


# Improve consensus partitioning via a hierarchical procedure

Zuguang Gu  and Daniel Hübschmann†

Corresponding author: Zuguang Gu, Molecular Precision Oncology Program, National Center for Tumor Diseases (NCT) Heidelberg, Im Neuenheimer Feld 280, Heidelberg 69120, Germany. Tel.: +49 6221 42 3607; E-mail: z.gu@dkfz.de.

†Daniel Hübschmann, Molecular Precision Oncology Program, National Center for Tumor Diseases (NCT) Heidelberg, Im Neuenheimer Feld 280, 69120 Heidelberg, Germany. Heidelberg Institute of Stem Cell Technology and Experimental Medicine (HI-STEM), Im Neuenheimer Feld 280, 69120 Heidelberg, Germany. German Cancer Consortium (DKTK), Im Neuenheimer Feld 280, 69120 Heidelberg, Germany. Department of Pediatric Immunology, Hematology and Oncology, University Hospital Heidelberg, 69120 Heidelberg, Germany. E-mail: d.huebschmann@dkfz.de

## Abstract

Consensus partitioning is an unsupervised method widely used in high-throughput data analysis for revealing subgroups and assigning stability for the classification. However, standard consensus partitioning procedures are weak for identifying large numbers of stable subgroups. There are two major issues. First, subgroups with small differences are difficult to be separated if they are simultaneously detected with subgroups with large differences. Second, stability of classification generally decreases as the number of subgroups increases. In this work, we proposed a new strategy to solve these two issues by applying consensus partitioning in a hierarchical procedure. We demonstrated hierarchical consensus partitioning can be efficient to reveal more meaningful subgroups. We also tested the performance of hierarchical consensus partitioning on revealing a great number of subgroups with a large deoxyribonucleic acid methylation dataset. The hierarchical consensus partitioning is implemented in the R package *cola* with comprehensive functionalities for analysis and visualization. It can also automate the analysis only with a minimum of two lines of code, which generates a detailed HTML report containing the complete analysis. The *cola* package is available at <https://bioconductor.org/packages/cola/>.

**Keywords:** consensus partitioning, unsupervised classification, hierarchical method, Bioconductor, R package

## Introduction

Consensus partitioning or consensus clustering is an unsupervised learning method that classifies samples into subgroups and evaluates the stability of the classification by resampling from original data [1]. It has become an important tool applied in high-throughput data analysis e.g. to reveal cancer subtypes [2] or to validate the agreement of the classification on known clinical factors. In our previous work [3], we developed an R/Bioconductor package named *cola* that provides a general framework for consensus partitioning. It allows simultaneously running multiple feature selection methods and partitioning methods and it provides comprehensive visualization and reporting utilities for automatic and deep interpretation on the results. *Cola* provides a new and efficient method named ATC (ability to correlate to other rows) for extracting top features and it recommends spherical *k*-means clustering [4] for subgroup classification. Through comprehensive benchmarks on public datasets, we demonstrated *cola* was able to generate new, stable and biologically meaningful classifications.

*Cola* provides a convenient toolkit for performing consensus partitioning analysis. It performs well when the expected number of subgroups is relatively small e.g. no larger than six as demonstrated in our previous study [3]. However, when the number of expected subgroups increases, issues for general consensus partitioning procedures [5, 6] rise and they would significantly affect the classification. In consensus partitioning procedures, first the top *n* features scored by a certain method e.g. standard deviation (SD), are selected. Later, sample classification is only applied to the top features. A good classification is expected to select those features which have the ability to separate all subgroups, in other words, consensus partitioning procedures take into account all samples equally. However, in real-world datasets, this condition cannot always be met. It is possible that features good at separating major subgroups (i.e. subgroups with large difference) are weak for secondary subgroups (i.e. subgroups with small difference) if the secondary subgroups have different sets of features that are efficient for

Zuguang Gu is a senior scientist at the National Center for Tumor Disease, Heidelberg, Germany. His research interests include statistical analysis on various types of high-throughput data. He is also active in R/Bioconductor package development for data visualization and analysis.

Daniel Huebschmann is a group leader of the Molecular Precision Oncology Program, National Center for Tumor Disease, Heidelberg, Germany.

Received: November 5, 2021. Revised: January 20, 2022. Accepted: January 30, 2022

© The Author(s) 2022. Published by Oxford University Press.

This is an Open Access article distributed under the terms of the Creative Commons Attribution-NonCommercial License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited. For commercial re-use, please contact [journals.permissions@oup.com](mailto:journals.permissions@oup.com)

classification. When the real number of subgroups becomes larger, it is highly possible that subgroups have different sets of efficient features for classification, and this leads to the effect that it may be difficult to reach stable separation for secondary subgroups when classifying them with major subgroups at the same time. The second issue is that, when the number of subgroups gets larger, the probability of two samples to be in different subgroups tends to increase, which results in the loss of stability of the classification. Both issues hinder the classification to reach a large number of stable subgroups.

In this work, to solve the previously raised issues, we propose a strategy named *hierarchical consensus partitioning* (HCP) that applies standard *cola* consensus partitioning (CP) in a hierarchical procedure. Simply speaking, one could first classify samples into  $k$  groups where  $k$  is a small number which corresponds to major subgroups. Then for each subgroup of samples, one could repeatedly apply CP with a new set of top features extracted only to that subset of samples. The hierarchical procedure stops until certain criteria are reached. By these means, theoretically, small subgroups or secondary subgroups could be detected in later steps of the hierarchical procedure. This process can generate a hierarchy of subgroups where subsets of samples are represented as nodes. The idea of executing CP hierarchically has also been applied when identifying consensus network modules to reveal multiresolution modularity of the network [7]. Hierarchical classification has been widely used in various fields, especially for large-scale classifications [8, 9] e.g. for prediction of gene functions [10]. Here we implement it in the framework of CP.

For large datasets with huge numbers of samples, in early steps of the hierarchical procedure, numbers of samples in the subsets could still be large. Due to that, CP by-nature is a time-consuming analysis. To improve the efficiency of partitioning on large datasets, we propose a strategy which randomly picks samples to a small subset, on which CP is applied, later the class labels of the deselected samples are predicted based on the classification of the selected samples. This downsampling strategy ensures analysis of thousands of samples can be done in an acceptable time.

HCP extends the *cola* framework and it has been implemented in the *cola* package from version 2.0.0 on. For submatrices represented as nodes in the subgroup hierarchy, standard CP by *cola* is applied where specific combinations of feature selection methods and partitioning methods can be either user-defined or selected from the built-in methods. HCP provides rich visualizations for interpretation of the results, as well as comprehensive tools for downstream analysis, such as dimension reduction, signatures analysis and functional enrichment analysis if signatures can be mapped to genes. For the ease of use, HCP automates the analysis with a minimum of only two lines of code,

which generates a detailed HTML report containing the complete analysis.

In this paper, we first illustrate issues of CP with a simulated dataset and a real-world dataset. Then we demonstrate the use of HCP with an RNA-sequencing (RNAseq) dataset with an intermediate number of samples. The results show that HCP was able to reveal more subgroups than standard CP analysis. Next, we applied HCP on a single-cell RNA-sequencing (scRNAseq) dataset with a large number of cells, where the downsampling functionality was used for the analysis. The results show that HCP classification was similar to the one from the original study but cell clusters had larger separation under HCP classification. Last, we tested the performance of HCP on a large deoxyribonucleic acid (DNA) methylation dataset to demonstrate its ability to reveal a great number of subgroups in a completely unsupervised way.

## Methods

### A brief introduction to CP and the *cola* package

The HCP method proposed in this work is an extension of the CP implemented in the R package *cola*. To make the paper easy to read and self-explanatory, here we briefly describe the methods and terms used in *cola*. We furthermore kindly refer readers to the original publication for more details [3].

CP is applied on columns of matrix-like data to discover subgroups of samples e.g. a gene expression matrix where rows are genes and columns are patients. Matrix rows are firstly assigned with scores by a certain method such as the widely used SD, then only the top  $n$  features with the highest scores are used for CP. This selection is called the *top-value method* in *cola*. We proposed a new top-value method ATC (ability to correlate to other rows) in *cola* which aims to capture top features that are potentially highly correlated to other features and to provide more consistent patterns for subgroup classification. For row  $i$  in a matrix, denote the variable  $X$  as a vector of absolute values of the correlation coefficients to all other rows, the ATC score for row  $i$  is defined as:

$$ATC_i = 1 - \int_0^1 F_X(x) dx$$

where  $F_X(x)$  is the cumulative distribution function (CDF) of  $X$ . The aim of using the top  $n$  features for partitioning is to keep the informative features that help partitioning while removing other features with irrelevant noise. We demonstrated that ATC can capture better and distinct features for partitioning that cannot be captured by other top-value methods [3].

