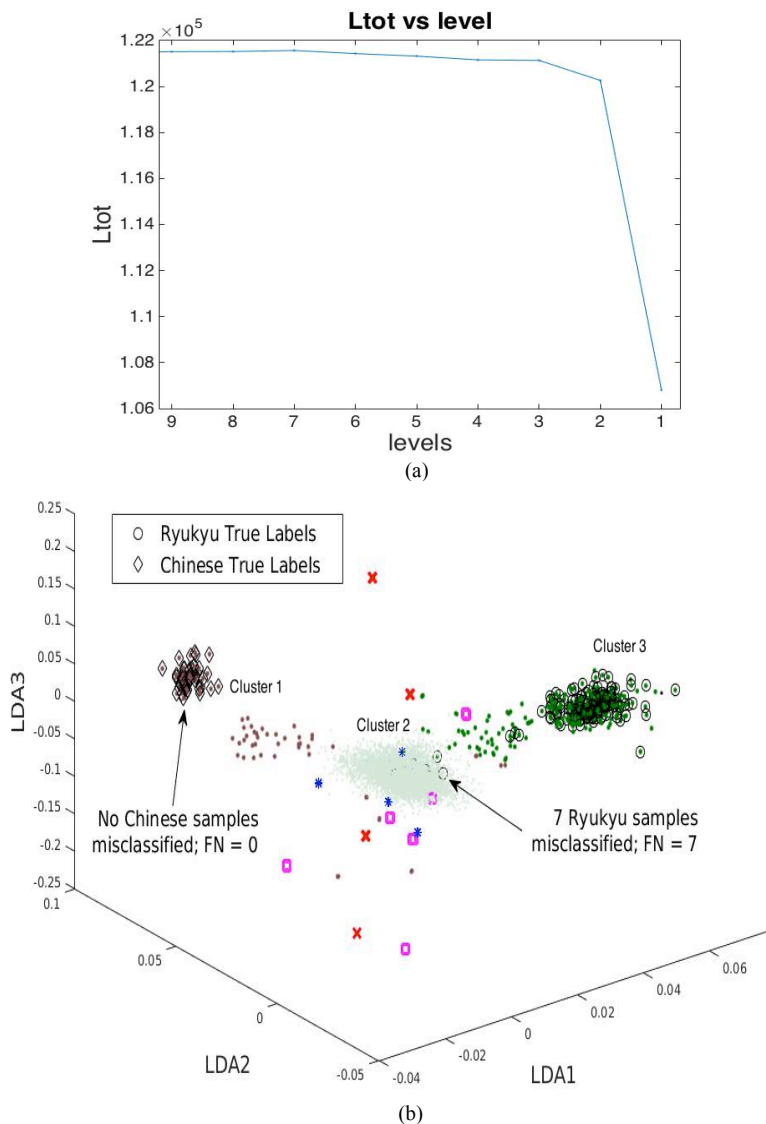


Figure 8: a)  $L_{tot}$  versus levels plot; b) Clustering by HML on 5-dimensional BBJ and HapMap data.

#### ACKNOWLEDGMENT

We thank the Editor and anonymous reviewers for providing constructive comments which greatly enhanced the presentation quality of the paper.

#### REFERENCES

[1] J. Adachi, M. Hasegawa, MOLPHY version 2.3: programs for molecular phylogenetics based on maximum likelihood, 1996.

[2] A. Ben-Hur et al., Support vector clustering, *J. Machine Learning Research*, vol. 2, pp. 125-137, 2001.

[3] E. Berndt et al., Estimation and Inference in Nonlinear Structural Models, *Annals of Economic and Social Measurement*, vol. 3, pp. 653-665, 1974.

[4] R. Castro, M. Coates, R. Nowak, Likelihood based hierarchical clustering, *IEEE Trans. Signal Process.*, vol. 42, pp. 2308-2321, 2004.

[5] C. Chen et al., Bayesian clustering algorithms ascertaining spatial population structure: a new computer program and a comparison study, *Molecular Ecology Notes*, vol. 7, pp. 747-756, 2007.

[6] J.-H. Chiang, P.-Y. Hao, A new kernel-based fuzzy clustering approach: support vector clustering with cell growing, *Fuzzy Systems, IEEE Transactions on Fuzzy Systems*, vol. 11, issue 4, pp. 518-527, 2003.

