COmbined Mapping of Multiple clUsteriNg ALgorithms (COMMUNAL): A Robust Method for Selection of Cluster Number K: R Package Vignette

Timothy E Sweeney
Stanford University

Albert Chen
Stanford University

Olivier Gevaert
Stanford University

October 9, 2015

Introduction

This vignette describes one main usage of the `COMMUNAL` algorithm. For a fuller description of the method, please see: Sweeney TE, Chen AC, Gevaert O. "COmbined Mapping of Multiple clUsteriNg ALgorithms (COMMUNAL): A Robust Method for Selection of Cluster Number K". Scientific Reports, 2015. Please also cite this paper if you use COMMUNAL in published work.

`COMMUNAL` attempts to solve a vexing problem in unsupervised learning (clustering), namely, how to choose the 'right' or 'optimal' number of clusters in a dataset. There are many methods available, but a review is outside the scope of this vignette. `COMMUNAL` has two basic functions. The first is the actual function COMMUNAL(). The COMMUNAL() function takes a data matrix as input, along with a range of K to test, and a set of clustering algorithms (such as k-means, hierarchical, etc.) and validity metrics (such as silhouette index, gap statistic, etc.). All algorithms are run on the data over the range of inputted K, and all validity metrics are applied to all algorithms. COMMUNAL() will then return the clustering results across the range of K. In general, we do not imagine most users will run the COMMUNAL() function directly.

The main algorithm which is described in the `COMMUNAL` paper actually begins with the function clusterRange(). clusterRange() is a handle that calls COMMUNAL() iteratively over progressive variable (row) subsets of data. By default, clusterRange() will sort the rows by decreasing order variance, and include the highest-variance rows first. This is modifiable by the user. The output from clusterRange() is used to locally optimize the choices of algorithms (throwing out those which choose too many very small clusters) and validity metrics (throwing out those which behave monotonically and are highly correlated). It will then use the remaining algorithms and validity metrics to produce a 3D plot of cluster optimality over the range of included variables. We suggest that this plot be used to inform a decision of which variables to use, and which K to pick.

The last part of the `COMMUNAL` package deals with assigning samples to clusters when algorithms disagree over cluster assignments. We call the resulting method the 'core' clustering. The assumption is that samples which are grouped together by multiple algorithms are truly similar, while those for which the algorithms all disagree are slightly noisy and may be outliers.

Finally, we note that the clusterRange() function can also be used to call just a single algorithm and metric (e.g., k-means with gap statistic). This would still improve a 'typical' clustering run by applying the algorithm and metric over a range of variable subsets, thus identifying stable optima for K. A 3D plot will still be produced using just one algorithm and validity metric– go on, try it!!

As an aside, we note that the `rgl` package, which is required by the 3D plotting function plotRange3D(), is somewhat finnicky. It's a fantasticly useful package and we are incredibly grateful to its makers. However, you will have to seek advice directly from them should `rgl` throw errors on your machine. We recommend trying `COMMUNAL` with a small number of simple parameters as a test run prior to throwing a massive dataset with all algorithms and all validity metrics. You can even run the code from this vignette! If so, we recommend limiting the 'varRange' function to just c(20, 40) to save yourself time.

Tutorial

Identifying k

The `clusterRange` function provides a harness to `COMMUNAL` to test progressive subsets of variables. Here we load some breast cancer data (100 gene expression levels (rows) of 533 samples (columns)). We'll use progressive subsets of 20, 40, 60, 80, and finally all 100 variables (rows).

Note that here we pick a subset of validation measures that runs quickly; to run all measures, simply input option "all"; be forewarned that this can cause delays, especially the gap statistic (since it requires bootstrapping, although this is modifiable by the user), and the measure 'g2'.

