**CEPIM Guideline**

1. **Scenarios**

Here we present four scenarios of using CEPIM. For each scenario, CEPIM generates a different version report which is available at <https://github.com/GaoLabXDU/CEPIM>.

* **Scenario 1**

We don’t know any subtype information of the patients. We want to get the subtyping results from different methods, and see which result or which number of subtypes is the most appropriate for the giving data. For this purpose, we can run CEPIM simply to get the results. **CEPIM calculates the NMI and ARI between every two methods from k=2 to the *k*-max.**

* **Scenario 2**

We know some empirical pre-determination of subtypes for samples by experts or clinicians based on clinical phenotypes, images or experience. We want to know the performance of each method comparing to the pre-determination results. For this purpose, **CEPIM will take pre-determination results as the gold standards (true labels) to evaluate and compare**. We input the true labels together with multi-omics data, **CEPIM calculates the NMI and ARI between the results produced by each method and the true labels.**

* **Scenario 3**

We have the subtyping results generated by our own methods, and we want to compare with other state-of-the-art methods in CEPIM. For this purpose, we can input our results together with multi-omics data. In order to draw heatmap and calculate silhouette coefficient, we should input the integrated data matrix which can be the patient-patient similarity matrix, patient-patient distance matrix, or patient-feature matrix after integration. We should also input the time (seconds) when executing our own methods. If we cannot offer the matrices mentioned above, we can just input the subtyping results which can be taken as true labels to make comparisons as **Scenario 2** mentioned.

* **Scenario 4**

This scenario is the combination of scenario 2 and 3. That means: we know the existing subtyping information (treated as true labels) about the patients, and we have the subtyping results generated by our own methods. We want to make a comparison with other state-of-the-art methods in CEPIM according to the true labels. For this purpose, we input the true labels, our results, and the integrated result matrix together with multi-omics data.

1. **Guideline**

CEPIM is an easily and friendly-used R package to provide a comparison and evaluation platform for multi-omics data integration methods for cancer subtyping.

## **Install CEPIM**

You can download CEPIM from <https://github.com/GaoLabXDU/CEPIM> and install the package from local file and then library “CEPIM”.

Before installing CEPIM, some essential packages listed below should be installed, and you can source “init.R” to install these packages except “ggthemr”.

* CRAN: cluster, devtools, DMwR, flexclust, ggplot2, pheatmap, PINSPlus, R.matlab, rmarkdown, SNFtool, survival.
* Bioconductor: iClusterPlus
* Others
  + LRAcluster (Download from <http://bioinfo.au.tsinghua.edu.cn/member/jgu/lracluster/> or <http://lifeome.net/software/lracluster/>)
  + ggthemr (install devtools package first.)

install.packages(“devtools”)

devtools::install\_github('cttobin/ggthemr')

## **Data pre-processing**

* **Load data.** Use **formatData()** to load your data. Two types of data are the minimum requirement. Each type of data should be formatted into a matrix with columns as samples (e.g, patients) and rows as features (e.g. genes). You can use the parameter “trans” to [transpose](javascript:;) [the](javascript:;) data [matrix](javascript:;).

>mRNA <- formatData("COAD\_mRNA.csv", trans=FALSE, sep=',')

You can also load the COAD data in CEPIM.

>data("COAD\_mRNA")

* **Filter features and samples.** Use **filterData()** to filter features or samples according to the ratio of missing value to one feature or sample. If you want to treat zeros as NA, set the "filter\_zero\_as\_NA" parameter as TRUE.

> filteredmRNA <- filterData(COAD\_mRNA, alpha1 = 0.8, filter\_zero\_as\_NA= FALSE)

* **Get overlap samples.** Use **getOverlap()** to selects samples from different omics datasets that belong to the same patient cohort.

>data("COAD\_miRNA")

>datalist <- getOverlap(list(COAD\_mRNA, COAD\_miRNA))

* **Data imputation.** CEPIM provides three methods to impute the missing data: impute with the mean of each feature, impute with the median of each feature, and *k*-nearest neighbor method. If you want to impute zeros and NA, set "trans\_zero\_to\_NA" as TRUE.

> mRNA <- imputeData(COAD\_mRNA, method = 'knn', k = 10, trans\_zero\_to\_NA = FALSE)

* **Data normalization.** CEPIM provides four methods to normalize the data: minus the mean of each object, minus the median of each object, calculate z-score for each data, and log-transformation. The parameter "object" can be chosen as sample or feature.