[7] W.C. Davidon, Variable metric method for minimization, *AEC Research and Development Report ANL-5990 Rev.* (1959).

[8] D. Defays, An efficient algorithm for a complete link method, *The Computer Journal (British Computer Society)*, vol. 20, no. 4, pp. 364-366, 1977.

[9] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society Series B*, vol. 39, no. 1, pp. 1-38, 1977.

[10] T. Denoeux, Maximum Likelihood Estimation from Uncertain Data in the Belief Function Framework, *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, issue 1, pp. 119-130, 2013.

[11] I.S. Dhillon, Y. Guan, J. Kogan, Iterative clustering of high dimensional text data augmented by local search, In *Proceedings of The 2002 IEEE International Conference on Data Mining*, 2002.

[12] R.O. Duda, P.E. Hart, D.G. Stork, *Pattern Classification*. Wiley, 2000.

[13] E. Elhamifar, R. Vidal, Sparse subspace clustering: Algorithms, theory, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.* Vol. 35, pp. 2765-2781, 2013.

[14] B.S. Everitt et al., *Cluster Analysis*, John Wiley & Sons, 5th edition, 2011.

[15] S. Farrell, C. Ludwig, Bayesian and maximum likelihood estimation of hierarchical response time models, *Psychon Bull Rev.*, vol. 15, no. 6, pp. 1209-1217, 2008.

[16] U.M. Fayyad, C.A. Reina, P.S. Bradley, Initialization of Iterative Refinement Clustering Algorithms, *Proceedings of the 4th International Conference on Knowledge Discovery & Data Mining (KDD98)*, R. Agrawal, P. Stolorz and G. Piatetsky-Shapiro (eds.), pp. 194-198, 1998.

[17] J. Felsenstein, G.A. Churchill, A hidden Markov model approach to variation among sites in rate of evolution, *Mol. Biol. Evol.*, vol. 13, no. 1, pp. 93-104, 1996.

[18] D. Fisher, Iterative optimization and simplification of hierarchical clustering, *Journal of Artificial Intelligence Research*, vol. 4, pp. 147-179, 1996.

[19] F. Fletcher, M.J.D. Powell, A rapidly convergent descent method for minimization, *Comput. J.* vol. 6, pp. 317-322, 1963.

[20] J. Goldberger, S. Roweis, Hierarchical clustering of a mixture model, *NIPS*, pp. 505-512, 2005.

[21] T.R. Golub et al., *Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring*, Science, vol. 286, pp. 531-537, 1999.

[22] T. Hastie, R. Tibshirani, Friedman, J, *The Elements of Statistical Learning* 2nd ed., New York, Springer, ISBN 0-387-84857-6, 2009.

[23] K.A. Heller, Z. Ghahramani, Bayesian hierarchical clustering, *Twenty-second International Conference on Machine Learning, ICML 2005*.

[24] S.-J., Horng et al., A novel intrusion detection system based on hierarchical clustering and support vector machines, *Expert Systems with Applications*, vol. 38, issue 1, pp. 306-313, 2011.

[25] A.K. Jain, Data clustering: 50 years beyond K-means, *Pattern Recognition Letters*, vol. 31, issue 8, pp. 651-666, 2010.

[26] A.K. Jain, M.N. Murty, Data clustering: a review, *ACM Computing Surveys*, vol. 31, no. 3, pp. 264-323, 1999.

