A Hierarchical Clustering algorithm based on Silhouette Index for cancer subtype discovery from genomic data

N. Nidheesh¹ • K. A. Abdul Nazeer² • P. M. Ameer¹

Received: 7 October 2018 / Accepted: 22 November 2019
© Springer-Verlag London Ltd., part of Springer Nature 2019

Abstract

Identifying potential novel subtypes of cancers from genomic data requires techniques to estimate the number of natural clusters in the data. Determining the number of natural clusters in a dataset has been a challenging problem in Machine Learning. Employing an internal cluster validity index such as Silhouette Index together with a clustering algorithm has been a widely used technique for estimating the number of natural clusters, which has limitations. We propose a Hierarchical Agglomerative Clustering algorithm which automatically estimates the numbers of natural clusters and gives the associated clustering solutions along with dendrograms for visualizing the clustering structure. The algorithm has a Silhouette Index-based criterion for selecting the pair of clusters to merge, in the process of building the clustering hierarchy. The proposed algorithm could find decent estimates for the number of natural clusters, and the associated clustering solutions when applied to a collection of cancer gene expression datasets and general datasets. The proposed method showed better overall performance when compared to that of a set of prominent methods for estimating the number of natural clusters, which are used for cancer subtype discovery from genomic data. The proposed method is deterministic. It can be a better alternative to contemporary approaches for identifying potential novel subtypes of cancers from genomic data.

Keywords

Cluster analysis • Hierarchical Clustering • Silhouette Index • Cluster number estimation • Cancer subtype discovery • Gene expression data • Consensus Clustering

1 Introduction

Cluster analysis is a fundamental data mining technique that is used in a wide variety of areas for data analysis, including the Biomedical domain. Cluster analysis aims to find natural clusters of data points in a given dataset. By ‘natural clusters,’ we mean groups of data points in feature space such that the data points in a group are close to one another, and there is significant separation between the groups. A clustering algorithm or a clustering method is a tool that performs cluster analysis. There are many clustering algorithms proposed in the literature. Hierarchical Clustering is a classical clustering method and has been found to be used extensively in the Biomedical literature. Cluster analysis of genomic data such as gene expression data has been an area where Hierarchical Clustering has been quite popular. The reasons for such popularity, according to [10], are its ease of use, and the availability of implementations as part of many data analysis software. Classical clustering algorithms such as K-Means [23] require the user to give the number of clusters as input. Finding an appropriate value for the number of clusters is not an easy task in general. Tasks demanding exploratory data analysis, such as discovering potential novel subtypes of cancers from molecular data of cancer tissues, require methods to estimate the exact number of natural clusters in the data. Generally, it is hard to assess the number of natural clusters in a dataset.

Electronic supplementary material

The online version of this article (https://doi.org/10.1007/s00521-019-04636-5) contains supplementary material, which is available to authorized users.

N. Nidheesh
nidheesh.n@gmail.com

¹ Department of Electronics and Communication Engineering, National Institute of Technology Calicut, Calicut, Kerala 673601, India
² Department of Computer Science and Engineering, National Institute of Technology Calicut, Calicut, Kerala 673601, India

Published online: 30 November 2019

Springer
Cancer is caused by faulty cellular mechanisms that lead to an out-of-control growth of tissues. There are several types of cancer. Each cancer type has multiple subtypes. A subtype may further have subtypes. Knowledge of the exact subtype of cancer for a patient helps in deciding the most appropriate treatment for the patient. Cancer subtype discovery is the process of finding out previously unknown subtypes of cancers [12]. Clustering algorithms have been instrumental in the discovery of cancer subtypes from genomic data [3, 4, 6, 12, 20, 24]. Clustering methods developed with a facility to automatically estimate the number of natural clusters (such as Consensus Clustering [28, 43]), and frameworks or pipelines employing a clustering algorithm together with an internal cluster validity index have been used extensively for the task. Hierarchical Clustering with a manual inspection of dendrograms for deciphering the number of natural clusters has been another popular choice.

1.1 Limitations in estimating the true number of clusters

The task of estimating the correct number of natural clusters for a dataset, in general, is hard. A reason for this limitation is that, in most cases, one is dealing with samples from a population and one needs to estimate the number of actual clusters in the population. The quality of estimation heavily depends on how well the sample at hand resembles the underlying population in the distribution of the data points. For instance, in the case of cancer gene expression data, we have the expression profile of a limited number of tissue samples and the distribution of the underlying population is not known. Furthermore, the aptness of the estimated number of clusters may be context dependent. Consider Fig. 1. There are four groups of data points named G1, G2, G3, and G4. Consider the following three estimations of the number of clusters from the dataset—(1) Four clusters: each group forming a cluster. (2) Three clusters: Groups G2 and G3 together forming a cluster and G1 and G4 forming their respective clusters. (3) Two clusters: Groups G2, G3, and G4 together forming a cluster and G1 forming its own cluster. None of these groupings can be treated superior to others without having any additional information about the data. The goodness of an estimated number of clusters and the associated clustering solution is dependent on the context and the dataset.

The limitations discussed above are a few among the inherent limitations that one faces when trying to estimate the number of natural clusters from the data, and which are unavoidable. The Biomedical research community has been using many methods to assess the most appropriate number of clusters from molecular data, from within these limitations. A popular technique followed by the Biomedical research community is to use an internal cluster validity index in conjunction with a clustering algorithm (a ‘clustering algorithm—internal cluster validity index pipeline’).

1.2 Silhouette Index: a cluster validity index

Cluster validity indexes are measures used to assess the quality of clustering solutions produced by clustering algorithms. They are broadly classified into two: Internal and External [39]. External cluster validity indexes require the true class information of each data point along with the clustering result for the computation. Adjusted Rand Index (ARI) [16] is an example. In contrast, internal cluster validity indexes do not need true class information. Instead, in general, they use just the clustering result and the similarity or dissimilarity measurements among the data points, to compute the index.

Several internal cluster validity indexes have been proposed in the literature. Silhouette Index [34] is one of the most widely used cluster validity indexes in Biomedical data analysis. Details of computation of Silhouette Index are given in Sect. 3. Silhouette Index is defined for a data point with respect to a clustering solution. The value of the index fall in the interval \([-1, +1]\). The closer the Silhouette Index of a data point to +1, the better clustered the data point is. For a clustering solution, the more the average of Silhouette Indexes of all data points closer to +1, the better the clustering solution is [34].
A recent study on Biomedical clustering algorithms [44] considers Silhouette Index as one of the best internal cluster validity indexes and suggests the use of Silhouette Index in the cluster analysis of Biomedical data where no ground truth class information of data points is available. Silhouette Index is found to be used for two purposes in the literature related to biomolecular data processing. One is as a measure to check the validity of the obtained clustering solution. The other is to use along with a clustering algorithm to make Silhouette Index-based pipelines to estimate the number of natural clusters in a dataset (such as in discovering cancer subtypes from omics data).

1.3 Silhouette Index-based pipelines for cluster number estimation

Silhouette Index has been widely used to make ‘clustering algorithm—internal cluster validity index’ pipelines. Such a pipeline is used to infer the optimal number of clusters in a dataset in the following way. Let K-Means [23] be the clustering algorithm used to make the pipeline. K-Means is run multiple times with the value of the parameter k (which is the required number of clusters) varied from a minimum to a maximum (for example, 2 to 20). For each of the resulting clustering solution, the Average Silhouette Index is computed. The value of k which leads to a clustering solution which has the maximum Average Silhouette Index is considered as an appropriate number of natural clusters in the dataset. The popular clustering algorithms, which are frequently used along with Silhouette Index to infer the number of natural clusters, are K-Means [23], Partitioning Around Medoids (PAM) [19], and Spectral Clustering (SC) [30, 37].

A point worth noting is that K-Means, SC, and PAM are non-deterministic clustering algorithms since they have steps involving randomness. These algorithms may produce different results for the same dataset during different executions. Therefore, the ‘clustering algorithm—Average Silhouette Index’ pipelines built using these algorithms are also non-deterministic.

Finding potential novel subtypes of cancers from multi-omics data is a key area of current Biomedical research. Silhouette Index has been in widespread use in estimating the number of natural clusters in the biomolecular data.

