flowPeaks: a fast unsupervised clustering for flow cytometry data via $K$-means and density peak finding

Yongchao Ge* and Stuart C. Sealfon

Department of Neurology and Center of Translational System Biology, Mount Sinai School of Medicine, New York, NY 10029, USA

1 INTRODUCTION

In analyzing flow cytometry data, one fundamental question is how to divide the cells into distinct subsets with the phenotypes defined by the fluorescent intensity of the cell surface or intracellular markers. The unsupervised clustering for flow cytometry data is traditionally done by manual gating, where cells are sequentially clustered (gated) in one-dimension (1D) or 2D with the aid of 2D contour plots and 1D histograms. Manual gating has two problems: it is (i) highly subjective, depending on the users’ expertise and the sequences of the markers to draw the gates and where to draw the gates and (ii) tedious, for data consisting of $n$ channels, the user needs to check and draw the gates on possibly $\binom{n}{2}$ pairs of 2D contour plots. The automatic gating of the cells, in machine learning called unsupervised clustering, has become an active research area for the past several years. There are currently two common approaches to address the unsupervised clustering problem, one is based on the finite mixture model and the other on spatial exploration of the histograms. The former is computationally slow and has difficulty to identify clusters of irregular shapes. The latter approach cannot be applied directly to high-dimensional data as the computational time and memory become unmanageable and the estimated histogram is unreliable. An algorithm without these two problems would be very useful.

Results: In this article, we combine ideas from the finite mixture model and histogram spatial exploration. This new algorithm, which we call flowPeaks, can be applied directly to high-dimensional data and identify irregular shape clusters. The algorithm first uses $K$-means algorithm with a large $K$ to partition the cell population into many small clusters. These partitioned data allow the generation of a smoothed density function using the finite mixture model. All local peaks are exhaustively searched by exploring the density function and the cells are clustered by the associated local peak. The algorithm flowPeaks is automatic, fast and reliable and robust to cluster shape and outliers. This algorithm has been applied to flow cytometry data and it has been compared with state of the art algorithms, including Misty Mountain, FLOCK, flowMeans, flowMerge and FLAME.


Contact: yongchao.ge@mssm.edu

Supplementary information: Supplementary data are available at Bioinformatics online

Received on March 19, 2012; revised on April 27, 2012; accepted on May 14, 2012

2 METHODS

2.1 What is a cluster

As said in Jain (2010), there is inherent vagueness in the definition of a cluster. We want to illustrate what a cluster is with a toy example. Figure 1 shows a density function of two Gaussian distributions when varying the mean of the first distribution. In Figure 2, the means are fixed, and the proportion for the first Gaussian distribution is varied. Most figures show two distinct peaks. However, we can see that the data should be considered...
K-means++ is not for the final clustering, but for a first partition of the cells, for which the black curve is the probability density at -3 and the clustering, and was applied to flow cytometry data as early as in Murphy (2011). In fact, we could have applied the Hartigan and Wong’s (1979) algorithm to recompute the cluster centers and cluster membership to decrease the objective function \( \sum_{i=1}^{n} \|x_i - c_{k_i}\|^2 \), where \( L_1 \leq \ldots \leq L_K \) is the cluster label of \( x_i \) and \( x_j \) is the center vector for cluster \( k \). We could have applied the Hartigan and Wong’s algorithm directly to the seeds, but the computation is too slow.

In general clustering, it is important to specify a good K in the K-means algorithm. For our purpose, a very accurate specification of K is not necessary. However, it is still important that the K can give a smooth density in which the peaks can reveal the clustering structure. This specification of K is similar to the determination of the number of bins in drawing histograms. We adopted the formula of Freedman and Diaconis (1981)

\[
K = \lceil \text{median}(K_1, \ldots, K_d) \rceil,
\]

where \( K_1, \ldots, K_d \) are, respectively, the minimum and maximum of the j-th dimension of the data \( x_1, x_2, \ldots, x_n \) and IQR(\( x_i \)) is the interquartile range of the data, defined as the difference between the 75th percentile and 25th percentile. Then our K is defined as the median of \( K_1, \ldots, K_d \).