After top  $n$  features are selected, a certain partitioning method is repeatedly applied on randomly sampled subsets (e.g. 80%, either by rows or by columns) of features and stability of the partitioning is evaluated from the list of individual partitioning results i.e. how often two samples stay in the same subgroup. According to

the extensive benchmarks performed in our previous study [3], we demonstrated that the spherical  $k$ -means clustering (skmeans) could classify samples with higher stability.

To obtain the optimal number of subgroups in CP, a list of numbers of subgroups denoted as  $k$  is iterated. The best  $k$  is evaluated by several metrics. *Cola* mainly uses three metrics to determine the best  $k$ : mean silhouette score, proportion of ambiguous clustering (PAC) score and concordance scores, which measure the stabilities of the CP from different aspects. The silhouette score measures how close one sample is to its own subgroup compared with the closest neighboring subgroup and the mean of silhouette scores over all samples is used to measure the overall stability of the classification; PAC scores measure the proportion of ambiguous clustering's [11] where ambiguity is defined as two samples in the same group with probability between 0.1 and 0.9 in the repeated clusterings (the form 1-PAC is actually used in *cola* to let the direction of changes be the same as the other two metrics); and concordance scores measure the agreement of individual partitions to the consensus partition.

The *cola* package implements a comprehensive framework and also an easy interface for CP analysis. It allows various user-defined methods to be easily integrated into different steps of the analysis e.g. for feature selection, sample classification or definition of signatures. *Cola* provides a complete set of tools for comprehensive subgroup analysis, including partitioning, signature analysis, functional enrichment, as well as rich visualizations for interpretation of the results. Moreover, to find the method that best explains a user's dataset, *cola* allows running multiple methods simultaneously and provides functionalities for straightforward comparisons of results.

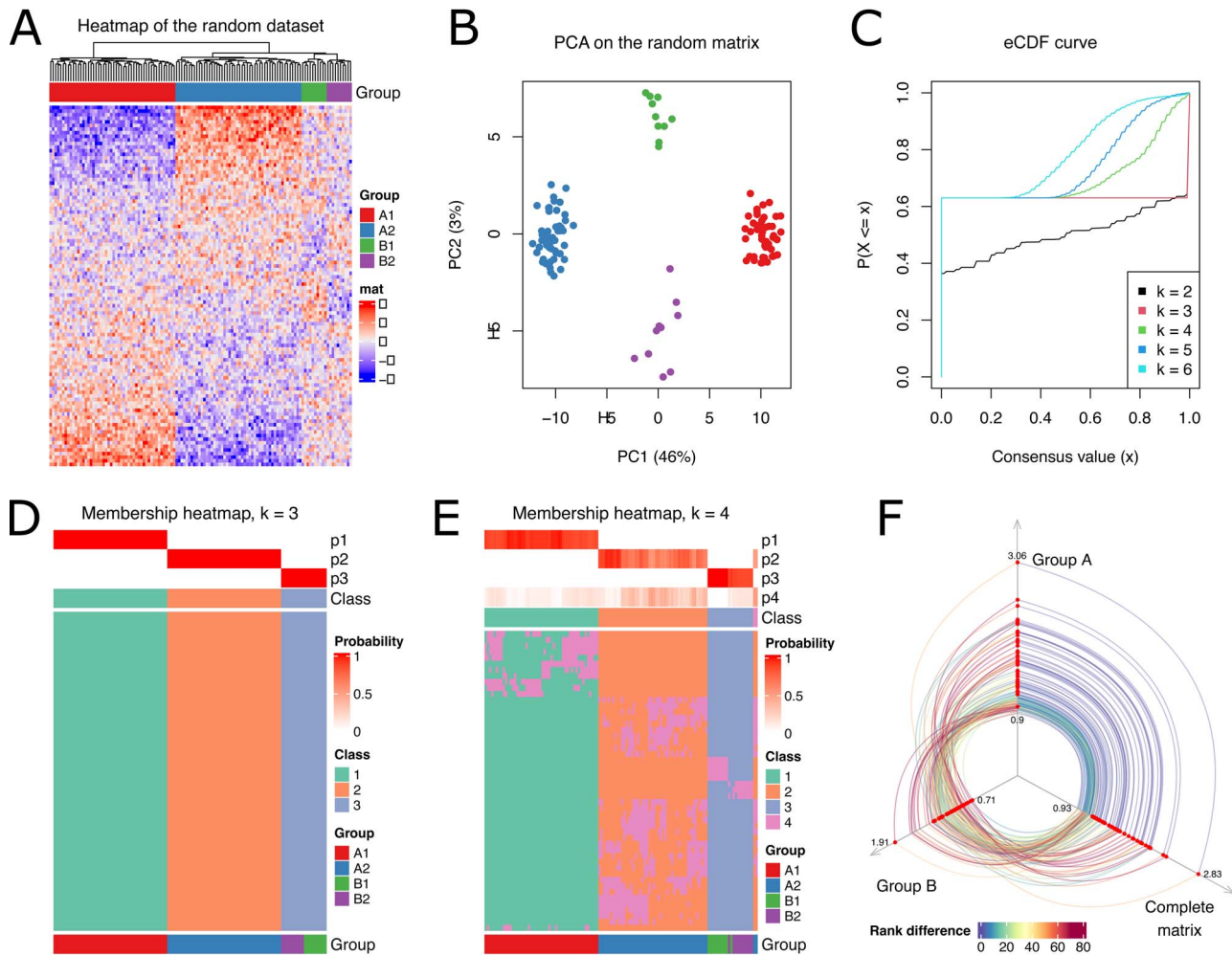
### CP methods perform weakly at simultaneously distinguishing major and secondary subgroups

We demonstrated this issue with a simulated dataset. Two random matrices denoted as  $\mathbf{M}_1$  and  $\mathbf{M}_2$  were generated with 100 columns and 20 columns, respectively. Both matrices had 100 rows.  $\mathbf{M}_1$  was simulated as a set of samples from a two-condition comparison where the first 50 columns were labeled as group 'A1' and the second 50 columns were labeled as group 'A2'. Rows in the two groups were assigned with different levels of difference. For row  $i$  in  $\mathbf{M}_1$ , values in group A1 were generated from a normal distribution  $N(\mu_{1,i}, 1)$  and values in group A2 were generated from  $N(-\mu_{1,i}, 1)$ . To simulate that rows in  $\mathbf{M}_1$  had varying differences between the two groups, the vector of mean values  $\mu_1$  was generated from the standard normal distribution  $N(0, 1)$ , thus SD of row  $i$  in  $\mathbf{M}_1$  increased when  $\mu_{1,i}$  had a higher absolute value. Similarly,  $\mathbf{M}_2$  was also simulated as a set of samples from a two-condition comparison where the first 10 columns were labeled as 'B1' and the second 10 columns were labeled as 'B2'. To simulate  $\mathbf{M}_2$  as a matrix with smaller row differences, rows in  $\mathbf{M}_2$  were generated from

$N(0.5\mu_{2,i}, 1)$  and  $N(-0.5\mu_{2,i}, 1)$  where the vector of mean values  $\mu_2$  was also generated from  $N(0, 1)$ . The order of values in  $\mu_2$  was set in a way that rank of the absolute values of  $\mu_2$  is identical to the reverse rank of the absolute values of  $\mu_1$  i.e.  $\text{rank}(|\mu_2|) \equiv \text{rank}(-|\mu_1|)$ . In this setting, if row  $i$  showed the highest difference between group A1 and A2, it showed the smallest difference between group B1 and B2.