Verbose is set to T; this is an effective way to find areas of delay. Note that the output may be disordered if the parallel option is used.

```
library(COMMUNAL)

## Loading required package:  cluster
## Loading required package:  clValid
## Loading required package:  fpc

data(BRCA.100)
varRange <- seq(20,100,20)
ks <- 2:8
measures <- c("average.between", "dunn", "widestgap", "dunn2",
                      "pearsongamma", "g3", "max.diameter", "avg.silwidth")

BRCA.results <- clusterRange(dataMtx=BRCA.100, ks = ks,
                             varRange=varRange,
                             validation=measures,
                             verbose = T)

## Running COMMUNAL over range of variables...
## Calculating distance matrix...
## Clustering  hierarchical kmeans diana som sota pam clara agnes ...

## Loading required package:  kohonen
## Loading required package:  class
## Loading required package:  MASS
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"

## [1] "Finished internal validation, hierarchical 2 clusters"
## [1] "Finished internal validation, hierarchical 3 clusters"
## [1] "Finished internal validation, hierarchical 4 clusters"
## [1] "Finished internal validation, hierarchical 5 clusters"
## [1] "Finished internal validation, hierarchical 6 clusters"
## [1] "Finished internal validation, hierarchical 7 clusters"
## [1] "Finished internal validation, hierarchical 8 clusters"

## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
## [1] "Finished internal validation, kmeans 2 clusters"
## [1] "Finished internal validation, kmeans 3 clusters"
## [1] "Finished internal validation, kmeans 4 clusters"
## [1] "Finished internal validation, kmeans 5 clusters"
## [1] "Finished internal validation, kmeans 6 clusters"
## [1] "Finished internal validation, kmeans 7 clusters"
## [1] "Finished internal validation, kmeans 8 clusters"
## [1] "Finished internal validation, diana 2 clusters"
## [1] "Finished internal validation, diana 3 clusters"
## [1] "Finished internal validation, diana 4 clusters"
## [1] "Finished internal validation, diana 5 clusters"
## [1] "Finished internal validation, diana 6 clusters"
## [1] "Finished internal validation, diana 7 clusters"
## [1] "Finished internal validation, diana 8 clusters"
## [1] "Finished internal validation, som 2 clusters"
## [1] "Finished internal validation, som 3 clusters"
## [1] "Finished internal validation, som 4 clusters"
## [1] "Finished internal validation, som 5 clusters"
## [1] "Finished internal validation, som 6 clusters"
## [1] "Finished internal validation, som 7 clusters"
## [1] "Finished internal validation, som 8 clusters"
## [1] "Finished internal validation, sota 2 clusters"
## [1] "Finished internal validation, sota 3 clusters"
## [1] "Finished internal validation, sota 4 clusters"
## [1] "Finished internal validation, sota 5 clusters"
## [1] "Finished internal validation, sota 6 clusters"
## [1] "Finished internal validation, sota 7 clusters"
## [1] "Finished internal validation, sota 8 clusters"
## [1] "Finished internal validation, pam 2 clusters"
## [1] "Finished internal validation, pam 3 clusters"
## [1] "Finished internal validation, pam 4 clusters"
## [1] "Finished internal validation, pam 5 clusters"
## [1] "Finished internal validation, pam 6 clusters"
## [1] "Finished internal validation, pam 7 clusters"
## [1] "Finished internal validation, pam 8 clusters"
## [1] "Finished internal validation, clara 2 clusters"
## [1] "Finished internal validation, clara 3 clusters"
## [1] "Finished internal validation, clara 4 clusters"
## [1] "Finished internal validation, clara 5 clusters"
## [1] "Finished internal validation, clara 6 clusters"
## [1] "Finished internal validation, clara 7 clusters"
## [1] "Finished internal validation, clara 8 clusters"
## [1] "Finished internal validation, agnes 2 clusters"
## [1] "Finished internal validation, agnes 3 clusters"
## [1] "Finished internal validation, agnes 4 clusters"
## [1] "Finished internal validation, agnes 5 clusters"
## [1] "Finished internal validation, agnes 6 clusters"
## [1] "Finished internal validation, agnes 7 clusters"
## [1] "Finished internal validation, agnes 8 clusters"
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