Cai and Li [4] introduced a multi-view clustering approach (CMC) and its enhanced version (ECMC) for subtype identification from heterogeneous cancer datasets. The authors employed both Spectral Clustering and K-Means together with Silhouette Index to identify novel cancer subtypes. Wu et al. [45] presented an integrative clustering method for multi-omics data of cancer tissues, named ‘LRA Cluster’, which employs ‘K-Means-Average Silhouette Index pipeline’ to determine the number of clusters (subtypes). Shi et al. [38] presented a pattern fusion approach (PFA) for identifying cancer subtypes from heterogeneous omics data. The authors used K-Means to cluster the resulting data of the PFA process and considered the number of clusters indicated by Average Silhouette Index as the number of subtypes. Chalise and Fridley [8] proposed ‘intNMF’ for integrative clustering of multi-omics data of cancer tissues, to identify novel subtypes. The authors employed Average Silhouette Index as one of the methods to get an optimum number of clusters. Li et al. [21] proposed ‘Integrated Consensus Clustering’ for cancer subtype discovery by integrated analysis of mRNA, miRNA, and IncRNA expression data. Silhouette Index is one of the methods they employed to find out the optimal number of clusters (subtypes).

1.4 Our proposal

For the clustering algorithm—Average Silhouette Index pipelines, the clustering solutions are constructed based on a criterion particular to the clustering algorithm, whereas the evaluation and selection of a particular solution are based on a different criterion—the criterion followed by Silhouette Index. Hence, high-quality results are obtained only for those datasets which suit the criteria of both the clustering algorithm and Silhouette Index.

Here, we propose a clustering algorithm which uses a criterion based on Silhouette Index for computing clustering solutions—we propose a Hierarchical Agglomerative Clustering algorithm which can find the most likely numbers of natural clusters and the associated clustering solutions, in addition to providing dendrograms for visualizing the clustering structure. The algorithm (named SilHAC) uses an objective function based on Average Silhouette Index for selecting the pair of clusters to merge, in the iterative merging process for making the clustering hierarchy. Salient features of the proposed method are:

- The method directly gives appropriate numbers of natural clusters and the associated clustering solutions.
- Unlike the other methods which are being used for estimating the number of clusters, the proposed method produces dendrograms. A user can inspect the dendrogram to find whether there can be meaningful subclusters possible for the obtained clusters.
- The method is deterministic—different executions of the algorithm for the same dataset is guaranteed to produce the same results since the method has no steps involving randomness.
2 Related work

Hierarchical Agglomerative Clustering (HAC) is a classical clustering algorithm proposed decades ago. The algorithm works as follows. Initially, the algorithm considers that there are as many singleton clusters as there are data points in the input dataset, with each cluster constituted by a unique data point. The algorithm merges the most appropriate pair of clusters in a merging step to build the clustering solution for the next level in the clustering hierarchy. The iterative merging process is continued until there is only one cluster left. The appropriateness of a pair of clusters for merging is decided by a criterion particular to the variant of HAC, and HAC has many variants which differ from one another in the merge criterion employed.

Ward [42] has proposed a framework for HAC. According to the framework, one can employ an objective function that reflects the chosen criterion for selecting the pair of clusters for merging. The pair which leads to the maximal value for the objective function (or minimal value if the objective function is a function to be minimized) is selected for merging at each step. Ward suggested using the Total Within-cluster Sum of Squared Errors (TWSSE) as an objective function.

The variant of HAC with TWSSE as the objective function (to minimize) is popularly known in the literature both as Hierarchical Agglomerative Clustering (Ward's method) and as Hierarchical Agglomerative Clustering (Ward's minimum variance method). Hereafter, we use HAC (Ward's method) to refer to Hierarchical Agglomerative Clustering (Ward's method).

The proposed method (SilHAC) differs from HAC (Ward's method) in the following ways. Firstly, SilHAC straightforwardly gives the most likely number or numbers of the natural clusters in the dataset (and also the associated clustering solutions), whereas HAC (Ward's method) does not. Secondly, SilHAC employs an objective function based on Average Silhouette Index (to maximize) for selecting clusters to merge, whereas HAC (Ward's method) uses the TWSSE as the objective function (to minimize). Thirdly, SilHAC has a hybrid mode, in which it starts the iterative merging of clusters from the group of clusters produced by HAC (Ward's method), whereas HAC (Ward's method) and the other variants of HAC always start the iterative merging process from the group of singleton clusters constituted by each of the data points.

3 Materials and methods

3.1 The proposed method

The basic idea of the proposed method is to use an Average Silhouette Index-based objective function in Ward’s framework for Hierarchical Clustering [42], so that the levels of the resulting clustering hierarchy for which the value of the objective function is the highest indicate both the numbers of natural clusters and the associated clustering solutions.

The relevant definitions of terms and functions, used in the algorithms for the proposed method, are given below.

3.1.1 Terminology

3.1.2 Clustering vector

The proposed method (SilHAC) outputs a set of one or more clustering vectors. A clustering vector is an integer vector of cluster labels which indicates the cluster to which each data point belongs. The size of a clustering vector is equal to the number of data points in the input dataset. For instance, if we have a dataset of five data points of which the first three data points belong to cluster number ‘1’ and the rest belong to cluster number ‘2’, then the corresponding clustering vector has the following structure:

\[
\begin{bmatrix}
1 & 1 & 1 & 2 & 2
\end{bmatrix}
\]

3.1.3 Dissimilarity

The dissimilarity between two data points refers to the difference between the data points. Any suitable distance measure can be used as a measure of dissimilarity in cluster analysis. A commonly used distance measure is the Euclidean distance. The Euclidean distance between two \(d\)-dimensional data points \(x\) and \(y\) is computed as follows:

\[
\delta(x, y) = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2},
\]

where \(x_i\) and \(y_i\) denote the values of the \(i\)th component of the Euclidean vectors for the data points \(x\) and \(y\), respectively.

A dissimilarity matrix is a symmetric matrix where the element at the \(i\)th row and the \(j\)th column is the measure of dissimilarity between the \(i\)th and the \(j\)th data points. Algorithm 1 computes a dissimilarity matrix for an input dataset. For the subsequent discussions, we use the terms ‘dissimilarity’ and ‘Euclidean distance’ interchangeably.
Algorithm 1: ComputeDissimilarityMatrix

| Input: | A set of data points: $D$ |
|---|---|
| Output: | A matrix with pair-wise dissimilarity measures (Euclidean distances) of data points in $D$. |
| 1. $M \leftarrow$ zerosSquareMatrix($|D|$) |
| 2. for $i \leftarrow 1$ to $(|D| - 1)$ do |
| 3. for $j \leftarrow (i + 1)$ to $|D|$ do |
| 4. $M[i, j] \leftarrow d(D[i], D[j])$ |
| 5. $M[j, i] \leftarrow M[i, j]$ |
| 6. end |
| 7. end |
| 8. return $M$ |

3.1.4 Silhouette Index

Silhouette Index [34] is one of the most widely used internal cluster validity indexes. Let $D$ be a dataset. Let $X = \{C_1, C_2, \ldots, C_k\}$ be a partition (a clustering solution) of $D$. Let $i$ be a data point in $D$ and let $C(i)$ denote the cluster to which $i$ belongs. Then, Silhouette Index of the data point $i$ for the clustering solution $X$ (which is denoted by $s(i, X)$) is computed as follows.

$$s(i, X) = \frac{b(i, X) - a(i, X)}{\max\{b(i, X), a(i, X)\}},$$

where

$$a(i, X) = \frac{1}{|C(i)| - 1} \left( \sum_{j \in C(i), j \neq i} \delta(i, j) \right),$$

and

$$b(i, X) = \min_{C \in (X \setminus C(i))} \left\{ \frac{1}{|C|} \sum_{j \in C} \delta(i, j) \right\}.$$  \(\text{(4)}\)

The function silhouetteIndex (invoked at Step 8 of Algorithm 5) computes the Silhouette Index for each data point and returns the result as a vector of size $|D|$. The Average Silhouette Index (ASI) of the data points of a dataset for a clustering solution gives a picture of how good the clustering solution is [34]. The more the ASI closer to one, the better the clustering solution is. The ASI for a clustering solution $X$ of a dataset $D$ is given by,

$$ASI(D, X) = \frac{1}{|D|} \left( \sum_{i \in D} s(i, X) \right).$$  \(\text{(5)}\)

3.1.5 Extended average Silhouette Index (ExtASI): the objective function

SilHAC is a Hierarchical Clustering algorithm which follows Ward’s framework for Hierarchical Clustering [42]. The objective function, employed by SilHAC for selecting the clusters to merge, is an extended version of ASI (named ExtASI). ExtASI of a clustering solution is the weighted sum of ASI and the fraction of data points having a positive Silhouette Index.

Let $D$ be a dataset and let $X$ be a clustering solution for $D$. Let $ASI(D, X)$ be the Average Silhouette Index for the clustering solution $X$. Let $PSICount(D, X)$ be defined as

$$PSICount(D, X) = \sum_{i \in D} \Psi(s(i, X)), \quad \text{(6)}$$

where

$$\Psi(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise.} \end{cases} \quad \text{(7)}$$

Then, ExtASI is computed as follows.

$$ExtASI(D, X) \propto \frac{\alpha ASI(D, X) + (1 - \alpha)}{|D|} \left( PSICount(D, X) \right), \quad \text{(8)}$$

where $\alpha$ is a user-supplied parameter called the weighting parameter.