$\mathbf{M}_1$  and  $\mathbf{M}_2$  were merged into a single matrix denoted as  $\mathbf{M}$  where A1/A2 were groups with major differences in  $\mathbf{M}$  and B1/B2 showed relatively smaller differences. Figure 1A illustrates the heatmap for the random dataset. According to the column dendrogram on the heatmap, the four groups are located in four separated branches. The separation of four groups was also confirmed by the principal component analysis (PCA) in Figure 1B. The columns were separated into three groups in the first principal component which explained 46% of the total variance of  $\mathbf{M}$ , whereas B1 and B2 were only separated in the second principal component which only explained 3% of the total variance. CP performed with *cola* was applied to  $\mathbf{M}$  where the top 50 rows with the highest SD were selected as features.  $k$ -means clustering was applied to classify samples and resampling was applied 50 times. The CP result showed that the best number of subgroups was three according to the empirical cumulative density function (eCDF) curve of the consensus values (the probability of two samples in a same subgroup) where a horizontal line extended almost from 0 to 1 for  $k=3$  (Figure 1C). This means that in the three-group classification, in the repetitive classifications by resampling from the complete feature set, any sample pair was either always in the same group (consensus value close to 1) or belonged to different groups (consensus value close to 0). This could also be confirmed by the membership heatmap with three subgroups which visualized every single partitioning result where the samples in all 50 partitionings almost had the same classifications (Figure 1D). As a comparison, when the number of subgroups was set to four, the membership heatmap showed that in  $\sim 90\%$  of the individual partitionings, either group1 or group2 was further split into two smaller subgroups, whereas only in 10% of all partitionings, B1 and B2 were correctly separated. This misclassification resulted in CP failing to assign B1 and B2 as stable classifications (Figure 1E), and thus it rejected four as the optimal number of subgroups. This problem was mainly due to the fact that almost all the top 50 rows with the highest SD from  $\mathbf{M}$  also had the highest SD in  $\mathbf{M}_1$ , thus efficient to separate A1 and A2 as well; while these 50 rows had very small SD if only counting  $\mathbf{M}_2$ , which in turn implied that these features were not good at separating B1 and B2 (Figure 1F).

Of note, here we constructed a simulated dataset with only four subgroups. In real-world datasets, if the expected number of subgroups is large, it is highly possible that hierarchical structures exist and secondary subgroups would be hidden if applying standard CP to



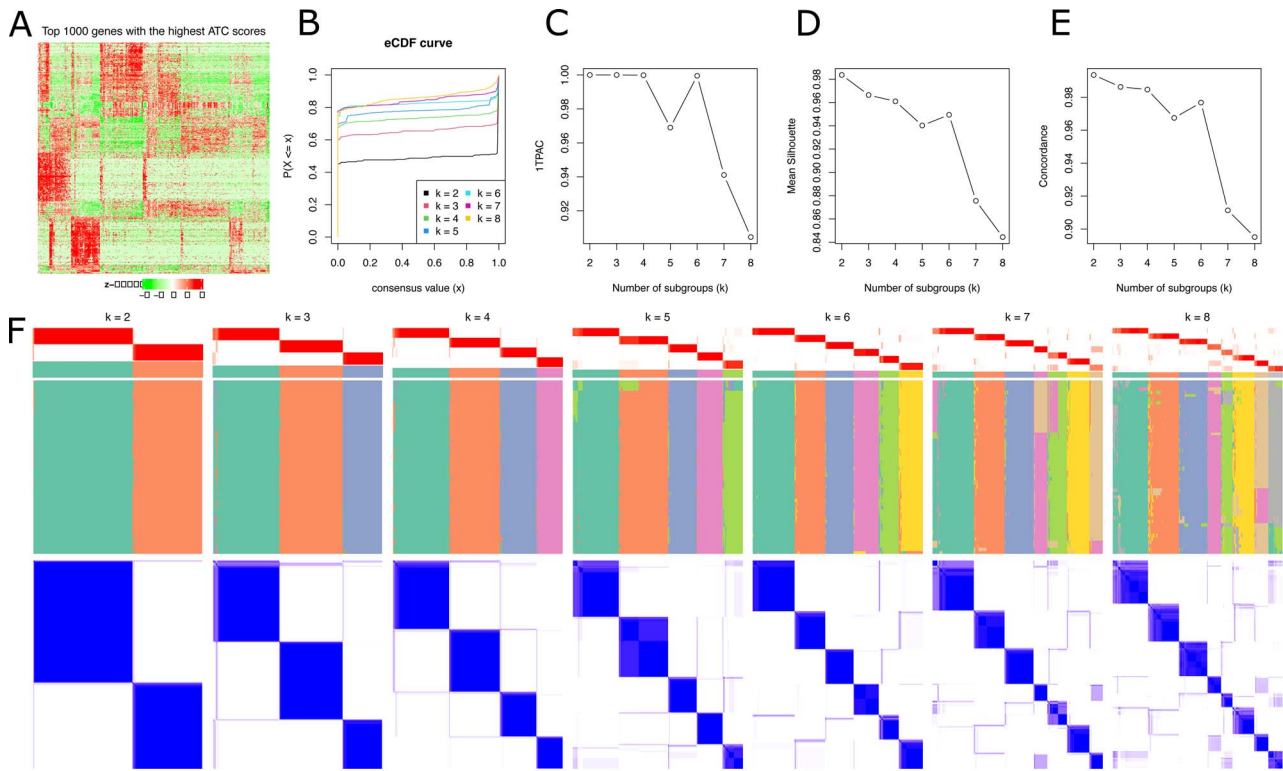
**Figure 1.** CP methods perform weakly at simultaneously distinguishing major and secondary subgroups. (A) Heatmap of the simulated dataset. (B) PCA of the simulated dataset. (C) Empirical CDF curves of the CP for each  $k$ . (D) Membership heatmap of the CP with  $k=3$ . Columns are samples and rows are individual partitions. (E) Membership heatmap of the CP with  $k=4$ . In D and E, top annotations with names p1–p4 correspond to the probability of samples belonging to each subgroup. Both membership heatmaps contain 50 individual partitionings on rows. (F) Comparison of the top 50 features in different groups. These 50 features are selected by highest SD calculated from the complete simulated matrix. The three axes correspond to SD values calculated in matrices of group A, group B and the complete matrix. The top 50 features are highlighted in red dots in the three axes. The same features are connected between axes and the lines are colored by the rank difference of the SD values in the two corresponding matrices. The numbers on the three axes represent ranges of SD values.

all samples. Thus, a method to capture the hierarchical structure of data is needed. In [Supplementary File 1](#), we demonstrate that HCP is able to detect all four groups of this random dataset.

### CPs are less stable for larger $k$

The second issue for standard CP procedures is that when the expected number of subgroups increases, the probability of two samples being in different subgroups tends to increase as well, which results in the decrease of the classification stability for larger  $k$ . Here, we demonstrate this issue with the human skeletal muscle myoblasts (HSMMs) scRNAseq dataset [12], on which we applied CP with ATC as top-value method and skmeans as partitioning method. The number of subgroups was iterated from 2 to 8. The heatmap of the top 1000 genes with the highest ATC scores suggested there should be a large number of subgroups (e.g.  $> 10$ ) with clear

patterns ([Figure 2A](#)). However, the optimal number of subgroups selected by CP only reached a small value. [Figure 2B–E](#) illustrates the selection of the best  $k$  by the eCDF curves and three metrics 1-PAC scores, mean silhouette scores and concordance scores. CPs showed high stability with  $k$  between 2 and 6 where 1-PAC scores, mean silhouette scores and concordance scores were very close to 1, although mean silhouette scores and concordance scores decreased slightly when  $k$  increased. When  $k$  exceeded 6, scores of 1-PAC, mean silhouette and concordance dropped dramatically compared with previous  $k$ . If the selection of the best  $k$  was based on the maximal votes from the three metrics according to their highest values,  $k=2$  was taken as the best result, and if robustness was taken into consideration,  $k=6$  might be the best result. Nevertheless, the two  $k$  (2 and 6) were still far away from the expected optimal  $k$ , which could not be identified with high stability in



**Figure 2.** Illustration of the issue of large  $k$  in CP with the HSMM scRNAseq dataset. (A) Heatmap of the top 1000 genes with the highest ATC scores. (B) eCDF curves. (C) 1-PAC scores versus  $k$ . (D) Mean silhouette scores versus  $k$ . (E) Concordance scores versus  $k$ . (F) Integrated visualization of the clustering results. Each column corresponds to the results for a specific  $k$ . For each  $k$ , from the top to bottom, there are the following plots: (i) an annotation showing the probability of samples in each subgroup; (ii) a one-row annotation showing the consensus classification; (iii) a membership heatmap that visualizes all 50 individual partitionings; (iv) the consensus heatmap that visualizes the probability of every pair of samples to be in the same subgroup. For each  $k$ , columns have the same orders for all plots.

the framework of CP. The decrease in stability could also be observed in Figure 2F which directly visualizes the membership of every individual partition (membership heatmap) and the probability of every two samples in the same subgroup (consensus heatmap). It shows that when  $k$  becomes larger (e.g.  $k=7$  and  $8$ ), samples tend to show more ambiguous classifications in individual partitionings.