##
## ###################
##  20  variables complete
##
## Calculating distance matrix...
## Clustering  hierarchical kmeans diana som sota pam clara agnes ...

## The "ward" method has been renamed to "ward.D"; note new "ward.D2"

## [1] "Finished internal validation, hierarchical 2 clusters"
## [1] "Finished internal validation, hierarchical 3 clusters"
## [1] "Finished internal validation, hierarchical 4 clusters"
## [1] "Finished internal validation, hierarchical 5 clusters"
## [1] "Finished internal validation, hierarchical 6 clusters"
## [1] "Finished internal validation, hierarchical 7 clusters"
## [1] "Finished internal validation, hierarchical 8 clusters"

## The "ward" method has been renamed to "ward.D"; note new "ward.D2"

## [1] "Finished internal validation, kmeans 2 clusters"
## [1] "Finished internal validation, kmeans 3 clusters"
## [1] "Finished internal validation, kmeans 4 clusters"
## [1] "Finished internal validation, kmeans 5 clusters"
## [1] "Finished internal validation, kmeans 6 clusters"
## [1] "Finished internal validation, kmeans 7 clusters"
## [1] "Finished internal validation, kmeans 8 clusters"
## [1] "Finished internal validation, diana 2 clusters"
## [1] "Finished internal validation, diana 3 clusters"
## [1] "Finished internal validation, diana 4 clusters"
## [1] "Finished internal validation, diana 5 clusters"
## [1] "Finished internal validation, diana 6 clusters"
## [1] "Finished internal validation, diana 7 clusters"
## [1] "Finished internal validation, diana 8 clusters"
## [1] "Finished internal validation, som 2 clusters"
## [1] "Finished internal validation, som 3 clusters"
## [1] "Finished internal validation, som 4 clusters"
## [1] "Finished internal validation, som 5 clusters"
## [1] "Finished internal validation, som 6 clusters"
## [1] "Finished internal validation, som 7 clusters"
## [1] "Finished internal validation, som 8 clusters"
## [1] "Finished internal validation, sota 2 clusters"
## [1] "Finished internal validation, sota 3 clusters"
## [1] "Finished internal validation, sota 4 clusters"
## [1] "Finished internal validation, sota 5 clusters"
## [1] "Finished internal validation, sota 6 clusters"
## [1] "Finished internal validation, sota 7 clusters"
## [1] "Finished internal validation, sota 8 clusters"
## [1] "Finished internal validation, pam 2 clusters"
## [1] "Finished internal validation, pam 3 clusters"
## [1] "Finished internal validation, pam 4 clusters"
## [1] "Finished internal validation, pam 5 clusters"
## [1] "Finished internal validation, pam 6 clusters"
## [1] "Finished internal validation, pam 7 clusters"
## [1] "Finished internal validation, pam 8 clusters"
## [1] "Finished internal validation, clara 2 clusters"
## [1] "Finished internal validation, clara 3 clusters"
## [1] "Finished internal validation, clara 4 clusters"
## [1] "Finished internal validation, clara 5 clusters"
## [1] "Finished internal validation, clara 6 clusters"
## [1] "Finished internal validation, clara 7 clusters"
## [1] "Finished internal validation, clara 8 clusters"
## [1] "Finished internal validation, agnes 2 clusters"
## [1] "Finished internal validation, agnes 3 clusters"
## [1] "Finished internal validation, agnes 4 clusters"
## [1] "Finished internal validation, agnes 5 clusters"
## [1] "Finished internal validation, agnes 6 clusters"
## [1] "Finished internal validation, agnes 7 clusters"
## [1] "Finished internal validation, agnes 8 clusters"
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
##
## ####################
##  40  variables complete
##
## Calculating distance matrix...
## Clustering  hierarchical kmeans diana som sota pam clara agnes ...
```