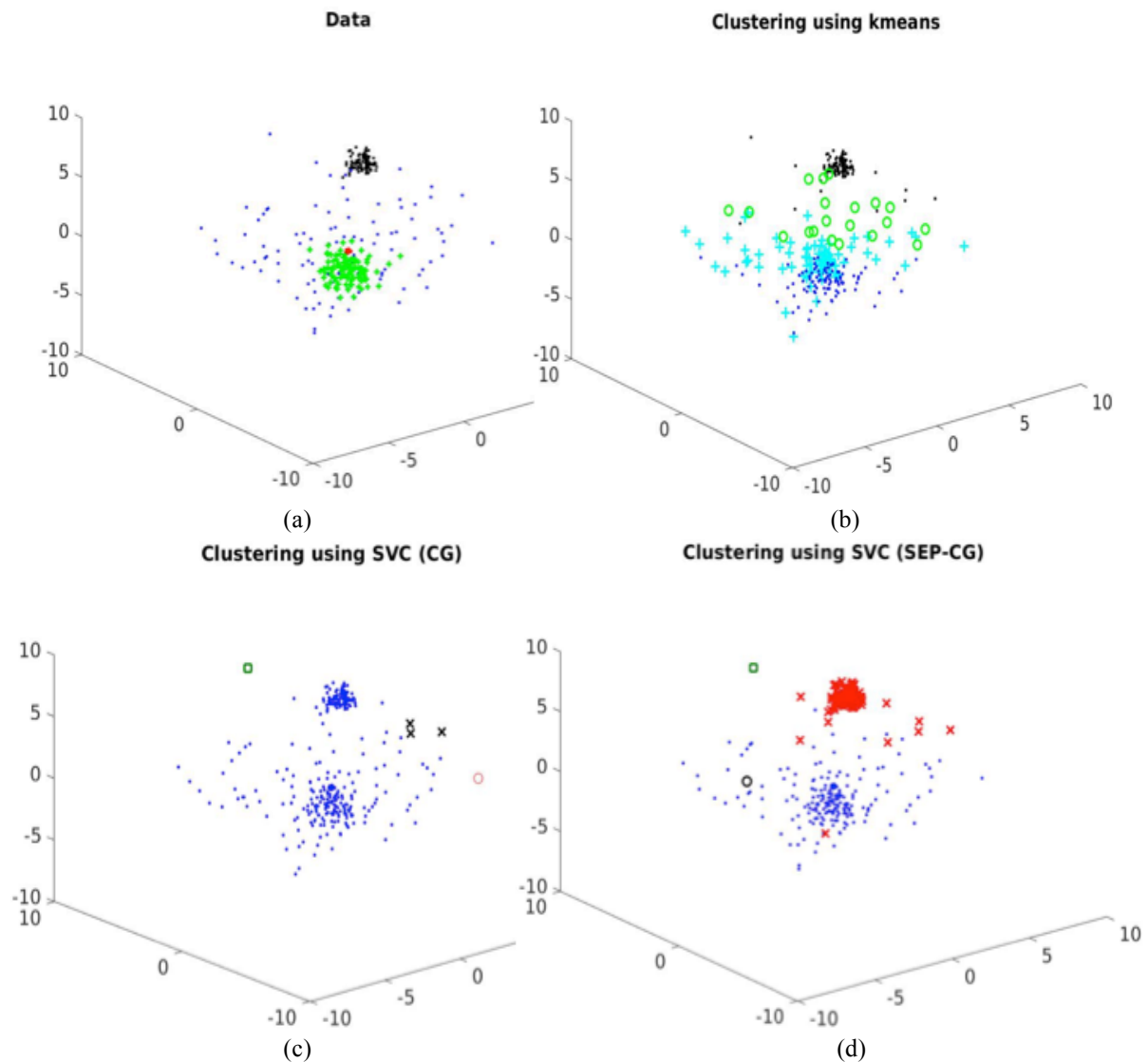
- [27] R.I. Jennrich, P.F. Sampson, Newton-Raphson and related algorithms for maximum likelihood variance component estimation, *Technometrics*, vol. 18, issue 1, pp. 11-17, 1976.
- [28] S. Jun, S.-S. Park, D.-S. Jang, Document clustering method using dimension reduction and support vector clustering to overcome sparseness, *Expert Systems with Applications*, vol. 41, issue 7, pp. 3204-3212, 2014.
- [29] L. Kaufman, P.J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis*, John Wiley & Sons, Inc, 2005.
- [30] H.-P. Kriegel, P. Kröger, A. Zimek, Clustering high-dimensional data: a survey on subspace clustering, pattern-based clustering, and correlation clustering, *ACM Transactions on Knowledge Discovery from Data*, vol. 3, no. 1, pp. 1-58, 2009.
- [31] E.K. Latch et al., Relative performance of Bayesian clustering software for inferring population substructure and individual assignment at low levels of population differentiation, *Conservation Genetics*, vol. 7, issue 2, pp. 295-302, 2006.
- [32] J. Lee, D. Lee, An improved cluster labeling method for support vector clustering, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 1-4, 2005.
- [33] J. Lee, D. Lee, Dynamic characterization of cluster structures for robust and inductive support vector clustering, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1869-1874, 2006.
- [34] P. Legendre, L. Legendre, *Numerical Ecology*, 2nd Edition, *Developments in Environmental Modelling 20*, Elsevier, Amsterdam, 1998.
- [35] J.S. Liu et al., Bayesian clustering with variable and transformation selections, *Bayesian Statistics*, vol. 7, pp. 249-275, 2003.
- [36] E.F. Lock, D.B. Dunson, Bayesian consensus clustering, *Bioinformatics*, doi: 10.1093/bioinformatics/btt425, 2013.
- [37] S. Long, *Regression Models for Categorical and Limited Dependent Variables*, London: Sage Publications, 1997.
- [38] G. McLachlan, D. Peel. *Finite Mixture Models*. Hoboken, NJ: John Wiley & Sons, Inc., 2000
- [39] L. McQuitty, Similarity analysis by reciprocal pairs for discrete and continuous data, *Educational and Psychological Measurement*, vol. 26, pp. 825-831, 1967.
- [40] B. Mirkin, *Clustering for Data Mining: A Data Recovery Approach*, Chapman and Hall, Boca Raton, Fla., 2005.
- [41] I. Miszta, Comparison of computing properties of derivative and derivative-free algorithms in variance-component estimation by REML, *Journal of Animal Breeding and Genetics*, vol. 111, issue 1-6, pp. 346-355, 1994.
- [42] S. Mo et al., Pattern discovery and cancer gene identification in integrated cancer genomic data, *PNAS*, vol. 110, no. 11, pp. 4245-4250, 2013.
- [43] S. Monti et al., Consensus clustering: a resampling-based method for class discovery and visualization of gene microarray data, *Machine Learning*, vol. 53, pp. 91-118, 2003.
- [44] J. Novembre et al., Genes mirror geography within Europe, *Nature*, pp. 98-101, 2008.
- [45] J.C. Pinheiro, D.M. Bates, *Mixed-effects models in S and S-Plus*, New York, NY, Springer, 2000.
- [46] J. Podani, *Multivariate data analysis in ecology and systematics*, *Ecological Computations Series (ECS)*: vol. 6, 1994.