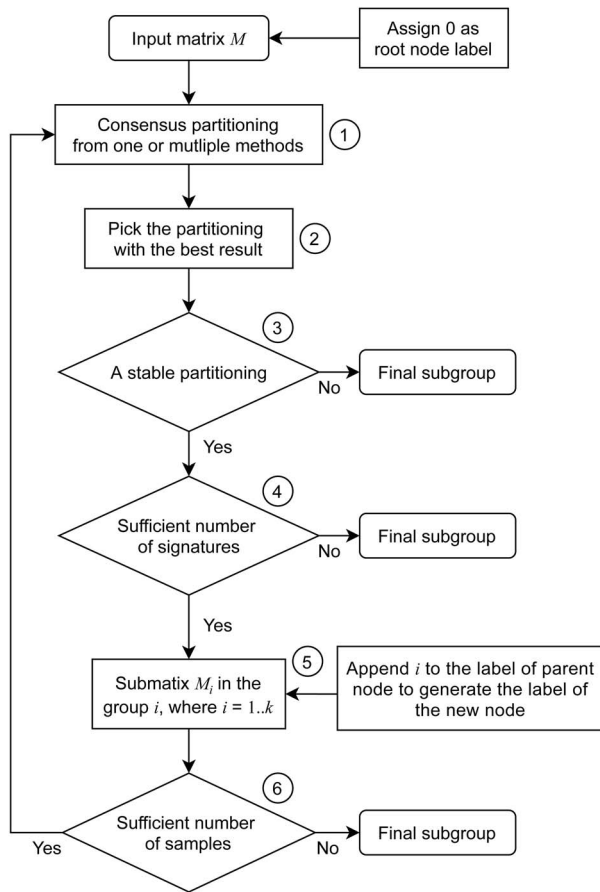
This example implies CP has a preference to assign high stability to small  $k$ , whereas it is difficult for large  $k$  to reach stability. This issue can be solved by applying CP with a hierarchical procedure where at each step of the iteration, CP is only applied with small  $k$  that ensures the gain of stability, and more subgroups are found in later steps.

### Workflow for HCP

We propose a method named HCP to perform CP via a hierarchical procedure. The HCP procedure is illustrated in Figure 3. The complete matrix is taken as the input of the whole analysis and the full set of samples is taken as the root node labeled as '0' in the hierarchy. CP with a single combination of a top-value method and a partitioning method or multiple combinations of both methods are applied to the matrix with a list of  $k$  (Figure 3, step 1) and the best  $k$  from the best method is selected (Figure 3, step 2). If there are multiple methods

showing stable partitionings, the one with the highest number of signatures is selected. Note  $k$  should be set up to a small value. After the best partitioning is selected, next there are two filters to decide whether the samples should be split according to the classification. First the stability measured by mean silhouette score is tested against a cutoff (Figure 3, step 3). If the best partitioning is not stable, the whole set of samples are treated as unclassifiable and the node is treated as a leaf in the hierarchy. If it is stable, next the biological meaningfulness of the classification is tested by the number of signatures in the classification where signatures are the features showing significant differences between subgroups (Figure 3, step 4). If the number of signatures is sufficient, the partitioning result is accepted and subgroups are taken as its child nodes (Figure 3, step 5). For each child node, the number of samples of the corresponding submatrix is then tested. HCP is iteratively applied to each child node only if there are enough samples (Figure 3, step 6). In HCP, these filters can also be applied when the subgroup hierarchy is completely generated and users can manually fine-tune the hierarchy by merging or further splitting the nodes.

In HCP, node labels are encoded as a list of digits. The number of digits corresponds to the depth of the node in the hierarchy and the value of the digits corresponds to the subgroup index on the node e.g. a label of '012' means the node is the second subgroup of the partitioning that



**Figure 3.** Workflow for HCP. The steps are as follows: (a) Apply CP to the data matrix with one combination of top-value method and partitioning method or multiple combinations of methods. (b) Pick the partitioning that shows the best result. (c) Test whether the best partitioning is stable. (d) Test whether the best partitioning generates a sufficient number of signatures. (e) If criteria in steps 3 and 4 are passed, each subgroup in the partitioning is taken as a child node in HCP. (f) For each submatrix, test whether the number of columns is sufficient. If yes, HCP is applied to the child nodes recursively.

comes from the first subgroup of the partitioning on the complete dataset.

### Automatically selecting the number of top features

In the iterative execution of CP on every submatrix, top features are firstly selected. Generally, secondary subgroups have less efficient features for classification, thus setting the same number of top features for all submatrices would bring additional noise and destabilize the classification for secondary subgroups. Therefore, a method is needed for automatic selection of proper numbers of top features on each node in HCP. This corresponds to the task of selecting a cutoff for filtering top-values. A reasonable way is to select the ‘elbow’ of the top-value curve if top-values are sorted increasingly. Here we use the method proposed in Satopaa et al. [13]. It selects the point that has the largest vertical offset to the straight line that connects the points with the minimal and the maximal top-values. More details can be found in [Supplementary File 2](#).

### CP with downsampling for large datasets

CP is by-nature a time-consuming analysis. For large datasets with huge numbers of samples, in early steps of the hierarchical procedure, numbers of samples in the subsets could still be large. To improve the efficiency of partitioning on large datasets, we propose a strategy that only applies CP to a small subset of samples that are uniformly picked from the complete set. Later class labels of deselected samples are predicted by the classification from selected samples. The prediction is based on the *signature centroid matrix*. For a selected  $k$ , signatures that significantly discriminate  $k$  subgroups are first extracted. The signature centroid matrix is defined as a  $k$ -column matrix where each column is the centroid of the confident samples i.e. those with silhouette score  $> 0.5$ , in the corresponding subgroup (here, the centroid is the mean across samples). The class prediction is performed as follows: For each deselected sample, we test which signature centroid the current sample is the closest to. For the vector denoted as  $\mathbf{x}$  which corresponds to a deselected sample, to predict the class label, the distance calculated by e.g. Euclidean, cosine or correlation methods to all  $k$  signature centroids is calculated and denoted as  $d_1, d_2, \dots, d_k$ . The class with the smallest distance is assigned to the sample:

$$\arg \min_{i \in \{1, \dots, k\}} d_i$$

Only using a small subset of samples for classification might completely miss samples in small subgroups, but they can be first attributed to the main subgroup that they are closest to and then they can be more precisely attributed in later steps of HCP.

Of note, in the vignette of the *cola* package, we additionally proposed a method that calculates  $P$ -values for the class label assignment by permuting rows of the signature centroid matrix. It provides confidence for the class label assignment; however, the  $P$ -value calculation is ignored in the process of HCP because all deselected samples are assigned to the corresponding subgroups regardless of their confidence.

Besides the centroid-based method for predicting class labels of deselected samples, *cola* additionally supports utilizing machine learning methods such as supporting vector machine (SVM) and random forest. Taking the selected samples as the training set, the class label prediction for the deselected samples is based on the signature matrix where features show significant differences between subgroups in the training set. Thus, for the machine learning methods, as for the training set, it is very easy to find hyperplanes that well separate classes. On the other hand, the classification is based on a randomly sampled subset of the original data, if the features can well separate subgroups in the training set, it is very likely that features would have very similar patterns in the deselected samples. Therefore, using the centroid-based method or SVM/random forest would

give very similar classifications. More explanations and comparisons can be found in [Supplementary File 9](#).

### Comparison of different classifications

In the Results section below, we will compare HCP classifications to the ones from the respective original studies. To this end, here we define a similarity measure for pairwise comparisons of classifications. For two classifications denoted as  $C_1$  and  $C_2$  with number of subgroups  $n_1$  and  $n_2$ , respectively, let  $g_i$  and  $h_j$  be the  $i$ th group in  $C_1$  and the  $j$ th group in  $C_2$ . Since different classification methods can hardly generate consistent classifications for all subgroups, especially when  $n_1$  and  $n_2$  are large, we define  $g_i$  to completely agree with  $h_j$  if  $g_i$  is included in  $h_j$  i.e.  $g_i \in h_j$ , thus the overlap coefficient is used to measure the similarity of subgroups of two classifications. Assume  $A$  is the set of samples in  $g_i$  and  $B$  is the set of samples in  $h_j$ , the overlap coefficient denoted as  $a_{ij}$  is defined as:

$$a_{ij} = \frac{|A \cap B|}{\min(|A|, |B|)}$$

where  $|A|$  and  $|B|$  are the numbers of samples in the two sets. The agreement of  $g_i$  with  $C_2$  is calculated as the maximal overlap coefficient across all subgroups in  $C_2$ :

$$s_{i,C_2} = \max_j a_{ij}.$$

Then the overall similarity of  $C_1$  to  $C_2$  is calculated as the mean of the agreement of each subgroup to  $C_2$  weighted by the subgroup size:

$$s_{C_1,C_2} = \frac{\sum_i^{n_1} k_i s_{i,C_2}}{\sum_i^{n_1} k_i}$$

where  $k_i$  is the size of  $g_i$ . We name it the *overall classification agreement* in the paper. The definition of the overall classification agreement is not exactly symmetric i.e.  $s_{C_1,C_2} \neq s_{C_2,C_1}$ , but the two values are very similar. A detailed explanation of the similarity measure can be found in [Supplementary File 7](#).