## The "ward" method has been renamed to "ward.D"; note new "ward.D2"

```
## [1] "Finished internal validation, hierarchical 2 clusters"
## [1] "Finished internal validation, hierarchical 3 clusters"
## [1] "Finished internal validation, hierarchical 4 clusters"
## [1] "Finished internal validation, hierarchical 5 clusters"
## [1] "Finished internal validation, hierarchical 6 clusters"
## [1] "Finished internal validation, hierarchical 7 clusters"
## [1] "Finished internal validation, hierarchical 8 clusters"
```

## The "ward" method has been renamed to "ward.D"; note new "ward.D2"

```
## [1] "Finished internal validation, kmeans 2 clusters"
## [1] "Finished internal validation, kmeans 3 clusters"
## [1] "Finished internal validation, kmeans 4 clusters"
## [1] "Finished internal validation, kmeans 5 clusters"
## [1] "Finished internal validation, kmeans 6 clusters"
## [1] "Finished internal validation, kmeans 7 clusters"
## [1] "Finished internal validation, kmeans 8 clusters"
## [1] "Finished internal validation, diana 2 clusters"
## [1] "Finished internal validation, diana 3 clusters"
## [1] "Finished internal validation, diana 4 clusters"
## [1] "Finished internal validation, diana 5 clusters"
## [1] "Finished internal validation, diana 6 clusters"
## [1] "Finished internal validation, diana 7 clusters"
## [1] "Finished internal validation, diana 8 clusters"
## [1] "Finished internal validation, som 2 clusters"
## [1] "Finished internal validation, som 3 clusters"
## [1] "Finished internal validation, som 4 clusters"
## [1] "Finished internal validation, som 5 clusters"
## [1] "Finished internal validation, som 6 clusters"
## [1] "Finished internal validation, som 7 clusters"
## [1] "Finished internal validation, som 8 clusters"
## [1] "Finished internal validation, sota 2 clusters"
## [1] "Finished internal validation, sota 3 clusters"
## [1] "Finished internal validation, sota 4 clusters"
## [1] "Finished internal validation, sota 5 clusters"
## [1] "Finished internal validation, sota 6 clusters"
## [1] "Finished internal validation, sota 7 clusters"
## [1] "Finished internal validation, sota 8 clusters"
## [1] "Finished internal validation, pam 2 clusters"
## [1] "Finished internal validation, pam 3 clusters"
## [1] "Finished internal validation, pam 4 clusters"
## [1] "Finished internal validation, pam 5 clusters"
## [1] "Finished internal validation, pam 6 clusters"
## [1] "Finished internal validation, pam 7 clusters"
## [1] "Finished internal validation, pam 8 clusters"
## [1] "Finished internal validation, clara 2 clusters"
## [1] "Finished internal validation, clara 3 clusters"
## [1] "Finished internal validation, clara 4 clusters"
## [1] "Finished internal validation, clara 5 clusters"
## [1] "Finished internal validation, clara 6 clusters"
## [1] "Finished internal validation, clara 7 clusters"
## [1] "Finished internal validation, clara 8 clusters"
## [1] "Finished internal validation, agnes 2 clusters"
## [1] "Finished internal validation, agnes 3 clusters"
## [1] "Finished internal validation, agnes 4 clusters"
## [1] "Finished internal validation, agnes 5 clusters"
## [1] "Finished internal validation, agnes 6 clusters"
## [1] "Finished internal validation, agnes 7 clusters"
## [1] "Finished internal validation, agnes 8 clusters"
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
##
## ##################
##  60  variables complete
##
## Calculating distance matrix...
## Clustering  hierarchical kmeans diana som sota pam clara agnes ...

## The "ward" method has been renamed to "ward.D"; note new "ward.D2"

## [1] "Finished internal validation, hierarchical 2 clusters"
## [1] "Finished internal validation, hierarchical 3 clusters"
## [1] "Finished internal validation, hierarchical 4 clusters"
## [1] "Finished internal validation, hierarchical 5 clusters"
## [1] "Finished internal validation, hierarchical 6 clusters"
## [1] "Finished internal validation, hierarchical 7 clusters"
## [1] "Finished internal validation, hierarchical 8 clusters"

## The "ward" method has been renamed to "ward.D"; note new "ward.D2"

## [1] "Finished internal validation, kmeans 2 clusters"
## [1] "Finished internal validation, kmeans 3 clusters"
## [1] "Finished internal validation, kmeans 4 clusters"
## [1] "Finished internal validation, kmeans 5 clusters"
## [1] "Finished internal validation, kmeans 6 clusters"
## [1] "Finished internal validation, kmeans 7 clusters"
## [1] "Finished internal validation, kmeans 8 clusters"
## [1] "Finished internal validation, diana 2 clusters"
## [1] "Finished internal validation, diana 3 clusters"
## [1] "Finished internal validation, diana 4 clusters"
## [1] "Finished internal validation, diana 5 clusters"
## [1] "Finished internal validation, diana 6 clusters"
## [1] "Finished internal validation, diana 7 clusters"
## [1] "Finished internal validation, diana 8 clusters"
## [1] "Finished internal validation, som 2 clusters"
## [1] "Finished internal validation, som 3 clusters"
## [1] "Finished internal validation, som 4 clusters"
## [1] "Finished internal validation, som 5 clusters"
## [1] "Finished internal validation, som 6 clusters"
## [1] "Finished internal validation, som 7 clusters"
## [1] "Finished internal