- [47] M.M. Rahman, D.N. Davis, Fuzzy Unordered Rules Induction Algorithm Used as Missing Value Imputation Methods for K-Mean Clustering on Real Cardiovascular Data, *Proceedings of the World Congress on Engineering (WCE)*, vol 1, 2012.
- [48] M. Ramoni, P. Sebastiani, P. Cohen, Bayesian Clustering by Dynamics, *Machine Learning*, vol. 47, issue 1, pp. 91-121, 2002.
- [49] S.W. Raudenbush, A.S. Bryk, *Hierarchical linear models: Applications and data analysis methods*. 2<sup>nd</sup> ed.. Thousand Oaks, CA, Sage, 2002.
- [50] A. Sharma, K.K. Paliwal, Fast principal component analysis using fixed-point algorithm, *Pattern Recognition Letters*, vol. 28, issue 10, pp. 1151-1155, 2007.
- [51] A. Sharma, S. Imoto, S. Miyano, A top-r feature selection algorithm for microarray gene expression data, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 3, pp. 754-764, 2012.
- [52] A. Sharma, K.K. Paliwal, A gradient linear discriminant analysis for small sample sized problem, *Neural Processing Letters*, vol. 27, no. 1, pp. 17-24, 2008a.
- [53] A. Sharma, K.K. Paliwal, Cancer classification by gradient LDA technique using microarray gene expression data, *Data & Knowledge Engineering*, vol. 66, issue 2, pp. 338-347, 2008b.
- [54] A. Sharma, K.K. Paliwal, G.C. Onwubolu, Class-dependent PCA, MDC and LDA: A combined classifier for pattern classification, *Pattern Recognition*, vol. 39, no. 7, 1215-1229, 2006.
- [55] A. Sharma, K.K. Paliwal, Rotational linear discriminant analysis technique for dimensionality reduction, *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 10, pp. 1336-1347, 2008c.
- [56] A. Sharma, K.K. Paliwal, A new perspective to null linear discriminant analysis method and its fast implementation using random matrix multiplication with scatter matrices, *Pattern Recognition* vol., 45, no. 6, pp. 2205-2213, 2012.
- [57] R. Sibson, SLINK: an optimally efficient algorithm for the single-link cluster method, *The Computer Journal (British Computer Society)*, vol. 16, no. 1, pp. 30-34, 1973.
- [58] D. Singh et al., Gene Expression Correlates of Clinical Prostate Cancer Behavior, *Cancer Cell*, vol. 1, pp. 203-209, 2002.
- [59] R. Sokal, C. Michener, A statistical method for evaluating systematic relationships, *University of Kansas Science Bulletin*, vol. 38, pp. 1409-1438, 1958.
- [60] Y. Yamaguchi-Kabat et al., Japanese population structure, based on SNP genotypes from 7003 individuals compared to other ethnic groups: effect on population-based association studies, *The American Journal of Human Genetics*, vol. 83, pp. 445-456, 2008.
- [61] K. Wang et al., Prediction of piRNAs using transposon interaction and a support vector machine, *BMC Bioinformatics*, 15:419, 2014.
- [62] M.D. Wilkerson, D.N. Hayes, ConsensusClusterPlus: a class discovery tool with confidence assessments and item tracking, *Bioinformatics*, vol. 26, no. 12, pp. 1572-1573, 2010.
- [63] S. Vaithyanathan, B. Dom, Model-Based Hierarchical Clustering, In *Proc. 16th Conf. Uncertainty in Artificial Intelligence*, pp. 599-608, 2000.
- [64] X. Zheng et al., A High-performance Computing Toolset for Relatedness and Principal Component Analysis of SNP Data, *Bioinformatics*, vol. 28, no. 24, pp. 3326-3328, 2012.