### Implementation of HCP in cola

HCP has been integrated in *cola* from version 2.0.0 with an object-oriented implementation. The main function `hierarchical_partition()` performs the analysis and returns a `HierarchicalPartition` object. *Cola* provides rich visualization utilities on the object and we aimed at implementing the application programming interface for the functions compatible with those in standard *cola* analysis so that it is seamless to switch analysis methods. To name a few: `collect_classes()` draws the hierarchy of the classification; `get_signatures()` calculates and visualizes the rows that are significantly different between subgroups; `dimension_reduction()` performs dimension reduction analysis to visualize how well the subgroups are separated; `top_rows_overlap()` compares the top features

on nodes since submatrices on different nodes may have different sets of top features and `functional_enrichment()` automatically applies function enrichment on the signatures if they can be mapped to genes. Example figures can be found in [Figure 4](#).

Similar to standard CP analysis in *cola*, there is also a `cola_report()` function that is applied on the `HierarchicalPartition` object. It automatically performs the complete analysis and generates all the tables and plots in an HTML report. Thus, to perform a HCP analysis, users only need a minimal set of code of using two functions, such as:

```
rh = hierarchical_partition(matrix, ...)
cola_report(rh, ...)
```

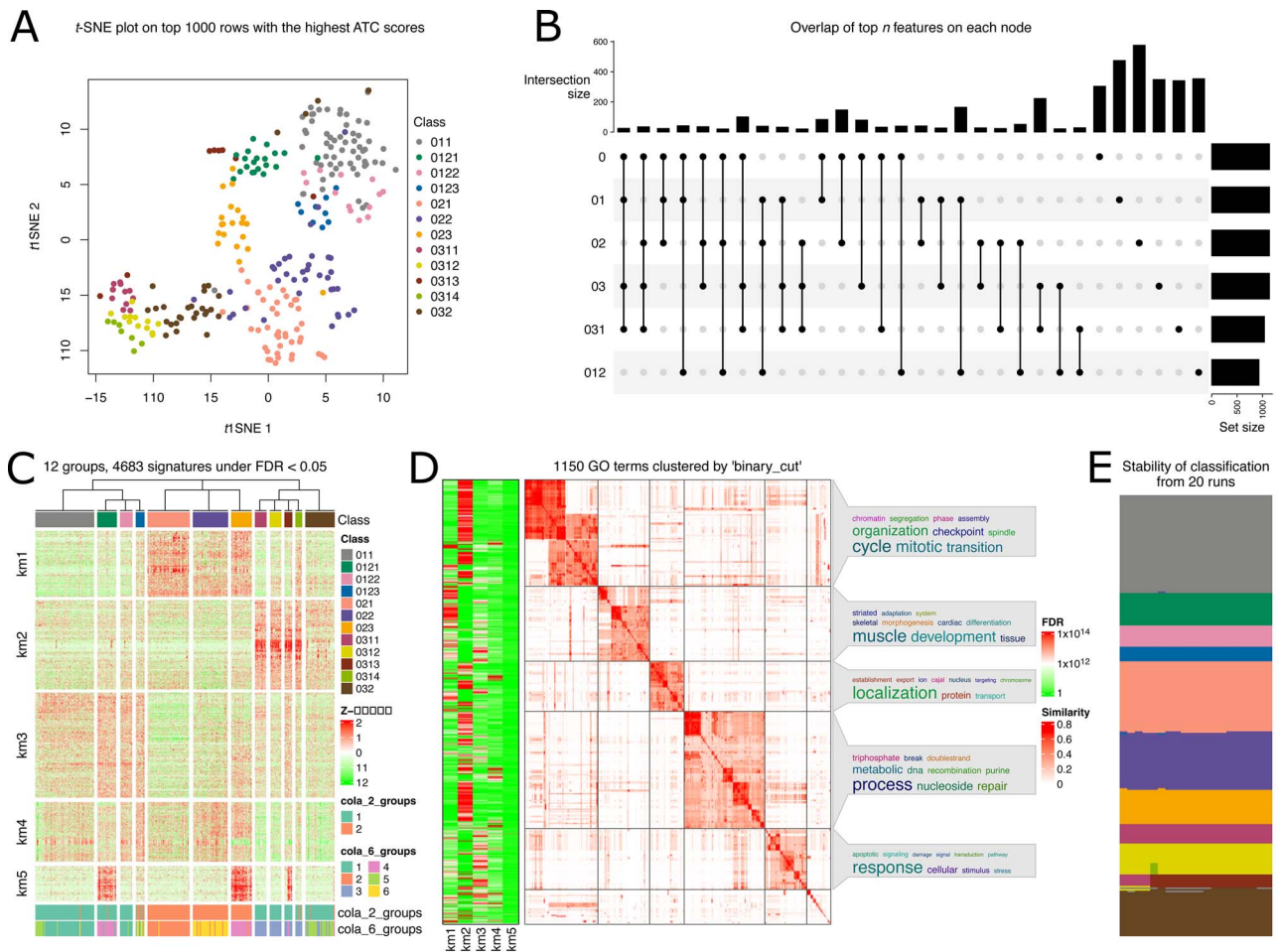
In [Figure 3](#), steps 3, 4 and 6 validate whether HCP should continue on the current node. The validation can also be applied after the classification hierarchy is completely generated. In most functions, users can control what level in the hierarchy they want by adjusting the number of samples or the number of signatures. Also the subgroup hierarchy can be manually merged by `merge_node()` and extended by `split_node()` functions on specific nodes.

## Results

### Comparison to standard CP – a case study

In [Figure 2](#), we demonstrated that CP was difficult to simultaneously identify a large number of subgroups with the HSMM dataset [12]. Here we applied HCP to the same dataset. For partitioning at the level of every node in the subgroup hierarchy, ATC was chosen as the top-value method and skmeans was chosen as the partitioning method.

HCP identified 12 subgroups which were well separated in the t-distributed stochastic neighbor embedding (t-SNE) plot ([Figure 4A](#)); for a comparison, CP only suggested 6 as the optimal number of subgroups ([Figure 2](#)). An overlap of the top features extracted on non-leaf nodes in the subgroup hierarchy showed each subset of samples had its own specific features ([Figure 4B](#)). For example, on node '02', 50.1% out of the 1139 top features were unique and not present in any of the other nodes ([Figure 4B](#)). The signature heatmap ([Figure 4C](#)) which included genes with significant differences in at least two subgroups showed that the 12 subgroups were well separated and had distinct patterns. The functional enrichment on the signature genes also showed they were biologically meaningful e.g. genes in row group 'km1' were enriched with functions related to muscle cell development and genes in row group 'km2' were enriched with functions of cell cycle and metabolic process ([Figure 4D](#)). HCP involves multiple executions of CP on nodes with random resampling. [Figure 4E](#) demonstrates that random sampling had almost no influence on the classification of the HSMM dataset, where HCP was repeatedly executed 20 times and the classification showed high stability with 97.9%



**Figure 4.** Application of HCP to the HSMM dataset. **(A)** t-SNE plot of the top 1000 genes with the highest ATC scores. The plot was made by the function *dimension\_reduction()*. **(B)** UpSet plot showing the overlap of top features selected on non-leaf nodes by HCP. Only combinations with sizes larger than 20 are included in the plot. The plot was made by the function *top\_rows\_overlap()*. **(C)** Heatmap of the signature genes according to HCP classification. The dendrogram on top of heatmap corresponds to the hierarchy of the HCP classification. The bottom annotation contains CP classifications with two and six subgroups. Rows were partitioned by k-means clustering into five clusters. The plot was made by the function *get\_signatures()*. **(D)** Gene ontology (GO) enrichment of the signature genes. The left heatmap visualizes the FDRs from the enrichment analysis on genes from each 'km' group. The right heatmap visualizes the similarity of GO terms where the GO terms are clustered by their similarities. The word clouds contain overrepresented keywords from the significant GO terms. The analysis was performed by the function *functional\_enrichment()* and visualized with the *simplifyEnrichment* package [14]. **(E)** Stability of the HCP classification from 20 repetitive runs.

concordance (measured as the average percent of samples having the same classifications).