2.3 Gaussian finite mixture to model the density function

After K-means, we may approximate the density function \( f(x) \) by the Gaussian mixture density function

\[
f(x) = \sum_{k=1}^{K} w_k \phi(x; \mu_k, \Sigma_k),
\]

where the proportion \( w_k \) of the k-th component satisfies \( 0 \leq w_k \leq 1 \) and \( \sum_{k=1}^{K} w_k = 1 \) and \( \phi(x; \mu_k, \Sigma_k) \) is the probability density function of the multivariate normal distribution with mean \( \mu_k \) and variance matrix \( \Sigma_k \). After applying the K-means algorithm of Section 2.2, we have already partitioned the data into K clusters, and for the K-th cluster, we can compute the sample proportion \( w_k \), sample \( \mu_k \) and sample variance matrix \( \Sigma_k \) (a rigorous writing would require the hat notation, which is ignored for the sake of simplicity). However, the estimate \( \Sigma_k \) may be too noisy, and we want to smooth the variance matrix by

\[
\Sigma_k = h_k \Sigma_k + (1 - h_k) b_h \Sigma_0,
\]

where \( h_k \) and \( b_h \) are customized parameters tuned to make the density function smoother or rougher. The default setting in the software is \( h = 1.5 \) and \( b_h = 1 \).

Here, \( L_k = w_{k+1}/(k+m_k) \) so that a greater \( w_k \) results in a larger \( L_k \) closer to 1; \( \Sigma_0 \) is the variance matrix assuming the data are uniformly distributed over the whole data range and is a diagonal matrix with its \( i, j \) element \( \Sigma_0^{ij} = \|x_i - x_j\|^2 \) for \( i, j = 1, \ldots, d \).
2.4 Peak search and merging

According to our definition, a cluster is defined by the local peak. For all cells, we can use the greatest gradient search (hill climbing) to find which local peak a given cell can reach. This rule out any global optimization strategy such as the conjugate gradient algorithm. It is computationally very time consuming to search all the local maxima of the density function for all cells. Since the cells are pre-grouped by the K-means, we only need to search the local peaks for the centers of the K-means clusters. The hill climbing method searches along the greatest gradient of the density function. If we take the negative of the density function as the optimization function, the hill climbing of peak search can be achieved by the deepest descent algorithm, which is implemented by the GSL library at http://www.gnu.org/ software/gsl/. We also need to restrict the step size in case it steps too far away and jumps to another local peak. When the data move from one K-means cluster into another K-means cluster, we can speed it up by moving directly to the center of the other cluster. When two peaks are relatively close, they should be joined together and considered as a single peak. We search the two peaks with the closest Euclidean distance and check if the two clusters may not be too different from a single cluster. The details on the peak search and peak merging are described in the Appendix. Algorithm 1 gives the summary of the steps to use in K-means and density peak finding in order to cluster the flow cytometry data as implemented in the software flowPeaks. In the end, we will obtain $K \leq K$ of merged clusters, each of which consists of one or many K-means clusters.

Algorithm 1 Summary of the flowPeaks algorithm
1. Apply the Freedman-Diaconis formula in each dimension of the data to obtain the number $K$ of clusters for K-means [see Equations (1) and (4)].
2. Use the K-means++ algorithm to generate the initial seeds of the $K$ clusters.
3. Use the $k$-d tree data representations to apply Lloyd’s K-means algorithm until it converges.
4. Further apply the Hartigan and Wong’s K-means algorithm to improve the compactness of the clusters.
5. Compute $k$, $K_k$, $\sum_2$ for $k = 1, \cdots , K$ using the partitions of the $K$-means.
6. Based on the density function generated by Gaussian finite mixture model, compute the local peak starting from the centers $\mu_k$, $k = 1, \cdots , K$ (see Algorithm A1 in the Appendix).
7. Apply Algorithm A2 in the Appendix to merge peaks hierarchically.
8. The $K$ clusters of the final K-means algorithm are regrouped according to the merged peaks.