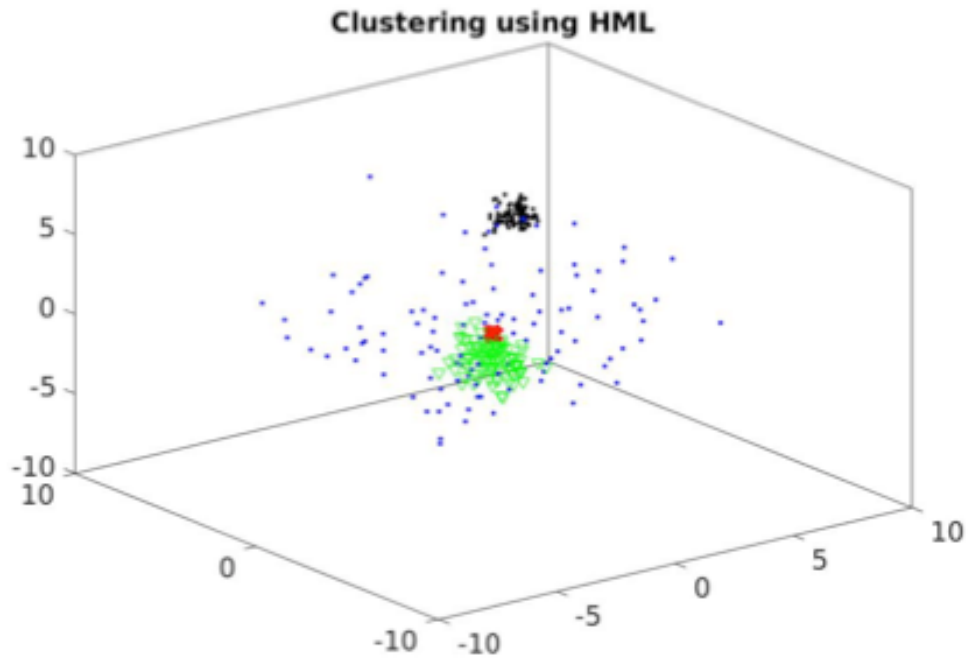
# Supplement 1

## An illustration using four clusters

A sample set with 400 samples, 3 dimensions and 4 clusters was generated using different normal distributions as shown in Fig. S1a. It can be seen that the clusters overlap each other. The goal is to track these clusters. We applied HML and compared with various other techniques such as k-means, SVC (CG) method [2] and SVC using SEP method [32]. All the methods are supposed to provide 4 clusters. It was quite straightforward to instruct the k-means and HML methods to provide the number of clusters expected at the output. However, the tuning process for SVC methods was not so friendly, as various support vector parameters had to be adjusted to get the desired number of clusters, which is a time consuming process. Fig. S1b shows the data clustered by the k-means method. It can be seen that k-means fails to track the different conformations of data. However, it can separate the samples by considering sample centroid information. The SVC (CG) method (Fig. S1c) also fails to track the clusters. The majority of the samples belong to one cluster and the remaining few samples belong to the three different clusters.