### Application to a large scRNAseq dataset

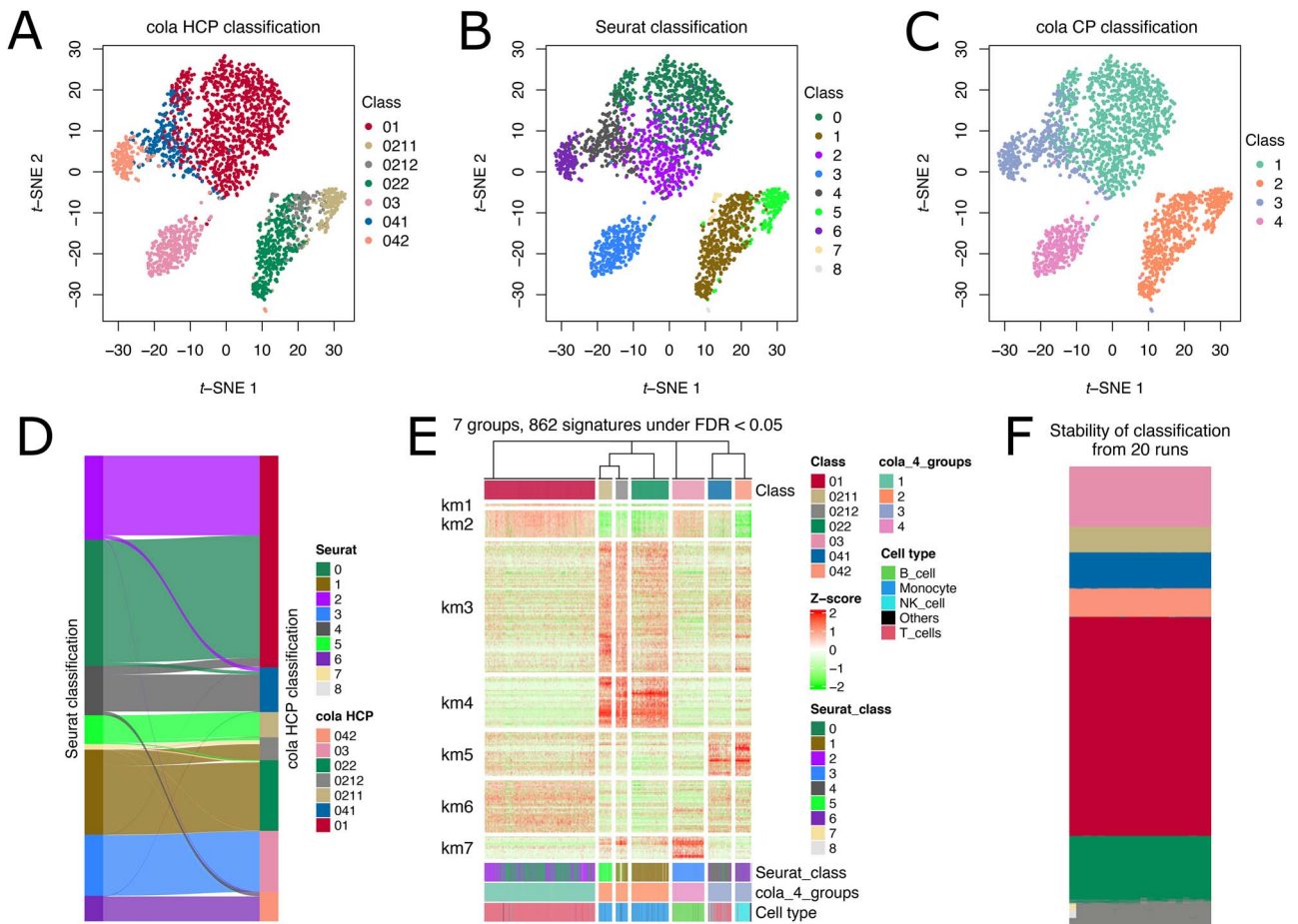
We applied HCP on the peripheral blood mononuclear cells (PBMCs) scRNASeq dataset with 2638 cells [15]. Whenever the subset corresponding to a node had more than 500 cells, downsampling was turned on to only randomly pick 500 cells. ATC was used as the top-value method and skmeans was used as the partitioning method. We compared HCP classification to CP classification as well as the original classification analyzed with the *Seurat* package [15, 16]. A table of class labels is provided in [Supplementary File 8](#).

HCP identified seven subgroups ([Figure 5A](#)), whereas CP only identified four subgroups ([Figure 5C](#)) as the optimal result, where group '2' in CP was additionally classified into three subgroups in HCP labeled '0211', '0212' and '022', and group '3' in CP was additionally classified into

two subgroups in HCP labeled '041' and '042'. According to the point distribution in [Figure 5C](#), there were indeed four well separated major subgroups, but HCP revealed more secondary subgroups.

When comparing the HCP classification to a classification obtained with *Seurat* [16], most samples had similar classifications, except that group '01' in HCP was split into two groups under *Seurat* ('0'/'2') and some disagreement existed between HCP '0212'/'022' and *Seurat* '1'/'7' ([Figure 5A, B](#) and [D](#)). The overall classification agreement between HCP classifications and *Seurat* is 0.947. In [Supplementary File 4](#), we demonstrate that HCP node '01' could be further split into two sub-nodes ('011'/'012') but with less stability. Classification of '011'/'012' was similar to *Seurat* '0'/'2' (80.1% of samples had the same classifications), but samples were more separated under '011'/'012' classification than by *Seurat*. We also compared the HCP group '0212'/'022' and *Seurat* group '1'/'7' in [Supplementary File 4](#). We found the two





**Figure 5.** Application of HCP to a PBMC scRNAseq dataset. (A–C) t-SNE plot of the top 1000 genes with the highest ATC scores. The colorings are based on HCP classification, *Seurat* classification and CP classification. (D) Correspondence between HCP and *Seurat* classifications. (E) Signature genes under HCP classification. Rows were partitioned by *k*-means clustering into seven clusters. (F) Stability of the HCP classifications from 20 repetitive runs.

classifications on this subset of samples generated different sets of signature genes and signature genes from the HCP classification were enriched with more significant biological functions.

Next, we annotated each cell to a cell type with the *SingleR* package [17] and cell type annotation was added to the bottom of the heatmap in Figure 5E. We found that the classification from HCP had a better agreement to cell types. *Seurat* additionally split the ‘T cell’ group into two subgroups labeled as ‘0’ and ‘2’. However, according to the heatmap in Figure 5E, cells in *Seurat* group ‘0’ and ‘2’ showed overall consistent patterns. Also, according to the heatmap of cell markers from the original *Seurat* analysis (the second last figure in [15]), cell markers had very similar expression patterns for group ‘0’ and ‘2’. Thus, we would conclude that HCP generated a better classification based on previously known cell types.

For this dataset, random sampling was additionally applied when the number of cells on a node was larger than 500, which brought a second layer of randomness. Nevertheless, Figure 5F illustrates that the classification was stable among 20 repetitive HCP runs, which implies that if there exists a clear classification, downsampling won’t bring significant noise from randomization

(concordance of the classifications from 20 repetitive runs was 99.1%).

### Application to a methylation dataset with a large number of subgroups

We applied HCP to a DNA methylation array dataset of central nervous system tumors (CNSTs) with 2803 samples [18]. The dataset contains 14 different tumor types (including controls) which were additionally classified into 91 subgroups based on methylation profiles (all samples had tumor cell content  $\geq 70\%$ ). The correspondence between tumor types and methylation classes is illustrated in Supplementary File 5). The classification in the original study had been performed in a semi-supervised way where tumor types were predefined and unsupervised clustering was applied only within each of the tumor types. The aim of this analysis here is to see whether HCP can recover such a great number of subgroups in a completely unsupervised way. In the analysis, we only considered CpG probes located in CpG islands. One reason was to reduce the dataset; another reason was that we have demonstrated it might be more proper to analyze CpGs for different CpG features (i.e.

CpG islands, shores and seas) separately since they might generate different classifications and correspond to different biological meanings [3]. The final matrix for HCP analysis contained 117 976 CpG probes. In the analysis, the configurations were as follows: for CP executed on each node, two top-value methods (SD and ATC) and two partitioning methods (kmeans and skmeans) were tested because SD/kmeans are popular in current studies of methylation data analysis and we demonstrated ATC/skmeans performed better for identifying subgroups [3], thus the four combinations of methods for CP were used and HCP automatically picked the one with the best result on each node. Matrix rows were not scaled because methylation data were all in the same scale i.e. [0, 1]. For every submatrix, a filtering step was applied ahead of CP which only took the top 30 000 probes with the highest SD values. Later, the top 1000 features extracted by SD/ATC from those 30 000 probes were used for partitioning. To prevent ATC from extracting rows showing high correlation but with small absolute differences which might come from systematic batches, the difference of methylation between subgroups for signature probes was required to be  $>0.25$  for every selected row. And finally, the minimal number of signature probes was set to  $>1000$ . When the number of samples in a submatrix was larger than 500, downsampling was applied. A table of class labels is provided in [Supplementary File 8](#).