2.5 Cluster tightening

The default setting in the flowPeaks algorithm is to not identify the outliers. Some data points may lie far from the center or cannot be unambiguously classified into a specific cluster. We determine whether a data point is an outlier using the following strategy. Let $f(x)$ be the final merged cluster label of data point $x$. Let $w_i$ and $f_i(x)$ (respectively) be the proportion and the probability density function of the $i$-th final merged cluster. The proportion $w_i$ is the sum of $w_i$'s of the K-means clusters that form the $i$-th final merged cluster. The density function $f_i(x)$ itself is a Gaussian finite mixture based on the K-means clusters that are merged into the $i$-th final cluster, while the overall density function $f(x)$ is based on all K-means clusters (see Section 2.3) and $f(x) = \sum_{i=1}^{n}w_i f_i(x)$. A point $x$ is an outlier if $f(x)/\max(f_i(x))$: $f(x)/f_i(x) \leq 0.01$, or $\sum_{i=1}^{n} w_i f_i(x) \leq 0.8$.

The numbers 0.01 and 0.8 can be adjusted in the software settings.

3 RESULTS

3.1 Datasets

Barcode data: The data were generated for a barcoding experiment by [Krutzik and Nolan, 2006] with varying concentrations of fluorophores (APC and Pacific Blue). The flow cytometry data have 180912 cells and three channels with an additional channel for Alexa. The manual gates for the 20 clusters to be used for assessing cluster algorithm performance were created from FlowJo (www. flowjo.com).

Simulated concave data: The data were simulated with two distinctive concave shapes based on the idea from the supplemental material of [Pyne et al., 2009]. It has 2729 rows and 2 columns. Both barcode data and simulated concave data along with their gold standard cluster labels are available in the flowPeaks package.

GvHD dataset: Graft versus host disease dataset and the manual gates are obtained from [Azaripour et al., 2011]. This dataset contains 12 samples, and the cells are stained with four markers, CD4, CD8b, CD3 and CD8. In addition, two channels FS and SS are also measured. These data are mostly analyzed based on the four markers unless specified otherwise. The numbers of cells of the 12 samples range from 12000 to 32000.

Rituximab data: The flow cytometry data that are obtained from the flowClust package [Lo et al. , 2006]. They have 1545 cells and two channels of interest. The data were originally produced by [Gasparetto et al., 2004]. The barcode data, simulated data and GvHD datasets have gold standard cluster labels (either by simulation or manual gating) to assess performance. The rituximab data are used for the purpose of exploration. Figure 3 displays all four datasets.

3.2 Different metrics to assess the cluster algorithm performance

The most widely used metric to assess how a candidate clustering algorithm compares with the gold standard, for which the correct cluster membership is known, is the adjusted Rand index [Hubert and Arabie, 1983; Rand, 1971]. The Rand index [Rand, 1971] is based on the percentage of the agreement between the two clustering methods. Let us assume that $n$ data points are labeled differently with two different clustering methods, say Method A and Method B with $K_A$ and $K_B$ clusters. Let $A_i, i = 1, \ldots , n$ and $B_i, i = 1, \ldots , n$ be the cluster labels for the two methods. The Rand index is defined as

$$\text{index} = \sum_{1 \leq i < j \leq n} I(A_i = A_j, B_i = B_j) \left( \begin{array}{c} n \\ 2 \end{array} \right)$$

where $I(\cdot)$ is the indicator function. The adjusted Rand corrects for chance, and the general form is

$$\text{Index} = \frac{\text{Expected Index}}{\text{Max Index} - \text{Expected Index}}$$

2054
In order to compute the adjusted Rand index, we first define the contingency tables

\[ n_{a,b} = \sum_{i=1}^{n} I(A_i = a \text{ and } B_i = b), \]

for \( a = 1, \ldots, K_A \) and \( b = 1, \ldots, K_B \). The marginal sums on the contingency tables are then defined as

\[ n_{a,+} = \sum_{b=1}^{K_B} n_{a,b}, \quad n_{+,b} = \sum_{a=1}^{K_A} n_{a,b}, \]