The values that $\alpha$ takes are required to be close to one in practice, so as to keep the value of ExtASI not much deviated from the value of ASI. Smaller values of $\alpha$ can severely affect the ability of the objective function (ExtASI) in rightly assessing the quality of the clustering solutions, and in correctly estimating the number of clusters. A suggested interval for the values that $\alpha$ can take, based on empirical studies, is $[0.85, 1]$.

ExtASI is used as the objective function instead of ASI, for SilHAC, because of the following reason. Average Silhouette Index (ASI) for a clustering solution is not always proportional to the number of data points which have a positive Silhouette Index. If a data point has a positive Silhouette Index, the data point can be considered as correctly clustered (or included in the most appropriate cluster). It is entirely possible that from among two different clustering solutions for a dataset $D$, say $X_1$ and $X_2$, $X_1$ has the highest ASI despite having more number of data points with negative Silhouette Index (a few data points with high Silhouette Index can compensate for the data points with negative Silhouette Indexes and still yield the highest ASI).
The above property of the ASI is a limitation for it being used as an objective function in Ward’s framework for Hierarchical Clustering. Hierarchical Clustering algorithms follow a greedy approach. They select a pair of clusters which gives the optimal value for the objective function (a pair which leads to the locally optimal solution) to merge. However, the pair selected in this way may not be the best to merge, in a global perspective, and once a pair is merged, the merging cannot be undone at a later stage in the process of building the hierarchy. This is a limitation faced by all the Hierarchical Clustering algorithms [33] and which can lead to suboptimal results. When ASI is used as an objective function (to maximize), without caring about the number of data points with positive Silhouette Index, clusters which would lead to a clustering solution with more number of incorrectly classified data points may get merged than the ones which would lead to a clustering solution having less number of such data points. Such a lousy merging can have a cumulative effect on the subsequent levels of the clustering hierarchy. Substandard results will be produced especially when such a merging happens in the early stages of building the hierarchy.

3.1.6 Precision parameter (or $\varepsilon$)

ASI is a real number in the interval $[-1, 1]$. Considering all the decimal places of the ASI values is not necessary in practice, for the task of cluster number estimation. The decimal places beyond the hundredth are not much significant in general, for the task. Rousseeuw, who proposed Silhouette Index, rounded off the ASI values to two decimal places, in the examples (of cluster number estimation) given in [34]. However, rounding can introduce errors in the process of estimating the number of clusters. For example, consider a run of SilHAC where the highest ExtASI is 0.724 and the second highest is 0.715. Here, both values get rounded off to 0.72 despite having a difference of 0.009 between them and hence, both the numbers of clusters corresponding to the values will be considered as the probable values of the number of clusters.

We employ a parameter named precision parameter($\varepsilon$) for SilHAC to avoid the above issue due to rounding. SilHAC outputs all the clustering vectors corresponding to the clustering solutions whose ExtASI score differs from the maximum ExtASI score at most by $\varepsilon$. For example, consider a run of SilHAC where $\varepsilon$ is set a value 0.005, and where the highest ExtASI obtained is 0.655 and the second highest ExtASI is 0.654. Here, both the clustering vectors corresponding to the ExtASI values will be the members of the set of clustering vectors output by SilHAC since the difference of the second highest ExtASI from the highest ExtASI is less than $\varepsilon$.

3.1.7 mergeClusters: a function to merge two clusters

SilHAC, being a Hierarchical Clustering algorithm, needs to merge clusters, in the process of building the clustering hierarchy. Each level of the hierarchy corresponds to a clustering solution. We represent a clustering solution corresponding to a level of the hierarchy using a clustering vector. Algorithm 2 presents a routine which takes a clustering vector and two cluster labels as inputs and modifies the clustering vector in such a way that it represents a clustering solution in which the two clusters—whose labels are given—are merged.

**Algorithm 2:** mergeClusters: An algorithm to merge two clusters.

```
Input: Labels of the two clusters to be merged: $c_1$ and $c_2$, a clustering vector representing the current clustering solution: CV.
Output: A modified version of CV representing a clustering solution where the clusters $c_1$ and $c_2$ are merged.
1. $L_{max} \leftarrow \max(c_1, c_2)$
2. $L_{min} \leftarrow \min(c_1, c_2)$
3. for $i \leftarrow 1$ to length(CV) do
   4. if $CV[i] = L_{max}$ then
      5. $CV[i] \leftarrow L_{min}$
   6. else if $CV[i] > L_{max}$ then
      7. $CV[i] \leftarrow CV[i] - 1$
   8. end
4. end
9. return CV
```

3.1.8 The algorithms

The essence of the proposed method is given as Algorithms 3, 4, and 5. The algorithms present only the core logic of
the proposed method (SilHAC). The part which deals with constructing the **dendrogram** has been omitted for brevity and ease of presentation.

**Algorithm 3: SilHACwrapper**: A wrapper algorithm for SilHAC.

**Input**: A set of data points: $D$, an integer threshold representing the maximum number of initial clusters SilHAC deals with: $\tau$, a weighting parameter: $\alpha$, a precision parameter: $\varepsilon$.

**Output**: A set of one or more clustering vectors representing clustering solutions which have an appropriate number of clusters.

```plaintext
1 M ← ComputeDissimilarityMatrix(D)
2 InitCV ← \{1, 2,...,|D|\}
3 if |D| > $\tau$ then
4     InitCV ← WardHAC(M, $\tau$) // HAC(Ward’s method)
5 end
6 CVSet ← SilHAC(M, InitCV, $\alpha$, $\varepsilon$)
7 return CVSet
```

SilHAC has two modes of operation—**pure mode** and **hybrid mode**. For large datasets, the algorithm can be run in the **hybrid mode** in order to minimize the execution time. We used HAC (Ward’s Method) as a component clustering algorithm to make the hybrid. In the **hybrid mode**, the agglomerative merging of clusters starts from a set of clusters obtained using the component clustering algorithm.

The mode in which SilHAC executes for a dataset is determined by the value of a user-supplied threshold $\tau$. If the size of the input dataset is less than or equal to $\tau$, the algorithm runs in the **pure mode**; otherwise, it runs in the **hybrid mode**, wherein HAC (Ward’s Method) is invoked first, to get $\tau$ number of clusters. SilHAC begins its hierarchical merging process on these $\tau$ clusters. A wrapper algorithm for SilHAC has been designed (Algorithm 3) which decides whether SilHAC needs to be run in the **pure mode** or the **hybrid mode**.

**Algorithm 4: SilHAC (The essence of the proposed method).**

**Input**: A dissimilarity matrix: $M$, a clustering vector representing an initial clustering solution: $CV$, a weighting parameter: $\alpha$, a precision parameter: $\varepsilon$

**Output**: A set of one or more clustering vectors representing clustering solutions which have an appropriate number of clusters.

```plaintext
1 MaxExtASI ← -1
2 BestCVSet ← $\Phi$
3 CVEstASIPairs ← $\Phi$
4 NC ← countUniqueValues(CV) // The number of clusters
5 while NC > 2 do
6     \{NextLevelCV, ExtASI\} ← getNextLevelCV(M, CV, $\alpha$)
7     CVEstASIPairs ← CVEstASIPairs $\cup$ \{\{NextLevelCV, ExtASI\}\}
8     if ExtASI > MaxExtASI then
9         MaxExtASI ← ExtASI
10    end
11    CV ← NextLevelCV
12    NC ← NC – 1
13 end
14 foreach (CV’, ExtASI’) $\in$ CVEstASIPairs do
15     if (MaxExtASI – ExtASI’) $\leq$ $\varepsilon$ then
16         BestCVSet ← BestCVSet $\cup$ \{CV’\}
17     end
18 end
19 return BestCVSet
```

The proposed method works as follows. The wrapper algorithm (SilHACwrapper) takes a dataset $D$, an integer threshold $\tau$, the weighting parameter $\alpha$, and a precision parameter $\varepsilon$ as inputs. At first, a dissimilarity matrix for the data points is prepared since such a matrix is necessary for computing Silhouette Index. Then, the algorithm makes an initial clustering vector (initCV) depending on the value of the $\tau$ parameter. HAC (Ward’s method) is employed to get $\tau$ number of clusters whenever the dataset has a higher number of data points than the value of $\tau$. The associated clustering vector is set as the initial clustering vector for SilHAC. Whenever the dataset has less than or equal to $\tau$ data points, the initial clustering vector represents the scenario where each data point forms a singleton cluster. Then, SilHAC (Algorithm 4) is invoked to build the clustering hierarchy starting from the clustering scenario represented by the initial clustering vector.
In essence, the algorithm SilHAC (Algorithm 4) works as follows. It takes every pair of clusters and computes the ExtASI score for the clustering scenario that would result when the pair is merged. It also computes the count of data points having a positive Silhouette Index (PSICount) concerning the clustering scenario. If there is only a single pair of clusters which leads to the maximum ExtASI, then that pair is (actually) merged to reduce the number of clusters by one. If there are multiple pairs which lead to the highest ExtASI, then one among the pairs, which leads to a clustering solution with the maximum PSICount, is selected for (actual) merging. Such a merging produces the clustering solution for the next (higher) level in the hierarchy. The ExtASI of the resulting clustering solution and the corresponding clustering vector are recorded after each (actual) merging. Such a merging produces the clustering results of multiple runs of a Consensus Clustering algorithm-Average Silhouette Index pipelines which are being used for cancer subtype discovery from omics data.