The SVC (SEP) method was able to track the two main clusters, however, was not able to find the other two clusters (Fig. S1d). The HML method is basically able to track all the four clusters in their appropriate locations (Fig. S1e).





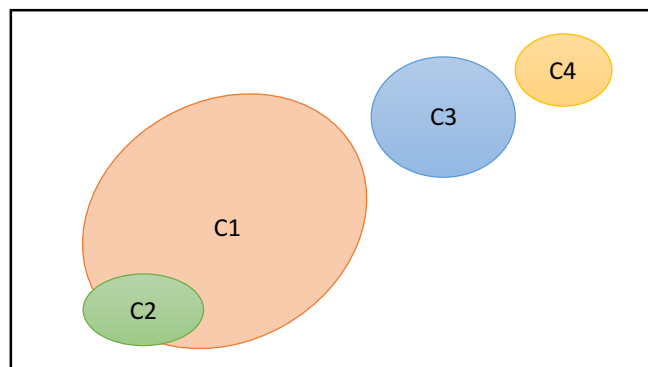
(e)

**Figure S1:** An illustration using 400 samples, 4 clusters and 3 dimensions. a) Normal data conformation; b) k-means clustering method; c) SVC (CG) method; d) SVC (SEP) method; e) HML method.

## Supplement 2

### Clustering on synthetic data-2

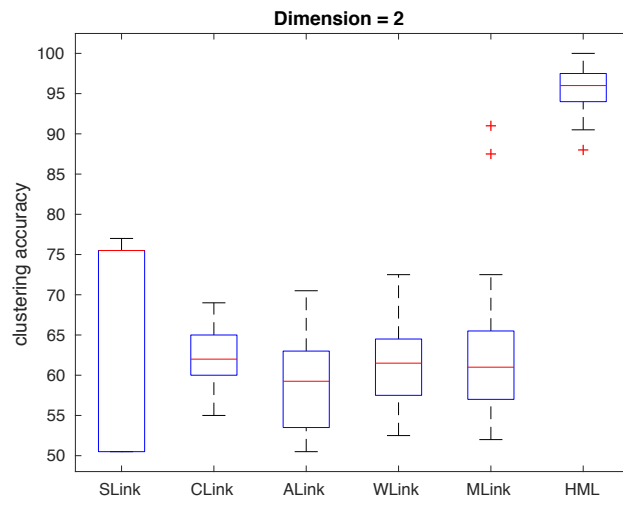
In this section, we performed clustering analysis on synthetic data. For this, we generated Gaussian data of  $d$ -dimensions and 200 samples (having 4 clusters) as depicted in Fig. S2a. Each cluster has 50 samples. The dimensionality of the sample set varied from 2 to 100. We generated the data 50 times for a particular dimension  $d$  and computed clustering accuracy of various hierarchical methods such as SLink, CLink, ALink, WLink, MLink and HML. The clustering accuracies of these methods for dimensions  $d = 2, 3, 4, 5, 10, 15, 20, 25$  and 100 are depicted in Figures S2b-j.



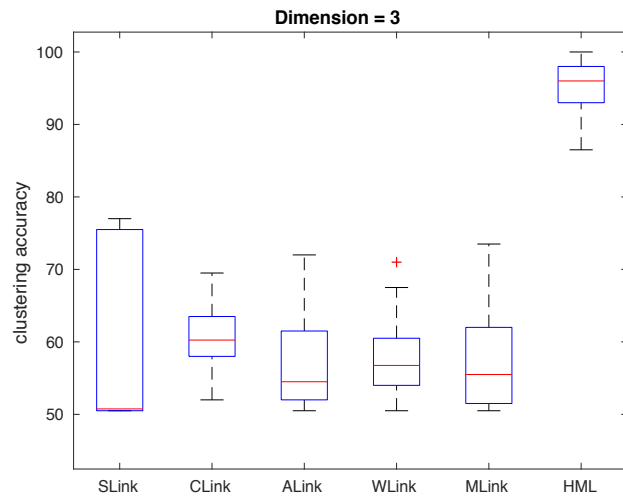
**Figure S2a:** Synthetic data having 4 clusters.