Note that \( n = \sum_{a=1}^{K_A} n_{a,+} = \sum_{b=1}^{K_B} n_{+,b} \). The adjusted Rand index can be quickly computed using the following formula (Hubert and Arabie, 1985):

\[ \frac{\sum_{a,b} n_{a,b}^2 - \sum_{a} n_{a,+}^2 \sum_{b} n_{+,b}^2}{\sum_{a} n_{a,+}^2 \sum_{b} n_{+,b}^2 - \sum_{a} n_{a,+}^2 \sum_{b} n_{+,b}^2 / 2 - \sum_{a} n_{a,+}^2 \sum_{b} n_{+,b}^2 / 2} \]

The \( F \)-measure was proposed by Hubert and Arabie (1985) is based on a greedy strategy to match the two clustering. It has been used in 2010s flowCAP1 (http://flowcap.flowsite.org/summit2010.html) and in the flowMeans algorithm paper (Aghaeepour et al. 2011) to assess the performance of different algorithms. The \( F \)-measure is defined as

\[ F = \frac{n_{a,b} \cdot \max_{a,b} F(a,b)}, \]

where \( F(a,b) = \frac{2n_{a,b}}{n \cdot n_a + n_b n}, \)

\( n_a = \sum_{b=1}^{K_B} n_{a,b}, \quad n = \sum_{a=1}^{K_A} n_{a,+} \)

Rosenberg and Hirschberg (2007) proposed the \( V \)-measure to evaluate the clustering algorithm. This measure uses entropy to assess how much a second clustering provides extra information for the first clustering. For the clustering Method A, the entropy is

\[ H(A) = \sum_{a=1}^{K_A} n_{a,+} \log \frac{n_{a,+}}{n} \]

and the conditional entropy

\[ H(A|B) = \sum_{a=1}^{K_A} n_{a,+} \log \frac{n_{a,+}}{n} \]

The conditional entropy \( H(A|B) \) is always no greater than the entropy \( H(A) \). The extra information provided by Method B for Method A is the reduced entropy \( H(A) - H(A|B) \). After normalization, we can define

\[ h = 1 - \frac{H(A|B)}{H(A)} \]

In the above equation, by definition \( h = 1 \) if \( H(A) = 0 \). If we reverse the positions of A and B, we can define

\[ c = 1 - \frac{H(B|A)}{H(B)} \]

If Method B is the candidate clustering to be compared with the gold standard clustering \( A \), \( h \) evaluates the homogeneity of clustering for Method B, while \( c \) evaluates the completeness. The homogeneity ensures that the gold standard labels (A labels) for all data points of a candidate cluster B are unique. Completeness ensures that for each gold standard cluster (A cluster), data points are all assigned to a single candidate cluster (B cluster). Details can be found in Rosenberg and Hirschberg (2007). The \( V \)-measure is a weighed harmonic mean of \( h \) and \( c \), \( V_{\beta} = \frac{1+\beta^2}{\beta^2 (h) + c} \). In this aritice, we will fix \( \beta \) to be 1.
Fig. 4. Application of flowPeaks to the barcode data. (A) the bold boundary displays the clusters output by flowPeaks with their centers (⊕), the dotted lines are the boundary for the underlying K-means clusters with their centers (◦). The local peaks are indicated by △. (B) The same as in (A) except the outliers have been identified as black points and other secondary information was not displayed, and the clusters are labeled according to their proportions ($w_k$) (A color version of this figure is available as Supplementary Material).

Table 1. Comparison of the running time of different flow cytometry clustering algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Concave</th>
<th>Barcode</th>
</tr>
</thead>
<tbody>
<tr>
<td>flowPeaks</td>
<td>0.13</td>
<td>2.3</td>
</tr>
<tr>
<td>Misty Mountain</td>
<td>0.59</td>
<td>24.7</td>
</tr>
<tr>
<td>FLOCK</td>
<td>6</td>
<td>14</td>
</tr>
<tr>
<td>flowMeans</td>
<td>8.2</td>
<td>82.3</td>
</tr>
<tr>
<td>FLAME</td>
<td>1434</td>
<td>80952</td>
</tr>
<tr>
<td>flowMerge</td>
<td>3202</td>
<td>132446</td>
</tr>
</tbody>
</table>

The running time is shown in wall-clock seconds on the same desktop computer except that FLOCK and FLAME were run, respectively, at immport (http://immport.niaid.nih.gov) and gene pattern websites (http://www.broadinstitute.org/cancer/software/genepattern).