An ensemble clustering algorithm known as Consensus Clustering [28] is another method which is quite popular among Biomedical researchers who deal with discovering cancer subtypes from genomic data. Consensus Clustering employs data item sub-sampling and feature sub-sampling and provides a mechanism to find a consensus across the results of multiple runs of a base clustering algorithm on variants of the input dataset obtained by applying random sub-sampling [28]. Quite a large number of recent studies aimed at cancer subtype discovery from multi-omics data [5, 6, 9, 27, 40, 49] have employed Consensus Clustering, particularly the implementation provided by the R package called ConsensusClusterPlus [43].

We selected Consensus Clustering also for performance comparison with SilHAC. Three variants of Consensus Clustering formed by using HAC (Average Linkage), HAC (Complete Linkage), and K-Means, respectively, as the base clustering algorithms, were used for the performance comparison.

A collection of ten cancer gene expression datasets (with the subtype information known for the tissue samples) was
used to compare the capability of the methods in accurately identifying the subtypes. A comparison was also made with respect to the performances on a collection of five ‘general datasets’ which are widely used in the literature for testing the performance of clustering algorithms.

4.1 Datasets

The selected collection of cancer gene expression datasets consists of five datasets from the famous repository of gene expression data namely Gene Expression Omnibus (GEO) [2] of National Center for Biotechnology Information (NCBI), four datasets introduced by the authors of Consensus Clustering for testing the performance of Consensus Clustering [28], and one dataset (RNA-Seq PanCancer gene expression data) from UCI Machine Learning Repository [22].

The properties of the gene expression datasets are summarized in Table 1. Details of the datasets with additional information such as cancer type/subtype information are provided in Table S1 and Table S2 of ‘Supplementary Materials’ file. Datasets have been assigned short names for convenience in the representation of their names in the bar charts. The datasets with short names GEO1, GEO2, GEO3, GEO4, and GEO5 are the feature-reduced (gene/probe-reduced, as described in [31]) versions of the NCBI GEO datasets with accession numbers GSE51082 (GPL-97), GSE57162 [13], GSE66354 [14], GSE85217 [7], and GSE94601 [18], respectively. The datasets Leukemia (LEUK), Novartis Multi-tissue (NOVA), St. Jude Leukemia (STJL), and Lung Cancer (LUNG) are the datasets introduced by the authors of Consensus Clustering [28] for testing the performance of their method. The datasets, since their introduction, are used as benchmark datasets in the literature dealing with cancer molecular data classification [26, 46–48]. RNA-Seq PANCAN dataset is a TCGA Pan-cancer dataset which consists of the gene expression data for tissue samples of five cancer types namely breast cancer (BRCA), colon cancer (COAD), kidney clear cell carcinoma (KIRC), lung adenocarcinoma (LUAD), and prostate cancer (PRAD).

A summary of the properties of the general datasets is given in Table 2. Olive oil dataset was obtained from [1]. All the other general datasets were obtained from the UCI Machine Learning repository [22].

4.2 Setup for the comparative study

The three Clustering algorithm—Average Silhouette Index pipelines considered for the comparative study are K-

| Dataset                        | Short name | No. of samples | No. of genes or probes | No. of types or subtypes |
|--------------------------------|------------|----------------|------------------------|--------------------------|
| GSE51082 Leukemia              | GEO1       | 140            | 2000                   | 6                        |
| GSE57162 RCC                   | GEO2       | 191            | 2000                   | 4                        |
| GSE66354 Brain Tumors          | GEO3       | 136            | 2000                   | 6                        |
| GSE85217 Medulloblastoma       | GEO4       | 763            | 2000                   | 4                        |
| GSE94601 Lung Carcinoma        | GEO5       | 137            | 2000                   | 5                        |
| Leukemia                       | LEUK       | 38             | 999                    | 3                        |
| Novartis Multi-Tissue          | NOVA       | 103            | 1000                   | 4                        |
| St. Jude Leukemia              | STJL       | 248            | 985                    | 6                        |
| Lung Cancer                    | LUNG       | 197            | 1000                   | 4                        |
| RNA-Seq PANCAN                 | PANC       | 801            | 20,531                 | 5                        |

RCC stands for Renal Cell Carcinoma. Additional information, such as the names of cancer or subtype and the count of tissue samples per cancer or subtype, are given in Table S1 and Table S2 of the ‘Supplementary Materials’ file. Short names are assigned for the datasets, for the convenience in presenting the results of performance comparison in bar charts.

| Dataset                        | Short name | No. of classes | No. of features | Size  |
|--------------------------------|------------|----------------|----------------|-------|
| Iris                           | Iris       | 3              | 4              | 150   |
| Wine                           | Wine       | 3              | 13             | 178   |
| Olive oil                      | Oliv       | 9              | 8              | 572   |
| Dermatology                    | Derm       | 6              | 34             | 358   |
| Wisconsin diagnostic breast cancer | WDBC     | 2              | 30             | 569   |

Short names are assigned for the datasets, for the convenience in presenting the results of performance comparison in bar charts.
means-Average Silhouette Index pipeline (hereafter referred to as KM-SI pipeline), Spectral Clustering—Average Silhouette Index pipeline (hereafter referred to as SC-SI pipeline), and PAM—Average Silhouette Index pipeline (hereafter referred to as PAM-SI pipeline). We used the implementation provided by Kernlab package [17] for Spectral Clustering. For PAM, we used the implementation provided by the cluster package [25]. For K-Means, we used the R’s native implementation of Lloyd’s algorithm [23]. The silhouette method provided by the cluster package was used to compute the Silhouette Index.

For Consensus Clustering, the three variants selected for the performance comparison with SilHAC are the ones with HAC (Average Linkage), HAC (Complete Linkage), and K-Means, respectively, as the base clustering algorithms (in the subsequent text, these variants are referred to as CCHC(a), CCHC(c), and CCKM, respectively). The implementation of Consensus Clustering provided by the R package ConsensusClusterPlus [43] was used for our experiments. PAC method introduced in [35] was employed to automatically select the most appropriate number of natural clusters from the results of Consensus Clustering.

For the experiments, values for the parameters of SilHAC were set as follows. The value of \( \tau \) (the integer threshold) was set to 125 for all the experiments. The value 125 was chosen for \( \tau \) because of the following reasons. Setting a value which is too small will lead to high influence of the component clustering algorithm [i.e., \( \text{HAC (Ward’s method)} \)] on the clustering results and cluster number estimation. On the other hand, setting a value which is too large will increase the time taken for execution of SilHAC significantly, while dealing with large datasets. With \( \tau \) set to the value 125, SilHAC could produce the results in reasonable lengths of time in all the experiments which we have carried out. Another parameter of SilHAC is the weighting parameter \( \alpha \). As the value of \( \alpha \) is required to be close to one for keeping \( \text{ExtASI} \) not much deviated from \( \text{ASI} \), \( \alpha \) was set to 0.9, for all the experiments. Also, the precision parameter \( \epsilon \) was set to 0.005 for all the experiments.

For the KM-SI pipeline, K-Means was run 100 times for each value of \( k \) in the range 2–20. For each value of \( k \), out of the 100 clustering solutions, the one which has the minimum Total Within-cluster Sum of Squared Errors was selected. Then, Average Silhouette Index was computed for the selected 19 clustering solutions. Similarly to the experiments with SilHAC, a precision parameter was employed (with its value set to 0.005) for selecting the final set of clustering solutions. All the clustering vectors representing the clustering solutions having an \( \text{ASI} \), which differs from the maximum \( \text{ASI} \) at most by 0.005, were considered as representatives of the most likely clustering solutions.