validation, som 8 clusters"
## [1] "Finished internal validation, sota 2 clusters"
## [1] "Finished internal validation, sota 3 clusters"
## [1] "Finished internal validation, sota 4 clusters"
## [1] "Finished internal validation, sota 5 clusters"
## [1] "Finished internal validation, sota 6 clusters"
## [1] "Finished internal validation, sota 7 clusters"
## [1] "Finished internal validation, sota 8 clusters"
## [1] "Finished internal validation, pam 2 clusters"
## [1] "Finished internal validation, pam 3 clusters"
## [1] "Finished internal validation, pam 4 clusters"
## [1] "Finished internal validation, pam 5 clusters"
## [1] "Finished internal validation, pam 6 clusters"
## [1] "Finished internal validation, pam 7 clusters"
## [1] "Finished internal validation, pam 8 clusters"
## [1] "Finished internal validation, clara 2 clusters"
## [1] "Finished internal validation, clara 3 clusters"
## [1] "Finished internal validation, clara 4 clusters"
## [1] "Finished internal validation, clara 5 clusters"
## [1] "Finished internal validation, clara 6 clusters"
## [1] "Finished internal validation, clara 7 clusters"
## [1] "Finished internal validation, clara 8 clusters"
## [1] "Finished internal validation, agnes 2 clusters"
## [1] "Finished internal validation, agnes 3 clusters"
## [1] "Finished internal validation, agnes 4 clusters"
## [1] "Finished internal validation, agnes 5 clusters"
## [1] "Finished internal validation, agnes 6 clusters"
## [1] "Finished internal validation, agnes 7 clusters"
## [1] "Finished internal validation, agnes 8 clusters"
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
##
## ###################
##  80  variables complete
##
## Calculating distance matrix...
## Clustering  hierarchical kmeans diana som sota pam clara agnes ...

## The "ward" method has been renamed to "ward.D"; note new "ward.D2"

## [1] "Finished internal validation, hierarchical 2 clusters"
## [1] "Finished internal validation, hierarchical 3 clusters"
## [1] "Finished internal validation, hierarchical 4 clusters"
## [1] "Finished internal validation, hierarchical 5 clusters"
## [1] "Finished internal validation, hierarchical 6 clusters"
## [1] "Finished internal validation, hierarchical 7 clusters"
## [1] "Finished internal validation, hierarchical 8 clusters"

## The "ward" method has been renamed to "ward.D"; note new "ward.D2"

## [1] "Finished internal validation, kmeans 2 clusters"
## [1] "Finished internal validation, kmeans 3 clusters"
## [1] "Finished internal validation, kmeans 4 clusters"
## [1] "Finished internal validation, kmeans 5 clusters"
## [1] "Finished internal validation, kmeans 6 clusters"
## [1] "Finished internal validation, kmeans 7 clusters"
## [1] "Finished internal validation, kmeans 8 clusters"
## [1] "Finished internal validation, diana 2 clusters"
## [1] "Finished internal validation, diana 3 clusters"
## [1] "Finished internal validation, diana 4 clusters"
## [1] "Finished internal validation, diana 5 clusters"
## [1] "Finished internal validation, diana 6 clusters"
## [1] "Finished internal validation, diana 7 clusters"
## [1] "Finished internal validation, diana 8 clusters"
## [1] "Finished internal validation, som 2 clusters"
## [1] "Finished internal validation, som 3 clusters"
## [1] "Finished internal validation, som 4 clusters"
## [1] "Finished internal validation, som 5 clusters"
## [1] "Finished internal validation, som 6 clusters"
## [1] "Finished internal validation, som 7 clusters"
## [1] "Finished internal validation, som 8 clusters"
## [1] "Finished internal validation, sota 2 clusters"
## [1] "Finished internal validation, sota 3 clusters"
## [1] "Finished internal validation, sota 4 clusters"
## [1] "Finished internal validation, sota 5 clusters"
## [1] "Finished internal validation, sota 6 clusters"
## [1] "Finished internal validation, sota 7 clusters"
## [1] "Finished internal validation, sota 8 clusters"
## [1] "Finished internal validation, pam 2 clusters"
## [1] "Finished internal validation, pam 3 clusters"
## [1] "Finished internal validation, pam 4 clusters"
## [1] "Finished internal validation, pam 5 clusters"
## [1] "Finished internal validation, pam 6 clusters"
## [1] "Finished internal validation, pam 7 clusters"
## [1] "Finished internal validation, pam 8 clusters"
## [1] "Finished internal validation, clara 2 clusters"
## [1] "Finished internal validation, clara 3 clusters"
## [1] "Finished internal validation, clara 4 clusters"
## [1] "Finished internal validation, clara 5 clusters"
## [1] "Finished internal validation, clara 6 clusters"
## [1] "Finished internal validation, clara 7 clusters"
## [1] "Finished internal validation, clara 8 clusters"
## [1] "Finished internal validation, agnes 2 clusters"
## [1] "Finished internal validation, agnes 3 clusters"
## [1] "Finished internal validation, agnes 4 clusters"
## [1] "Finished internal validation, agnes 5 clusters"
## [1] "Finished internal validation, agnes 6 clusters"
## [1] "Finished internal validation, agnes 7 clusters"
## [1] "Finished internal validation, agnes 8 clusters"
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
## Calculating validation metrics ("." for each K) ....... done
##
## ###################
##  100  variables complete
```