A few observations can be made from Figures S2b-j. As the dimensionality of data increased, the clustering accuracies of all the hierarchical methods gradually decreased. The performance of all the methods are relatively better in lower dimensional space. The SLink hierarchical method showed the worst performance compared to other methods employed in this work. The methods like CLink, ALink, MLink and HML are able to cluster in the higher dimensional space as well. HML is showing comparatively better results.

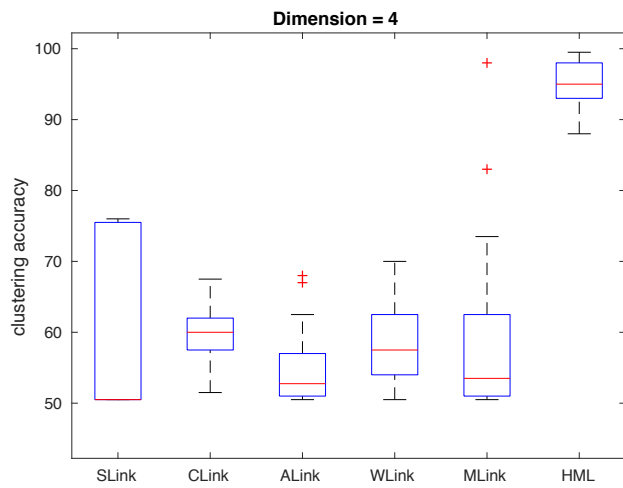
(b)



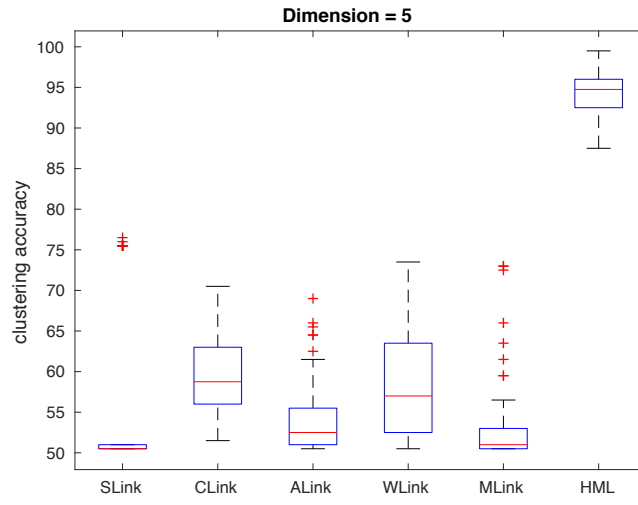
(c)



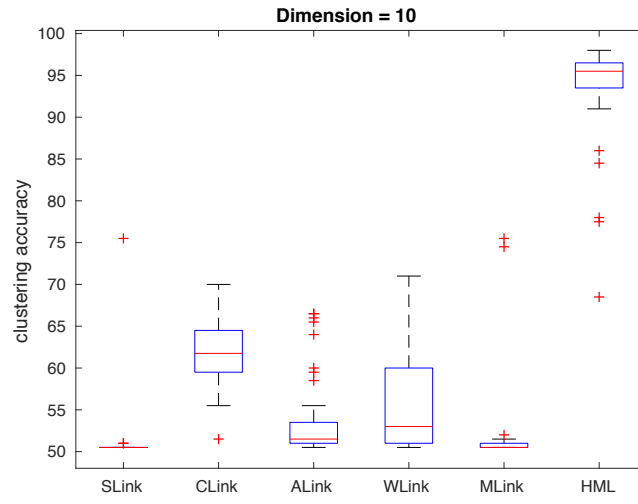
(d)



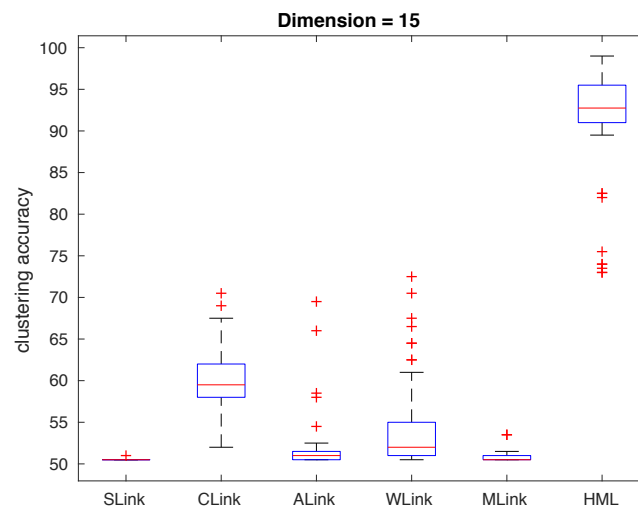
(e)