HCP identified in total 92 subgroups. In the original study, samples were classified into 14 different tumor types. [Figure 6A](#) demonstrated the agreement between tumor types and HCP classification. It shows that in most cases, samples in the same HCP subgroup always belonged to a single tumor type, with very few exceptions. The overall classification agreement for HCP classification to tumor types was 0.903. Only 15/92 (16.3%) HCP subgroups (covering 15.2% of all samples) were less compatible with tumor types as defined by an overlap coefficient  $<0.75$  ([Figure 6C](#)) where overlap coefficient measured similarities between tumor types and HCP classification.

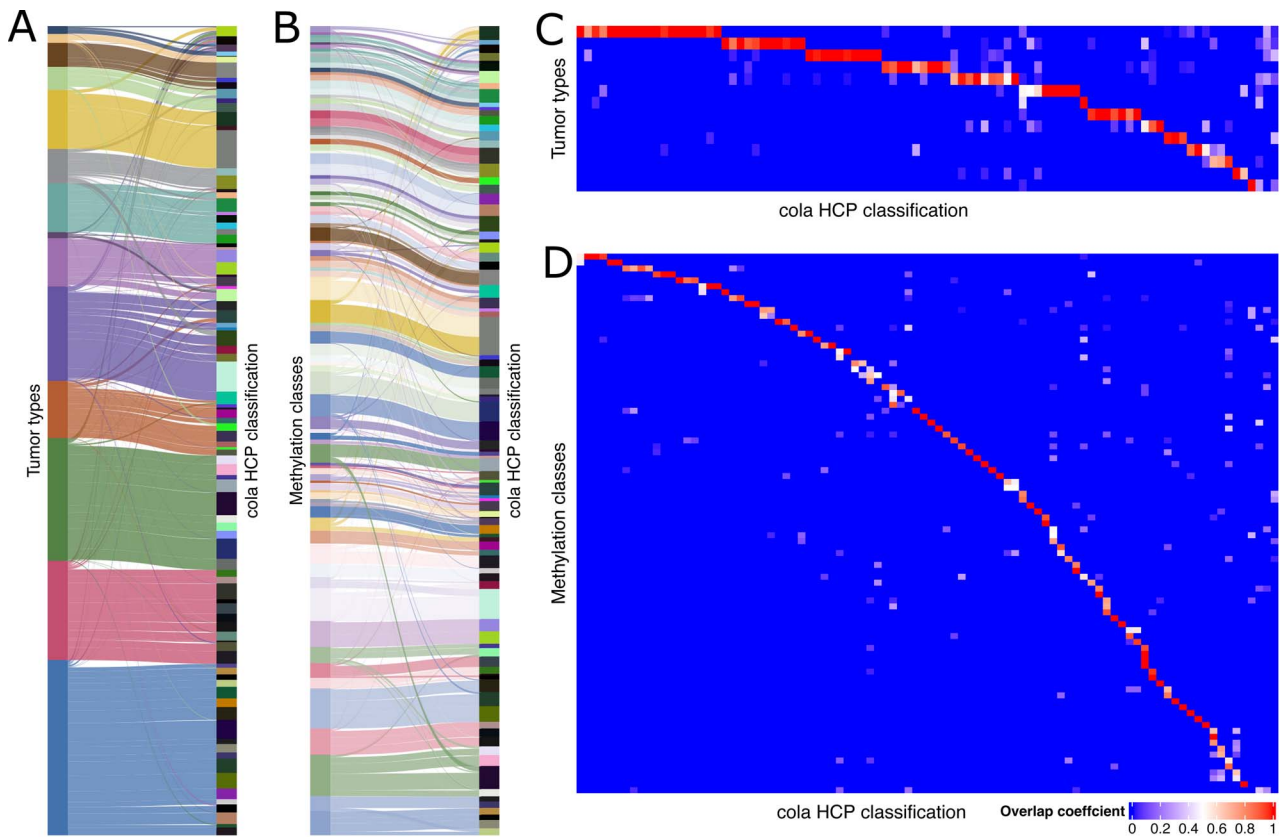
The 14 tumor types had additionally been classified into 91 subgroups (methylation classes) based on methylation profile in the original study (also see [Supplementary File 5](#)). [Figure 6B](#) illustrates the correspondence between HCP classification and methylation classes. The overall classification agreement was 0.868 and only 17/92 (18.5%) HCP subgroups (covering 16.5% of all samples) were less compatible with methylation classes as defined by overlap coefficient  $<0.75$  ([Figure 6D](#)) where overlap coefficient measured similarities between methylation classes and HCP classification. Twenty-nine HCP subgroups had a one-to-one mapping to methylation classes, 11 methylation classes covered multiple HCP subgroups and 6 HCP subgroups covered multiple methylation classes, with overlap coefficient  $>0.75$  ([Figure 6D](#)).

## Discussion

With cohort studies increasing rapidly in size, new possibilities for detecting more subtle subgroups that have specific patterns arise. In cancer studies, tumor subtypes were frequently studied [19, 20]. However, due to the complex mechanisms of tumor generation and growth e.g. tumor microenvironment, inter/intracell interactions, spatial as well as temporal patterns, tumors have different molecular profiles in the respective scenarios. It is thus important to identify these subtypes with high specificity, which also contributes to more precise diagnosis for tumors [21, 22]. Single cell technologies now allow researchers to simultaneously measure a great number of cells, which also makes it possible to detect the hierarchy of various cell types and organs [23]. CP methods, although they have been successfully applied to reveal tumor subtypes and cell types, are still weak at identifying more subtypes which show more specific and subtle differences. A classification with hierarchical methods can solve this problem [8]. In this work, we proposed a new method named HCP which applies CP in a hierarchical procedure. It combines the advantages of evaluating the stability of a classification as well as revealing subgroups with multi-level differences. We applied HCP on real-world datasets and it showed HCP efficiently revealed more subgroups and had more meaningful classifications compared with those described in the original studies. In [Supplementary File 3](#), we additionally demonstrated the use of HCP on 66 real-world datasets including scRNAseq data and methylation data.

HCP has the limitation that misclassifications in early steps in the hierarchical process would accumulate and affect the downstream classifications. In early steps of HCP, when the real number of subgroups is higher than the number of subgroups tried on the node, subgroups with relatively similar patterns would be merged into larger ones and they will be separated later. If a subgroup locates between two major subgroups with a less consistent pattern, it is possible that the subgroup would be split into two parts and each part is partially assigned to a major subgroup. Once they are separated, they cannot be merged back in later steps of HCP. One solution is to increase the maximal  $k$  tried on each node, and the other solution is to try multiple partitioning methods simultaneously on the node where the partitionings with misclassification tend to have less stability and they can be automatically filtered out.

CP is a multi-step analysis where selection of parameters on each step might affect the final partitioning result. HCP involves a list of executions of CP on nodes of the subgroup hierarchy, thus CP should be applied in a way that parameters are optimally selected to ensure that subgroups are successfully separated on each node and that separation can be extended sufficiently well in the downstream part of the hierarchy. In HCP, several



**Figure 6.** Comparison of classifications on the CNST dataset. (A) Comparison of classifications between tumor types and HCP classification. (B) Comparison of classifications between methylation classes and HCP classification. (C) Overlap coefficient between tumor types and HCP classification. Row and column orders in C are the same as in A. (D) Overlap coefficient between methylation class and HCP classifications. Row and column orders in D are the same as in B.

strategies were implemented. The number of top features for each submatrix was automatically selected, to exclude those features that might bring additional noise to classification for secondary subgroups. Also, instead of using 1-PAC score as in standard CP analysis, the mean silhouette score was used to validate the stability of a classification because silhouette scores provide a stricter measure for the stability. Furthermore, silhouette scores lead to selection of smaller but more stable  $k$ , and more subgroups can be found in the later steps of the hierarchical procedure.