Similarly, for the SC-SI pipeline, Spectral clustering was set to run 20 times (limited to 20, as Spectral Clustering requires a significant amount of time to run when compared to K-Means) for each value of \( k \) in the range 2–20. For each \( k \), the clustering solution which has the maximum \( \text{ASI} \) was selected. From the 19 selected clustering vectors, the final set of clustering vectors representing the most likely clustering solutions were selected based on a precision parameter (with its value set to 0.005) the same way as was done for KM-SI pipeline.

PAM (Partitioning Around Medoids) was run 20 times for each value of \( k \) in the range 2–20. \( \text{ASI} \) was computed for each clustering solution. The same procedure which was carried out for KM-SI and SC-SI pipelines was carried out to get the final set of clustering vectors (with the value of the precision parameter set to 0.005).

The seed for the random number generator was set to one for all the experiments which involved the pipelines. This was done since the pipelines are non-deterministic, and hence, for the repeatability of the analyses.

For the Consensus Clustering variants, the parameters were set as follows. For all the variants, the respective base clustering algorithm was made to run 100 times, the maximum number of clusters expected was set to 15, and the seed for random number generator was assigned the value one (for the repeatability of the analyses). Euclidean distance was the distance measure used, similar to the other experiments. The value of the parameter \( \text{innerLinkage} \) was set to “average” for CCHC(a), and to “complete” for CCHC(c). The rest of the parameters were assigned their default values set in ConsensusClusterPlus package [43].

The quality of the obtained clusters of the different methods was compared based on the values obtained for a prominent ‘external cluster validity index’ known as the Adjusted Rand Index (ARI) [16]. The maximum achievable value for ARI is one and is achieved when there is a perfect match between the clustering solution and the ground truth class information. There is no lower bound defined for ARI. The value of ARI can be negative when the quality of the clustering solution is terrible. We used ClusterR package [29] to compute ARI.

All the datasets were subjected to min-max normalization before they were used for the experiments.

4.3 Dendograms

Unlike the other methods for cluster number estimation, SilHAC provides dendograms for visualizing the clustering structure of the data. Two sample dendograms, produced by SilHAC in the pure mode and the hybrid mode, respectively, are given in Figs. 2 and 3. The complete set of dendograms for all the datasets used for the
experiments and the respective class-cluster contingency tables are given in the ‘Supplementary Materials’ file.

4.4 Performance on cancer gene expression datasets

Bar charts presenting ARI scores obtained by SilHAC and the three clustering algorithm-Silhouette Index pipelines are given in Fig. 4. Besides, bar charts showing ARI scores obtained by SilHAC and the three Consensus Clustering variants are given in Fig. 5.

The ARI scores and the numbers of clusters estimated for the datasets are given in Tables 3 and 4. It can be observed that the numbers of clusters identified by SilHAC are either equal to the numbers of true classes or are more closer to the numbers of true classes than those computed by the other methods for most of the datasets. Moreover, when the overall performance is considered, the quality of the clustering solutions produced (indicated by ARI) by SilHAC are superior to those of the other methods.

4.5 Performance on general datasets

A comparison of the quality of the clustering results of the methods for the general datasets is presented in Figs. 6 and 7. The numbers of clusters estimated and the ARI scores are given in Tables 3 and 4. It can be seen that the proposed method is better than the other methods considering the overall performance.

5 Discussion

SilHAC has many advantages when compared to the widely used ‘clustering algorithm-cluster validity index’ pipelines, in addition to providing comparatively better results. Unlike the pipelines, SilHAC produces dendrograms. A user can inspect the dendrogram and see if any of the obtained clusters can further be divided into meaningful sub-clusters. If there exists a promising cluster with a good clustering substructure, the user can run SilHAC for the
produced by them, for the cancer gene expression datasets listed in Table 1. CCHC(a) and CCHC(c) stand for Consensus Clustering having HAC (Average Linkage) and HAC (Complete Linkage), respectively, as the base clustering methods. CCKM stands for Consensus Clustering with K-Means as the base clustering method.

Another advantage of SilHAC over the pipelines and Consensus Clustering is that being a deterministic method, SilHAC gives reliable results. As discussed earlier, for the pipelines, K-Means, PAM, and Spectral Clustering need to be run a large number of times for each value of \( k \) (the number of clusters) in a user-specified range. For Consensus Clustering, the base clustering algorithm needs to be run a large number of times. Setting a value for the number of times to run, so that the best result is obtained, is a hard task. Moreover, repeating the experiment may not always yield the same results since the clustering methods are non-deterministic. Selecting an appropriate range for \( k \), for the pipelines as well as Consensus Clustering, is hard too, since there is always a risk of the true number of clusters falling outside the selected range.

A vast majority of cancer genomic datasets available at the prominent repositories such as NCBI Gene Expression Omnibus\(^1\) have less than 1000 samples per dataset. Since

---

\(^1\) [https://www.ncbi.nlm.nih.gov/geo/](https://www.ncbi.nlm.nih.gov/geo/)
### Table 3
Results of the comparisons of the performance of SilHAC (the proposed method) with three prominent clustering algorithm—Average Silhouette Index pipelines for cluster number estimation

| Dataset | Ground truth No. of classes | SilHAC NC | ARI | KM-SI NC | ARI | SC-SI NC | ARI | PAM-SI NC | ARI |
|---------|-----------------------------|-----------|-----|----------|-----|----------|-----|----------|-----|
| GEO1    | 6                           | 4         | 0.570 | 4         | 0.570 | 4         | 0.570 | 4         | 0.570 |
| GEO2    | 4                           | 2         | 0.240 | 2         | 0.202 | 2         | 0.202 | 2         | 0.191 |
| GEO3    | 6                           | 5         | 0.860 | 4         | 0.782 | 4         | 0.782 | 4         | 0.773 |
| GEO4    | 4                           | 3         | 0.665 | 3         | 0.663 | 3         | 0.663 | 3         | 0.657 |
| GEO5    | 5                           | 4         | 0.774 | 3         | 0.780 | 3         | 0.813 | 3         | 0.702 |
| LEUK    | 3                           | 3         | 1     | 3         | 1     | 3         | 1     | 4         | 0.600 |
| NOVA    | 4                           | 4         | 0.973 | 4         | 0.946 | 4         | 0.946 | 4         | 0.973 |
| STIL    | 6                           | 6         | 0.949 | 6         | 0.939 | 5         | 0.910 | 6         | 0.956 |
| LUNG    | 4                           | 2         | 0.397 | 2         | 0.397 | 2         | 0.397 | 2         | 0.397 |
| PAN C   | 5                           | 5         | 0.968 | 5         | 0.981 | 5         | 0.959 | 6         | 0.881 |
| Iris    | 3                           | 2         | 0.568 | 2         | 0.568 | 2         | 0.568 | 2         | 0.568 |
| Wine    | 3                           | 3         | 0.854 | 3         | 0.869 | 3         | 0.917 | 2         | 0.365 |
| Oliv    | 9                           | 5         | 0.794 | 5         | 0.793 | 5         | 0.798 | 5         | 0.777 |
| Derm    | 6                           | 3         | 0.571 | 3         | 0.579 | 3         | 0.579 | 3         | 0.550 |
| WDBC    | 2                           | 2         | 0.773 | 2         | 0.730 | 2         | 0.005 | 2         | 0.755 |

ARI: Adjusted Rand Index, NC: Number of clusters. KM-SI: K-Means-Average Silhouette Index pipeline, SC-SI: Spectral clustering—Average Silhouette Index pipeline, PAM-SI: Partitioning Around Medoids—Average Silhouette Index pipeline. For all the methods, when more than one number is estimated as the number of clusters, the number of clusters and the ARI score corresponding to the clustering solution with the maximum ARI only are presented here. The complete set of cluster numbers estimated by the methods and the corresponding ARI scores are given in Section S3 of ‘Supplementary Materials’ file.