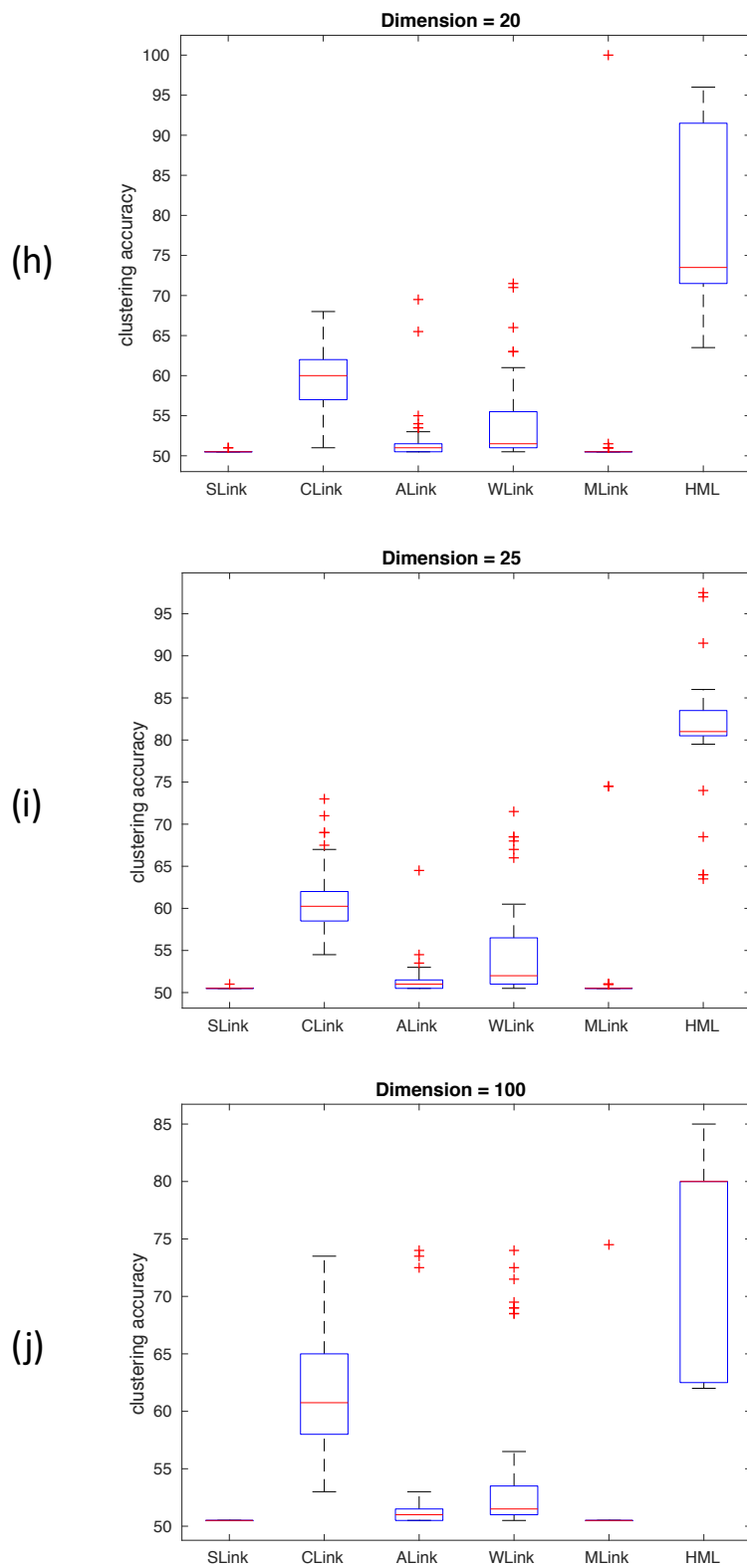


(f)



(g)





**Figure S2b-j:** Clustering accuracies of hierarchical methods over 50 attempts.