Table 2. The performance of different algorithms on the barcode data

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Concave</th>
<th>Barcode</th>
</tr>
</thead>
<tbody>
<tr>
<td>flowPeaks</td>
<td>0.997</td>
<td>0.971</td>
</tr>
<tr>
<td>Misty Mountain</td>
<td>0.971</td>
<td>0.894</td>
</tr>
<tr>
<td>FLOCK</td>
<td>0.258</td>
<td>0.341</td>
</tr>
<tr>
<td>flowMeans</td>
<td>0.998</td>
<td>0.993</td>
</tr>
<tr>
<td>FLAME</td>
<td>0.859</td>
<td>0.995</td>
</tr>
<tr>
<td>flowMerge</td>
<td>0.801</td>
<td>0.887</td>
</tr>
</tbody>
</table>

Adj-Rand is for the adjusted Rand index.

Table 3. Performance of different algorithms on the concave data

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Concave</th>
<th>Barcode</th>
</tr>
</thead>
<tbody>
<tr>
<td>flowPeaks</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>Misty Mountain</td>
<td>0.501</td>
<td>0.683</td>
</tr>
<tr>
<td>FLOCK</td>
<td>0.723</td>
<td>0.884</td>
</tr>
<tr>
<td>flowMeans</td>
<td>0.232</td>
<td>0.438</td>
</tr>
<tr>
<td>FLAME</td>
<td>0.952</td>
<td>0.987</td>
</tr>
<tr>
<td>flowMerge</td>
<td>0.675</td>
<td>0.438</td>
</tr>
<tr>
<td>FLAME</td>
<td>0.287</td>
<td>0.848</td>
</tr>
<tr>
<td>flowMerge</td>
<td>0.292</td>
<td>0.946</td>
</tr>
</tbody>
</table>

Adj-Rand is for the adjusted Rand index.

Table 4. The comparison of the performance on the 12 samples of the GvHD dataset

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Sample 1</th>
<th>Sample 2</th>
<th>Sample 3</th>
<th>Sample 4</th>
<th>Sample 5</th>
<th>Sample 6</th>
<th>Sample 7</th>
<th>Sample 8</th>
<th>Sample 9</th>
<th>Sample 10</th>
<th>Sample 11</th>
<th>Sample 12</th>
</tr>
</thead>
<tbody>
<tr>
<td>flowPeaks</td>
<td>0.807 (0.175)</td>
<td>0.675 (0.287)</td>
<td>0.573 (0.292)</td>
<td>0.924 (0.075)</td>
<td>0.859 (0.146)</td>
<td>0.848 (0.120)</td>
<td>0.816 (0.135)</td>
<td>0.664 (0.205)</td>
<td>0.639 (0.199)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Misty Mountain</td>
<td>0.675 (0.287)</td>
<td>0.573 (0.292)</td>
<td>0.859 (0.146)</td>
<td>0.816 (0.135)</td>
<td>0.664 (0.205)</td>
<td>0.639 (0.199)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>flowMeans</td>
<td>0.573 (0.292)</td>
<td>0.859 (0.146)</td>
<td>0.816 (0.135)</td>
<td>0.664 (0.205)</td>
<td>0.639 (0.199)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Adj-Rand is for the adjusted Rand index. Each entry lists the mean and standard deviation.