Highly heterogeneous data might result in many iterations in the hierarchical process and generate large numbers of subgroups. As demonstrated in the analysis of the Glioblastoma (GBM) microarray dataset from the Cancer Genome Atlas (TCGA) [24] (Supplementary Files 3 and 6), standard CP generated 4 subgroups as the best result, whereas HCP generated 16 subgroups where the mean subgroup size was only 11. Although on one hand this could be an example of showing standard CP not being able to detect a sufficiently large number of subgroups, on the other hand this could also be reflective of HCP over classifying samples and losing generality of the classification. In that example, the separation of the 16 subgroups by HCP was biologically reasonable and large numbers of signature genes ( $> 600$  under false discovery rate (FDR)  $< 0.05$ ) supported the classification

on the corresponding nodes, whereas a too specific classification would have increased the difficulty to interpret the results and to extend to different studies based on the same biological entities. To balance the generality and specificity of the classification which is a choice based on the level of heterogeneity users expect, a simple solution is to set a minimal number of samples on nodes; another solution is to filter the generated hierarchy by the number of signatures on each node, which is an indication of the biological importance of the classification; generally, the number of signatures decreases when subgroups have smaller separation. To help users to adjust the level of heterogeneity of classifications, HCP in *cola* allows to extract and analyze the classification at a self-specified level of subgroup hierarchy.

For large numbers of samples, we proposed to apply CP only to a subset of randomly sampled samples. Later class labels of the deselected samples were predicted based on the classification of selected samples. If major subgroups are well separated, downsampling CP tends to retain the classification of major groups, whereas if classification is not stable for the complete set of samples, neither is classified on randomly selected subsets. Thus, downsampling is a good strategy for data with clear structures for reducing running time. For heterogeneous datasets with large numbers of subgroups, downsampling might completely deselect samples in small

subgroups; however, they can be attached to major subgroups in the class label prediction step and classified in later steps of HCP. Therefore, downsampling CP is a good companion of HCP.

## Conclusion

In this work, we proposed a new method that extends CP via a hierarchical procedure. It can reveal more subgroups with various levels of differences which are normally difficult to detect by standard CP methods. We demonstrated the usage of HCP with real-world examples. We also demonstrated its ability to identify large numbers of subgroups with high agreement to the original studies. The method has been implemented as an extension of the previously published *cola* package. The functionality of HCP was designed to provide both ease and comprehensiveness. We believe it will be a convenient and powerful tool for users to dive deeper into their data and to reveal more structures.

### Key Points

- CP is widely used in high-throughput data analysis to reveal subgroups, but it is weak at identifying large numbers of stable subgroups with various levels of differences.
- We proposed a new method named HCP that applies CP in a hierarchical procedure. It can distinguish subgroups with different levels of differences and reveal more subgroups with more specific patterns.
- We benchmarked HCP on real-world datasets and it showed HCP efficiently revealed more subgroups and generated more meaningful classifications compared with current ones from original studies.
- The HCP method has been implemented as an extension of the previously published R package *cola*; it provides an easy-to-use user interface and it still keeps the comprehensiveness of the analysis. HCP can automate the analysis only with a minimum of two lines of code, which generates a detailed HTML report containing the complete analysis.

## Supplementary Data

Supplementary data are available online at <https://academic.oup.com/bib>.

## Funding

National Center for Tumor Disease (NCT) Molecular Precision Oncology Program)

## Data Availability

The HSMM single-cell RNA-Seq dataset is available in the *HSMMSingleCell* Bioconductor package [12]. Expression values were normalized by  $\log_{10}(\text{FPKM}+1)$  and only

the protein-coding genes were used. The PBMC dataset and the *Seurat* classification were obtained according to the tutorial of the *Seurat* package ([https://satijalab.org/seurat/articles/pbmc3k\\_tutorial.html](https://satijalab.org/seurat/articles/pbmc3k_tutorial.html)). The CNST dataset [18] was downloaded from the GEO database with accession ID GSE90496. The source and processing of the 66 test datasets can be found in [Supplementary File 3](#) with runnable code.

HCP has been implemented in the *cola* package from version 2.0.0 (<https://bioconductor.org/packages/cola/>). The website of HCP is at [https://github.com/jokergoo/cola\\_hcp](https://github.com/jokergoo/cola_hcp). The HTML reports for 66 test datasets are publicly available at <https://cola-rh.github.io/>. The scripts to perform the complete analysis are available at <https://www.github.com/cola-rh/manuscript>. The supplementary files are also available at <https://cola-rh.github.io/supplementary/>.

## References

1. Monti S, Tamayo P, Mesirov J, et al. Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Mach Learn* 2003;**52**:91–118.
2. Sturm D, Witt H, Hovestadt V, et al. Hotspot mutations in H3F3A and IDH1 define distinct epigenetic and biological subgroups of glioblastoma. *Cancer Cell* 2012;**22**:425–37.
3. Gu Z, Schlesner M, Hübschmann D. Cola: an R/Bioconductor package for consensus partitioning through a general framework. *Nucleic Acids Res* 2021;**49**:e15.
4. Hornik K, Feinerer I, Kober M, et al. Spherical k-means clustering. *J Stat Softw* 2012;**50**:1–22.
5. Wilkerson MD, Hayes DN. ConsensusClusterPlus: a class discovery tool with confidence assessments and item tracking. *Bioinformatics* 2010;**26**:1572–3.
6. Kiselev VY, Kirschner K, Schaub MT, et al. SC3: consensus clustering of single-cell RNA-seq data. *Nat Methods* 2017;**14**:483–6.
7. Jeub LGS, Sporns O, Fortunato S. Multiresolution consensus clustering in networks. *Sci Rep* 2018;**8**:3259.
8. Silla CN, Freitas AA. A survey of hierarchical classification across different application domains. *Data Min Knowl Discov* 2011;**22**:31–72.
9. Babbar R, Partalas I, Gaussier E, et al. On flat versus hierarchical classification in large-scale taxonomies. In: Burges CJC, Bottou L, Welling M, et al. (ed.) *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'13)*. Red Hook, NY: Curran Associates Inc., 2013;1824–32.
10. Feng S, Fu P, Zheng W. A hierarchical multi-label classification method based on neural networks for gene function prediction. *Biotechnol Biotechnol Equip* 2018;**32**:1613–21.
11. Şenbabaoğlu Y, Michailidis G, Li JZ. Critical limitations of consensus clustering in class discovery. *Sci Rep* 2014;**4**:6207.
12. Trapnell C. HSMMSingleCell: single-cell RNA-Seq for differentiating human skeletal muscle myoblasts (HSMM). *Bioconductor* 2021. <https://bioconductor.org/packages/HSMMSingleCell/>.
13. Satopaa V, Albrecht J, Irwin D, et al. Finding a “kneedle” in a haystack: detecting knee points in system behavior. *31st International Conference on Distributed Computing Systems Workshops IEEE Computer Society, USA, 2011*, 166–71.
14. Gu Z, Hübschmann D. simplifyEnrichment: an R/Bioconductor package for clustering and visualizing functional enrichment results. *BioRxiv* 2020October 28, 2020.

- <https://doi.org/10.1101/2020.10.27.312116> Arxiv biorxiv;2020.10.27.312116v3, preprint: not peer reviewed.
15. Seurat - Guided Clustering Tutorial. [https://satijalab.org/seurat/articles/pbmc3k\\_tutorial.html](https://satijalab.org/seurat/articles/pbmc3k_tutorial.html).
  16. Satija R, Farrell JA, Gennert D, et al. Spatial reconstruction of single-cell gene expression data. *Nat Biotechnol* 2015;**33**:495–502.
  17. Aran D, Looney AP, Liu L, et al. Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage. *Nat Immunol* 2019;**20**:163–72.
  18. Capper D, Jones DTW, Sill M, et al. DNA methylation-based classification of central nervous system tumours. *Nature* 2018;**555**:469–74.
  19. Ceccarelli M, Barthel FP, Malta TM, et al. Molecular profiling reveals biologically discrete subsets and pathways of progression in diffuse glioma. *Cell* 2016;**164**:550–63.
  20. Liu Y, Sethi NS, Hinoue T, et al. Comparative molecular analysis of gastrointestinal adenocarcinomas. *Cancer Cell* 2018;**33**:721–735.e8.
  21. Ogino S, Fuchs CS, Giovannucci E. How many molecular subtypes? Implications of the unique tumor principle in personalized medicine. *Expert Rev Mol Diagn* 2012;**12**:621–8.
  22. Rich JN. Cancer stem cells: understanding tumor hierarchy and heterogeneity. *Medicine* 2016;**95**:S2–7.
  23. Liberzon A, Birger C, Thorvaldsdóttir H, et al. The molecular signatures database (MSigDB) hallmark gene set collection. *Cell Syst* 2015;**1**:417–25.
  24. Verhaak RGW, Hoadley KA, Purdom E, et al. Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in PDGFRA, IDH1, EGFR, and NF1. *Cancer Cell* 2010;**17**:98–110.