### Table 4
Results of the comparisons of the performance of SilHAC (the proposed method) with three variants of Consensus Clustering—a prominent method for cluster number estimation

| Dataset | Ground truth No. of classes | SilHAC NC | ARI | CCHC(a) NC | ARI | CCHC(c) NC | ARI | CCKM NC | ARI |
|---------|-----------------------------|-----------|-----|-------------|-----|-------------|-----|----------|-----|
| GEO1    | 6                           | 4         | 0.570 | 5           | 0.776 | 4           | 0.570 | 10        | 0.656 |
| GEO2    | 4                           | 2         | 0.240 | 2           | 0.191 | 2           | 0.138 | 4         | 0.741 |
| GEO3    | 6                           | 5         | 0.860 | 14          | 0.832 | 6           | 0.896 | 2         | 0.465 |
| GEO4    | 4                           | 3         | 0.665 | 3           | 0.666 | 2           | 0.578 | 15        | 0.336 |
| GEO5    | 5                           | 4         | 0.774 | 2           | 0.032 | 2           | 0.058 | 2         | 0.599 |
| LEUK    | 3                           | 3         | 1     | 15          | 0.301 | 15          | 0.229 | 15        | 0.270 |
| NOVA    | 4                           | 4         | 0.973 | 2           | 0.001 | 5           | 0.934 | 15        | 0.403 |
| STIL    | 6                           | 6         | 0.949 | 4           | 0.198 | 6           | 0.901 | 5         | 0.926 |
| LUNG    | 4                           | 2         | 0.397 | 2           | 0.397 | 2           | 0.397 | 4         | 0.874 |
| PAN C   | 5                           | 5         | 0.968 | 2           | 0.000 | 15          | 0.855 | 6         | 0.876 |
| Iris    | 3                           | 2         | 0.568 | 2           | 0.568 | 15          | 0.303 | 2         | 0.568 |
| Wine    | 3                           | 3         | 0.854 | 8           | 0.792 | 9           | 0.681 | 3         | 0.915 |
| Oliv    | 9                           | 5         | 0.794 | 11          | 0.809 | 14          | 0.735 | 4         | 0.716 |
| Derm    | 6                           | 3         | 0.571 | 2           | 0.210 | 4           | 0.681 | 3         | 0.579 |
| WDBC    | 2                           | 2         | 0.773 | 2           | 0.005 | 2           | 0.331 | 2         | 0.730 |

ARI: Adjusted Rand Index, NC: Number of clusters. CCHC(a) and CCHC(c) stand for Consensus Clustering with Hierarchical Clustering as the base clustering algorithm with linkage criteria Average Linkage and Complete Linkage, respectively. CCKM stands for Consensus Clustering with K-Means as the base clustering algorithm. For all the methods, when more than one number is estimated as the number of clusters, the number of clusters and the ARI score corresponding to the clustering solution with the maximum ARI only are presented here. The complete set of cluster numbers estimated by the methods and the corresponding ARI scores are given in Section S3 of ‘Supplementary Materials’ file.
Fig. 6 A comparison of the performance of SilHAC (the proposed method) with three prominent clustering algorithms—Average Silhouette Index pipelines for cluster number estimation, in terms of the quality of the clustering solutions produced by them (indicated by Adjusted Rand Index), for the set of general datasets listed in Table 2. KM-SI: K-Means-Average Silhouette Index pipeline, SC-SI: Spectral Clustering—Average Silhouette Index pipeline, PAM-SI: Partitioning Around Medoids—Average Silhouette Index pipeline.

Fig. 7 A comparison of the performance of SilHAC (the proposed method) with three variants of Consensus Clustering, in terms of the quality of the clustering solutions produced by them (indicated by Adjusted Rand Index), for the set of general datasets listed in Table 2. CCHC(a) and CCHC(c) stand for Consensus Clustering having HAC (Average Linkage) and HAC (Complete Linkage), respectively, as the base clustering methods. CCKM stands for Consensus Clustering with K-Means as the base clustering method.

the size of the datasets of the target domain is small and limited, and the accuracy and the ability to identify subtypes are more important than the time for execution in the critical task of cancer subtype discovery, it is not much relevant analyzing the time consumed for execution by the methods or the complexity of the methods which are employed for cancer subtype discovery from genomic data. However, for the ease of use in practice, one may prefer methods which run faster, from among a group of methods producing high-quality results. Here, we present the time taken for execution by all the methods used for performance comparison when the input was the largest and high-dimensional dataset considered for the study—the RNA-Seq PANCAN dataset. The time consumed by each method for execution is given in Table 5. Clearly, SilHAC has a significant edge in the execution time over the pipelines. Also, SilHAC is much faster than the variant of Consensus Clustering with K-Means as the base clustering algorithm (CCKM). The need for distance computations for a large number of times for the data points in a high-dimensional space makes the pipelines and the Consensus Clustering variant slow. For SilHAC, once the pair-wise dissimilarity matrix is computed, there is no further need to compute distances.

Hierarchical Clustering algorithms have the advantages of ease of use and facility to visualize the clustering hierarchy by means of dendrograms. SilHAC too has these advantages, being a HAC. Moreover, there are only a few parameters for SilHAC, and values can be assigned for them easily.

SilHAC has a limitation that, when we have a priori information about the true number of natural clusters \((k)\), we may not always get the correct clusters by cutting the dendrogram at level \(k\), just as we do with the dendrograms of the prominent Hierarchical Clustering algorithms. The reason for this limitation is that SilHAC may, in some cases, merge two natural clusters before it assembles the data points of another natural cluster. An example of this scenario is the case of GSE51082 Leukemia (GEO1) dataset. The dendrogram produced by SilHAC for this dataset is given in Fig. 3. In fact, the cluster which is second from the left is formed by merging three natural clusters, with two of the natural clusters merged before the first cluster (the leftmost one) was assembled.

A method to follow in such a situation is the following. Consider the clustering solution corresponding to the maximum number of clusters estimated by SilHAC. Inspect the dendrogram to see if there is further scope for finding sub-clusters for the obtained clusters. Select such clusters, if any, and apply SilHAC on each of them to get more clusters. The procedure may be repeated until we get a total of \(k\) clusters.

The same procedure can be applied when a user expects a higher number of clusters than what is estimated by SilHAC for a dataset. The procedure, in such cases, may be repeated as long as the quality of the resulting clustering solution (indicated by the ExtASJ score) remains above acceptable levels. For an illustration of the procedure, let us consider the GEO1 dataset whose dendrogram, produced
by SilHAC, is presented in Fig. 3. The cluster located second from the left, in the dendrogram, has a visibly good clustering substructure. The cluster has 76 data points. When this cluster is separately given as input to SilHAC, it estimated two (sub)clusters. The corresponding dendrogram is given in Fig. 8.

Now again the second cluster in the new dendrogram has a visibly proper clustering substructure. When the process is repeated with the second cluster, which has 59 data points, once again SilHAC identified two (sub)clusters. The corresponding dendrogram is given in Fig. 9.

The GEO1 dataset originally has six classes (i.e., subtypes). The initial attempt to cluster, with SilHAC, resulted in four clusters. In the further analysis done above, we got one of those clusters divided into a total of three clusters. Now, considering the initial clustering solution and the subdivisions, we have identified six clusters in total, with SilHAC. This clustering solution achieves an ARI score of

| Method                                         | Time (in minutes) |
|------------------------------------------------|-------------------|
| Consensus Clustering (HC-Average Linkage)     | 2.50              |
| Consensus Clustering (HC-Complete Linkage)    | 2.50              |
| SilHAC (the proposed method)                   | 22.14             |
| Spectral Clustering—Average Silhouette Index pipeline | 226.20          |
| Consensus Clustering (K-Means)                 | 247.80            |
| PAM—Average Silhouette Index pipeline          | 514.20            |
| K-Means—Average Silhouette Index pipeline      | 727.80            |

Fig. 8 Finding further clusters for the GEO1 dataset. The second cluster (second from the left) of the dendrogram shown in Fig. 3 was taken for further clustering as it has a proper sub-clustering structure as seen in the dendrogram in Fig. 3. SilHAC was applied to it to get this dendrogram. The cluster had 76 data points. SilHAC estimated two (sub)clusters.

Fig. 9 Finding further clusters for the GEO1 dataset. The second cluster (second from the left, and having 59 data points) of the dendrogram shown in Fig. 8 was taken for further clustering as it has a good sub-clustering structure as seen in the dendrogram in Fig. 8. SilHAC was applied to it to get this dendrogram. SilHAC estimated two (sub)clusters.
1.0 (i.e., 100% accuracy). The initial clustering solution (for the entire dataset) had an $\text{ExtASI}$ score of 0.555. When $\text{SilHAC}$ is executed for the cluster with 76 data points, the resulted $\text{ExtASI}$ score was 0.459. When applied $\text{SilHAC}$ for the (sub)cluster with 56 data points, the obtained $\text{ExtASI}$ score was 0.395 which is significantly high to be considered acceptable.