4 SOFTWARE

We have implemented the algorithm in C++ wrapped into an R package named ‘flowPeaks’. The following example illustrates how to use the basic functions of this R package

```r
library(flowPeaks)
data(barcode)
fp<-flowPeaks(barcode[,c(1,3)])
plot(fp,drawlocalpeaks=TRUE)
```

The above R script will display Figure 4A. In order to identify the outliers to obtain Figure 4B, we can proceed further with the following script

```r
fpc<-assign.flowPeaks(fp,fp$x)
plot(fp,classlab=fpc,drawboundary=FALSE,drawvor=FALSE,drawkmeans=FALSE,drawlab=TRUE)
```

For further use of the software flowPeaks, one can consult the package’s vignette pdf file and help documents.

5 DISCUSSION AND FUTURE WORK

In this article, we described the algorithm flowPeaks that combines the K-means and density function peak finding to partition the flow cytometry data into distinct clusters. We have compared our algorithm with other state of the art algorithms for real and simulated datasets. Our algorithm is fast and able to detect the non-convex shapes. We should point out that flowPeaks’s goal is to find the overall density shape and search for global structure. It will not be able to uncover overlapping clusters as shown in Figure 1C or the rare cluster as shown in Figure 2C. The flowPeaks algorithm is based on the geometrical shape of the density function. Prior to apply flowPeaks, data transformation may be necessary to reveal the structure, and irrelevant channels need to be first discarded to avoid the curse of dimensionality. Due to the curse of dimensionality, if the data dimension is too high and the number of cells is too low where...
code and windows binary built of the R package flowPeaks is available at https://github.com/yongchao/flowPeaks. The package is in the progress of being permanently hosted at the Bioconductor [Gentleman et al. 2004] [Hahne and Gentleman 1996] with open source code for algorithm developers and batch processing.

**APPENDIX**

**Mathematical notation**

For the sake of clarity, we will use the following notation. Assume the data consist of $n$ points in $d$ dimension.

Let the underlying clusters, obtained by K-means, be labeled as $1, \ldots, K$. The density function generated by the finite mixture model is

$$f(x) = \sum_{k=1}^{K} w_k / \Sigma_k \exp \left( -\frac{1}{2} \Sigma_k^{-1} (x - \mu_k) \right),$$

where $w_k, \mu_k, \Sigma_k$ are the weights, means, and the smoothed variance matrix of cluster $k$, respectively, for $k = 1, \ldots, K$. The derivative of the density function at $x$ is defined as

$$f'(x) = \frac{\partial f(x)}{\partial x}.$$

According to the $K$-means algorithm, the cluster label of $x$ can be defined as

$$L(x) = \arg\min_{k=1}^{K} ||x - \mu_k||^2.$$

**Searching the local peak starting from a point $x$**

As we do not want to jump over the local peak, when the data fall into a cluster $k$, we define the maximum step size

$$D^\max_k = \min_{1 \leq i \leq d} \sqrt{\Sigma_{x_i}}.$$

The detailed computations for the local peak search are described in Algorithm A1. We initially set a small step size $\beta$ (Step 0), and try to find a step size such that the density function $f$ improves (Step 2 and Step 3). If the same step size improves twice in a row ($N_{\text{success}}$ denote the number of continuous improvements), then we double the step size; otherwise we half the step size. If the point is falling into a new cluster, we want to find out if we can jump to the new center directly (Step 6). The details are described in Algorithm A1.

**The algorithm on merging local peaks**

When two peaks are close and the density function between the two peaks is relatively flat, the two peaks should be combined into one. For each underlying $K$-means cluster, we define the nearest neighbor cluster distance by

$$S_k = \min \{ |\mu_k - \mu_i| : i \in \{1, \ldots, K\} \ and \ i \neq k \}.$$

For an arbitrary point $x$, we can similarly define the function $S(x) = S_{\text{closest}}$. Let $x$ and $y$ be two points, we define the tolerance that describes how the density function of the line segment that connects $x$ and $y$ can be approximated by a straight line

$$\text{tol} = \max_{i \in \{0,1\}} \left( |f(x) - f(y)| / \sqrt{\frac{n(x,y) + n(y,x) / 2}{n/K}} \right).$$

The algorithm on merging local peaks is as follows.