> mRNA <- normData(mRNA, method = "z\_score", object = "sample")

## **Feature selection**

CEPIM provides three criteria to select features: based on variance, MAD, and PCA.

> mRNA <- selectFeature (mRNA, method = "MAD", select\_number = 0.9)

## **Integration, evaluation, and comparison**

There are five built-in methods in CEPIM: SNF, PFA, iClusterBayes, LRAcluster, and PINS, and you can select 1-5 methods from them to obtain cancer subtyping results. Each method has its own parameter settings, you can use default or your own settings. CEPIM needs to input *k*-max which means the possible maximum number of subtypes, then runs each method to identify two to *k*-max subtypes. **Please make sure that MATLAB has been installed on your computer if you want to run PFA.**

CEPIM provides six metrics to evaluate and compare the results generated by different methods: Kaplan-Meier survival curves and Cox p-value, Similarity heatmap, Silhouette coefficient, NMI, ARI, and Time consumption. You can choose 1-6 of them.

In order to generate Kaplan-Meier survival curves and calculate Cox p-values, clinical survival data should be provided as input and formatted as a data matrix. The names of columns are “name” (e.g. TCGA patients barcode), “A1\_OS” which means survival time, and “A2\_Event” which means sample status respectively. You can load “KMSURDATA” to see the example. For convenience, CEPIM has already collected clinical data of five cancers (BRCA, COAD, KIRC, LUAD, and LUSC) from TCGA. There is no need to set “KMdata” parameter, and CEPIM will match the TCGA sample barcodes (15 bits) of inputted data with the barcodes of clinical data.

>clinicaldata<- formatData("clinical\_data.csv", trans=FALSE, sep=',')

>CEPIM(… , KMdata = clinicaldata, …)

Here, we list some examples according to different scenarios.

> data("COAD\_cnv")

> data("COAD\_Methy")

> data("COAD\_miRNA")

> data("COAD\_mRNA")

> datalist <- list(COAD\_cnv, COAD\_Methy, COAD\_miRNA, COAD\_mRNA)

* **Scenario 1**

>CEPIM(datalist, c("gaussian", "gaussian", "gaussian", "gaussian"), functionList = list('PINS', 'LRA', 'SNF', 'PFA', 'iCluster'), evalList = list('Heatmap', 'KM', 'SI', 'ARI', 'RI', 'NMI'), kMax = 5)

* **Scenario 2**

If you have the empirical pre-determination of subtype, you can input it, and CEPIM will take it as the true lables (gold standard) to evaluate and compare the results generated by different methods and different number of subtypes. Here, we take the subtypes of COAD defined by TCGA previously as true labels.

>data("truelabel")

>CEPIM(datalist, c("gaussian", "gaussian", "gaussian", "gaussian"), functionList = list('PINS', 'LRA', 'SNF', 'PFA', 'iCluster'), evalList = list('Heatmap','KM', 'SI', 'ARI', 'RI', 'NMI'), kMax = 5, trueLabel = truelabel)

* **Scenario 3**

If you have the results generated by your own method, you can upload to CEPIM to compare with other methods. The results should be formatted as a list, including “data”, “name”, and “time”. “data” is also a list, including “cluster” and “data”. “Cluster” is a sample-size × (*k*-max-1) matrix, it shows the cluster assignments of each sample from *k*=2 to *k*-max. “Data” is a sample-sample similarity matrix or sample-feature matrix. “time” is the float number represents the running time of your own method. “name” is the name of your method. You can load “Res” to see the example. Here, we use *k*-means to cluster COAD mRNA expression data. We set K as 2 to 5 and run *k*-means respectively. Then take the results as the results of our own method. We calculate the patient-patient similarity matrix as the integrated data matrix.

>data("Res")

>CEPIM(datalist, c("gaussian", "gaussian", "gaussian", "gaussian"), functionList = list('PINS', 'LRA', 'SNF', 'PFA', 'iCluster'), evalList = list('Heatmap', 'KM', 'SI', 'ARI', 'RI', 'NMI'), kMax = 5, Res = Res)

* **Scenario 4**

We input the subtype information in scenario 2, the *k*-means results and the similarity matrix in scenario 3 together with omics data to run CEPIM.

> CEPIM(datalist, c("gaussian", "gaussian", "gaussian", "gaussian"), functionList = list('PINS', 'LRA', 'SNF', 'PFA', 'iCluster'), evalList = list('Heatmap','KM', 'SI', 'ARI', 'RI', 'NMI'), kMax = 5, trueLabel = truelabel, Res = Res)