A limitation of $\text{Silhouette Index}$ (and hence of $\text{ASI}$ and $\text{ExtASI}$) is that it tends to prefer a lower number of clusters especially when there is a significant variance in the inter-cluster distances. For instance, consider the dataset given in Fig. 1. Clusters G2 and G3 are closer to each other, and their second nearest neighbor cluster G4 is far away. Consider the calculations of $\text{Silhouette Index}$ of the data points in G2 and G3 (according to Formula 2). The $b$-values of the data points in G2 are computed with respect to their distances to the data points in G3 and vice versa. However, when G2 and G3 together are treated as one cluster, the $b$-values of the data points are computed with respect to their distances to the data points in G4 which are quite far. There is an increase in the $a$-values of data points of both G2 and G3 when we treat G2 and G3 are merged into one cluster. However, the increase in the $b$-values is relatively much higher. Because of this, the $\text{Silhouette Index}$ of data points in G2 and G3 are higher in the merged scenario than those of the data points when G2 and G3 are treated as separate clusters. The same effect can result when we consider the groups G2, G3, and G4 together as one cluster and G1 as another cluster. So, the methods for assessing the number of natural clusters based on $\text{Average Silhouette Index}$ or its derivative $\text{ExtASI}$ may find two as the most appropriate number of clusters for the dataset, even though there are four natural groups. This limitation makes it necessary to analyze the obtained clusters of $\text{SilHAC}$ for the scope for further clustering them. Rectifying this issue of $\text{Silhouette Index}$ is a future work worth investigating.

One of the reasons why we employed $\text{HAC}$ (Ward’s method) for making the hybrid mode-variant of $\text{SilHAC}$ is that HAC (Ward’s method) is a deterministic algorithm. Therefore, employing $\text{HAC}$ (Ward’s method) makes $\text{SilHAC}$ deterministic in its hybrid mode as well. It would be worth investigating whether the use of another clustering algorithm (irrespective of whether it is deterministic or non-deterministic) instead of HAC (Ward’s method) will improve the capability of the proposed method in class discovery. Furthermore, Euclidean distance is the measure of dissimilarity used for our experiments with the proposed algorithm. Exploring whether the use of another distance measure instead of Euclidean distance would improve the performance of the proposed algorithm is another significant future work.

Many studies in the literature aimed at ‘cancer subtype discovery’ from genomic data used only a single type of data—the gene expression data. Gene expression data, both from Microarray and RNA-Seq experiments, are able to capture important but partial biomolecular portraits of patients and the disease. As mentioned earlier, discovering potential novel subtypes by employing multi-omics data is the current thrust area in cancer research. Various methods such as iCluster [36], SNP [41], and SRF [15] have been proposed in the literature for the same. A possible enhancement of $\text{SilHAC}$ is to extend it to be able to do cancer subtype discovery from multi-omics data. The extension can employ the techniques such as the ones used by the above methods to get a unified data from multi-omics data and then perform clustering using $\text{SilHAC}$.

6 Conclusion

Discovery of novel subtypes of cancers leads to more refined and targeted development of drugs and which would ultimately lead to better survival rates of cancer patients. An extensive collection of molecular data of cancer tissues are now available at publicly accessible repositories such as NCBI Gene Expression Omnibus. Cancer subtype discovery from such molecular data is an exploratory analysis, and clustering algorithms have been used extensively for the purpose. Discovery of subtypes aims at finding natural groups among tissue samples based on molecular data. A popular method which is employed by researchers to identify novel subtypes of cancers from such genomic data is to use a clustering algorithm together with an internal cluster validity index. Consensus Clustering has been another method of choice for the researchers.

In this paper, we presented a Hierarchical Agglomerative Clustering algorithm named $\text{SilHAC}$. The algorithm employs Ward’s framework for Hierarchical Agglomerative Clustering and uses a newly proposed objective function named $\text{ExtASI}$ (which is based on the Average Silhouette Index) to select the pair of clusters to merge at each step. The algorithm runs in two modes—pure mode and hybrid mode. For large datasets, $\text{SilHAC}$ runs in the hybrid mode where it employs Hierarchical Clustering Algorithm (Ward’s Method) to reduce the initial set of clusters to consider. Similar to the other Hierarchical Clustering algorithms, $\text{SilHAC}$ enables dendrogram visualization.

A notable advantage of $\text{SilHAC}$, when compared to the other Hierarchical Agglomerative Clustering algorithms, is that $\text{SilHAC}$ finds out the number of natural clusters and gives the associated clustering solution. The number of clusters is directly indicated by the level of the clustering
hierarchy which has the maximum value for the objective function (ExtASi). SilHAC achieves it by taking values for three input parameters other than the dataset, and the values for which can be easily set by the user.

The abilities of SilHAC to estimate the number of natural clusters and to identify the clusters were tested on a collection of ten cancer gene expression datasets which includes five recently published datasets taken from NCBI GEO. A collection of five general datasets were also used for the performance analysis of the method. The method was able to find reasonable estimates for the number of natural clusters and has shown better overall performance than those of a collection of state-of-the-art methods. Hence, the proposed method can be a better alternative to the selected state-of-the-art methods being used to discover cancer subtypes from genomic data.

Furthermore, being a deterministic algorithm, SilHAC produces stable results unlike the widely used ‘clustering algorithm—Silhouette Index’ pipelines constituted using non-deterministic clustering algorithms such as K-Means, PAM, and Spectral Clustering, and variants of a prominent method called Consensus Clustering.

Silhouette Index is found to have a shortcoming that it tends to prefer a lower number of clusters especially in datasets where there is considerable variation in the inter-cluster distances. This variation influences the Silhouette Index-based cluster number identification methods, such as SilHAC, to prefer lower numbers of clusters than actual, which necessitates further cluster analysis of the obtained clusters. Rectifying this drawback of Silhouette Index is a work worth investigating. Moreover, exploring whether the use of distance measures other than Euclidean distance would lead to improvement in the performance of the method, and exploring whether the use of another clustering method in the hybrid mode instead of HAC (War’s method) are two possible future works. Furthermore, extending SilHAC so as to perform subtype discovery from multi-omics data is another line of future research.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Azzalini A, Menardi G (2014) Clustering via nonparametric density estimation: the R package pdfCluster. J Stat Softw 57(11):1–26
2. Barrett T, Wilhite SE, Ledoux P, Evangelista C, Kim IF, Tomashevsky M, Marshall KA, Phillippy KH, Sherman PM, Holko M et al (2012) NCBI GEO: archive for functional genomic data sets-update. Nucleic Acids Res 41(D1):D991–D995. https://doi.org/10.1093/nar/gks1193
3. Bhattacharjee A, Richards WG, Staunton J, Li C, Monti S, Vasa P, Ladd C, Beheshti J, Bueno R, Gillette M, Loda M, Weber G, Mark EJ, Lander ES, Wong W, Johnson BE, Golub TR, Sugarbaker DJ, Meyerson M (2001) Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. Proc Nat Acad Sci 98(24):13790–13795. https://doi.org/10.1073/pnas.191502998
4. Cai M, Li L (2017) Subtype identification from heterogeneous TCGA datasets on a genomic scale by multi-view clustering with enhanced consensus. BMC Med Genomics 10(4):75. https://doi.org/10.1186/s12920-017-0306-x
5. Cancer Genome Atlas Research Network et al (2012) Comprehensive molecular portraits of human breast tumors. Nature 490(7418):61. https://doi.org/10.1038/nature11412
6. Cancer Genome Atlas Research et al (2013) Integrated genomic characterization of endometrial carcinoma. Nature 497(7447):67–73. https://doi.org/10.1038/nature12113
7. Cavalli FM, Remke M, Rampasek L, Peacock J, Shih DJ, Loo B, Garzia L, Tchoria J, Nor C, Morrissy AS et al (2017) Intertumoral heterogeneity within Medulloblastoma subgroups. Cancer Cell 31(6):737–754. https://doi.org/10.1016/j.ccell.2017.05.005
8. Chalise P, Fridley BL (2017) Integrative clustering of multi-level omic data based on non-negative matrix factorization algorithm. PLoS ONE 12(5):1–18. https://doi.org/10.1371/journal.pone.0176278
9. Cirillo G, Gattz M, Beck A, Wilkerson M, Rhie S, Pastore A, Zhang H, McLellan M, Yao C, Kandoth C, Bowry RL, Shen H, Hayat S, Fieldhouse R, Lester S, Tse G, Factor R, Collins L, Allison K et al (2015) Comprehensive molecular portraits of invasive lobular breast cancer. Cell 163(2):506–519. https://doi.org/10.1016/j.cell.2015.09.033
10. de Souto MC, Costa IG, de Araujo DS, Ludermir TB, Schliep A (2008) Clustering cancer gene expression data: a comparative study. BMC Bioinform 9(1):497. https://doi.org/10.1186/1471-2105-9-497
11. Galli T (2015) dendextend: an R package for visualizing, adjusting, and comparing trees of hierarchical clustering. Bioinformatics. https://doi.org/10.1093/bioinformatics/btv428
12. Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, Coller H, Loh ML, Downing JR, Caligiuri MA et al (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286(5439):531–537. https://doi.org/10.1126/science.286.5439.531
13. Gowrishankar B, Przybycin CG, Ma C, Nandula SV, Rini B, Campbell S, Klein E, Chaganti R, Magi-Galluzzi C, Houldsworth J (2015) A genomic algorithm for the molecular classification of common renal cortical neoplasms: development and validation. J Urol 193(5):1479–1485. https://doi.org/10.1016/j.juro.2014.11.099
14. Griesinger AM, Josephson RJ, Donson AM, Levy JMM, Amani V, Birks DK, Hoffman LM, Furtet SL, Reigan P, Handler MH et al (2015) Interleukin-6/STAT3 pathway signaling drives an inflammatory phenotype in Group A ependymoma. Cancer Immunol Res. https://doi.org/10.1158/2326-6066.CIR-15-0061
15. Guo Y, Zheng J, Shang X, Li Z (2018) A similarity regression fusion model for integrating multi-omics data to identify cancer subtypes. Genes. https://doi.org/10.3390/genes9070314
16. Hubert L, Arabie P (1985) Comparing partitions. J Classif 2(1):193–218. https://doi.org/10.1007/BF01908075
17. Karatzoglou A, Smola A, Hornik K, Zeileis A (2004) kernlab—an S4 package for kernel methods in R. J Stat Softw 11(9):1–20
18. Karlsson A, Brunnström H, Micke P, Veerla S, Mattsson J, La Fleur L, Botling J, Jönsson M, Reuterswärd C, Planck M et al (2017) Gene expression profiling of large cell lung cancer links transcriptional phenotypes to the new histological WHO 2015 classification. J Thorac Oncol. https://doi.org/10.1016/j.jtho.2017.05.008