1. **Local Peak Search**
   - For each point $x_i$, find the local maximum $x^*$ within a specified radius $\theta$.
   - Let $\alpha$ be the parameter to control the smoothness of the density function.

2. **Cluster Merger**
   - For each pair of local peaks $x^*$ and $x^2$, quantify their difference in density using the tolerance $\text{tol}$. If this difference is below a certain threshold, merge the clusters.

**Algorithm A1**

1. **Initialization**
   - Set initial parameters $\theta$ and $\alpha$.
   - For each point $x_i$, find the nearest neighbor $x^*$ within a radius $\theta$.

2. **Local Peak Search**
   - For each point $x^*$, find the local maximum $x^*$ within a specified radius $\theta$.

3. **Cluster Merger**
   - For each pair of local peaks $x^*$ and $x^2$, quantify their difference in density using the tolerance $\text{tol}$. If this difference is below a certain threshold, merge the clusters.

**Algorithm A2**

1. **Local Peak Search**
   - For each point $x_i$, find the local maximum $x^*$ within a specified radius $\theta$.

2. **Cluster Merger**
   - For each pair of local peaks $x^*$ and $x^2$, quantify their difference in density using the tolerance $\text{tol}$. If this difference is below a certain threshold, merge the clusters.

The details of the algorithm are described in more formal mathematical notation and pseudocode in the appendix.
Algorithm A1 searching for the local peak starting from an input point $x$

1. If $\beta$ is small or $|f'(x_0)|$ is small, stop. If $n$ is too large, stop.
2. If $f(y) > f(x_0)$, then $N_{loc} := N_{loc} + 1$ and go to Step 4; otherwise $\beta := \beta/2$, $N_{loc} := 0$ and go to Step 1
3. If $N_{loc} > 2$, then $\beta := \min(2\beta, \beta_{max})$.
4. Let $y := x_0 + \beta f'(x_0)/|f'(x_0)|$ and $h(y)$.
5. Let $x_{n+1} := y$ and $k_{n+1} := f(y)$.
6. Update $\beta$ by setting $\beta$ to $\beta_{max}/10$. Check if we can jump to Step 7 directly to the new cluster: if $f(\mu_{k-1}) > f(x_{n+1})$, then set $\mu_{k+1} := \mu_{k-1}$.
7. Update $n := n + 1$, go to Step 1.

where $z_i = x + f(y - x)$ and $f(z_i) = f(x) + f(y - f(x))$. The function $f(z_i)$ is the fitted density function at the position $z_i$ by using a straight line to connect the two points $(x, f(x))$ and $(y, f(y))$.

The second term in defining tol corrects for cluster sample sizes.

Many K-means centers may reach the same local peak. A local peak can then be represented by a subset $f(x)$, where $\beta$ is the number of distinct local peaks. In other words, for each $k$ in $P_1$, $\beta_k$ will move to the same $y$ by using our local peak algorithm. Initially, set $G_0 = \{g\}$, $g = 1, \ldots, N_P$, i.e. each peak set just contains a single peak (Step 0). Two peak sets can be merged only if the two peaks are relative close and the density function between the peaks is relatively flat (Step 1). $G_0$ are merged hierarchically (Step 2). The details are given in Algorithm A2. After the algorithm completes, $N_G$ is the number of $G$ (see Section 2.3) final clusters.

Algorithm A2 peaking merging algorithm

1. Set $G_0 = \{g\}$, $g = 1, \ldots, N_P$, where $N_G$ is initially $N_P$.
2. Set $G_0 = \{g\}$, $g = 1, \ldots, N_P$, where $G_0$ is initially $N_P$.
3. If $N_G = 1$ stop; otherwise go to Step 1.

Acknowledgements

We thank Fernand Hayot, Istvan Sugar and German Nuñez for valuable comments and discussions. We thank Ryan Brinkman and Nima Aghaeepour for providing the GvHD dataset and the associated manual gates. We appreciate the reviewers’ insightful comments, resulting in a much improved article.

Funding: National Institute of Allergy and Infectious Diseases [contract HHSN 266200500021C].

Conflict of Interest: none declared.

REFERENCES