19. Kaufman L, Rousseuw P (1987) Clustering by means of medoids. In: Dodge Y (ed) Statistical data analysis based on the L1-norm and related methods. Elsevier Science Pub. Co., Amsterdam, pp 405–416

20. Laiho P, Kokko A, Vanharanta S, Salovaara R, Sammalkorpi H, Järvinen H, Mecklin J, Karttunen T, Tuppurainen K, Davalos V, Arango D, Aalten LA (2007) Serrated carcinomas form a subclass of colorectal cancer with distinct molecular basis. Oncogene 26(2):312

21. Li Z, Chen Y, Hu S, Zhang J, Wu J, Ren W, Shao N, Ying X (2016) Integrative analysis of protein-coding and non-coding RNAs identifies clinically relevant subtypes of clear cell renal cell carcinoma. Oncotarget 7(50):82671. https://doi.org/10.18632/oncotarget.12340

22. Lichman M (2013) UCI machine learning repository. https://doi.org/http://archive.ics.uci.edu/ml. Accessed 08 Mar 2018

23. Lloyd S (1982) Least squares quantization in PCM. IEEE Trans Inf Theory 28(2):129–137. https://doi.org/10.1109/TIT.1982.1056489

24. Lu J, Getz G, Alvarez-Saavedra E, Lamb J, Peck D, Sweet-Cordero A, Ebert BL, Mak RH, Ferrando AA, Downing JR, Jacks T, Horwitz HR, Golub TR (2005) MicroRNA expression profiles classify human cancers. Nature 435(7043):834

25. Maechler M, Rousseeuw P, Struyf A, Hubert M, Hornik K (2017) Cluster: cluster analysis basics and extensions. R package version 2.0.6

26. Mehmod R, El-Ashram S, Bie R, Dawood H, Kos A (2017) Clustering by fast search and merge of local density peaks for gene expression microarray data. Sci Rep. https://doi.org/10.1038/srep45602

27. Mertins P, Mani DR, Ruggles KV, Gillette MA, Clauser KR, Wang P, Wang X, Qiao JW, Cao S, Petralia F, Kawaler E, Munding F, Krug K, Tu Z, Lei JT, Gatza ML, Wilkerson M, Perou CM, Yellapantula V, Ki Huang et al (2016) Proteogenomics connects somatic mutations to signalling in breast cancer. Nature 534:55.

28. Monti S, Tamayo P, Mesirov J, Golub T (2003) Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. Mach Learn 52(1):91–118. https://doi.org/10.1023/A:1023949509487

29. Mouselimis L (2018) ClusterR: Gaussian mixture models, K-means, mini-batch-KMeans and K-Medoids clustering. https://CRAN.R-project.org/package=ClusterR. R package version 1.1.1

30. Ng AY, Jordan MI, Weiss Y (2002) On spectral clustering: a resampling-based method for class discovery and validation of gene expression microarray data. Mach Learn 49(2/3):105–137

31. Nidheesh N, Abdul Nazeer KA, Ameer PM (2017) An enhanced deterministic K-means clustering algorithm for cancer subtype prediction from gene expression data. Comput Biol Med 91:213–221. https://doi.org/10.1016/j.compbiomed.2017.10.014

32. R Core Team (2017) R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna. https://www.R-project.org/

33. Reddy CK, Vinzamuri B (2013) A survey of partitional and hierarchical clustering algorithms. In: Aggarwal CC, Reddy CK (eds) Data clustering: algorithms and applications, Chap 4. Chapman & Hall/CRC, Boca Raton, pp 87–110

34. Rousseeuw PJ (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 20:53–65

35. Senbabaog˘lu Y, Michailidis G, Li IZ (2014) Critical limitations of consensus clustering in class discovery. Sci Rep 4:6207. https://doi.org/10.1038/srep06207

36. Shen R, Olshen AB, Ladanyi M (2009) Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis. Bioinformatics 25(22):2906–2912. https://doi.org/10.1093/bioinformatics/btp543

37. Shi J, Malik J (2000) Normalized cuts and image segmentation. IEEE Trans Pattern Anal Mach Intell 22(8):888–905. https://doi.org/10.1109/34.868688

38. Shi Q, Zhang C, Peng M, Yu X, Zeng T, Liu J, Chen L (2017) Pattern fusion analysis by adaptive alignment of multiple heterogeneous omics data. Bioinformatics 33(17):2706–2714. https://doi.org/10.1093/bioinformatics/btx176

39. Tan PN, Steinbach M, Kumar V (2005) Introduction to data mining, 1st edn. Addison-Wesley Longman Publishing Co., Inc., Boston

40. Verhaak RG, Hoadley KA, Purdom E, Wang V, Qi Y, Wilkerson MD, Miller CR, Ding L, Golub T, Mesirov JP, Alexe G, Lawrence M, O’Kelly M, Tamayo P, Weir BA, Gabriel S, Winckler W, Gupta S, Jakkula L, Feiler HS et al (2010) Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in PDGFRA, IDH1, EGFR, and NF1. Cancer Cell 17(1):98–110. https://doi.org/10.1016/j.ccr.2009.12.020

41. Wang B, Mezlini AM, Demir F, Fiume M, Tu Z, Brudno M, Haibe-Kains B, Goldenberg A (2014) Similarity network fusion for aggregating data types on a genomic scale. Nat Methods 11:333. https://doi.org/10.1038/nmeth.2810

42. Ward JH Jr (1963) Hierarchical grouping to optimize an objective function. J Am Stat Assoc 58(301):236–244

43. Wilkerson MD, Hayes DN (2010) ConsensusClusterPlus: a class discovery tool with confidence assessments and item tracking. Bioinformatics 26(12):1572–1573

44. Wiwie C, Baumbach J, Rötting R (2015) Comparing the performance of biomedical clustering methods. Nat Methods 12(10):1333. https://doi.org/10.1038/nmeth.3583

45. Wu D, Wang D, Zhang MQ, Gu J (2015) Fast dimension reduction and integrative clustering of multi-omics data using low-rank approximation: application to cancer molecular classification. BMC Genom 16(1):1022. https://doi.org/10.1186/s12864-015-2223-8

46. Yu Z, Wong HS, You J, Yang Q, Liao H (2011) Knowledge based cluster ensemble for cancer discovery from biomolecular data. IEEE Trans Nanobiosci 10(2):76–85

47. Yu Z, Li L, You J, Wong HS, Han G (2012) ScS*3S: triple spectral clustering-based consensus clustering framework for class discovery from cancer gene expression profiles. IEEE/ACM Trans Comput Biol Bioinform 9(6):1751–1765

48. Yu Z, You J, Li L, Wong HS, Han G (2012) Representative distance: a new similarity measure for class discovery from gene expression data. IEEE Trans Nanobiosci 11(4):341–351

49. Zheng S, Cherniack A, Dewal N, Morfit R, Danilova L, Murray JR, Jacks T, Horwitz HR, Golub TR (2005) MicroRNA expression profiling of large cell lung cancer links transcriptional phenotypes to the new histological WHO 2015 classification. J Thorac Oncol. https://doi.org/10.1016/j.jtho.2017.05.008

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.