Now `BRCA.results` contains the results from running `COMMUNAL` repeatedly on the subsetted data. To get a set of locally optimized algorithms, we measure the percentage of clusterings for which a given algorithm returned any clusters with minSize or fewer members.

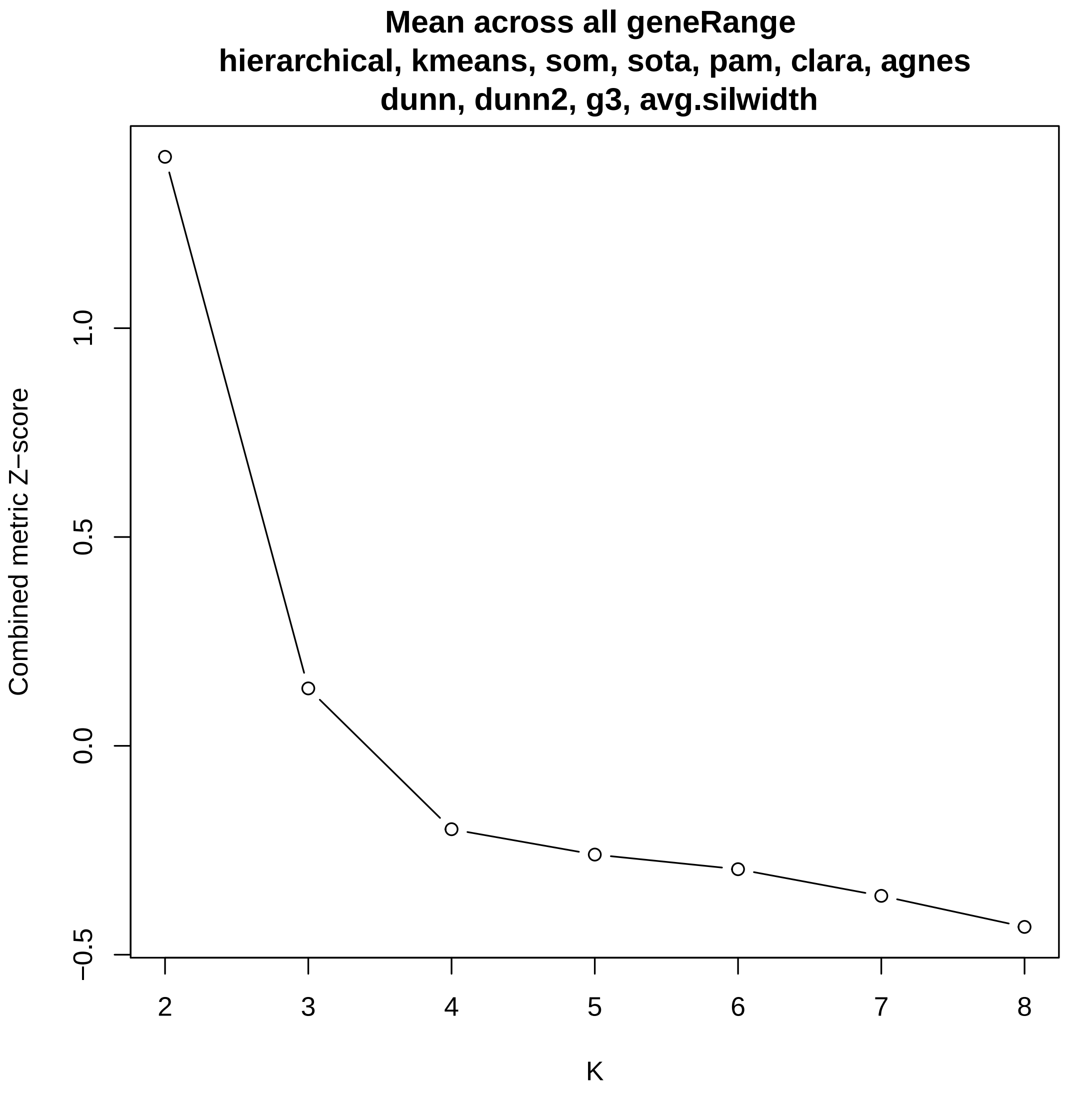
```
algs <- getGoodAlgs(BRCA.results, algs="all")

##
##  Testing for cluster assignments at  20 variables:
##  Warning: at k= 7, some clusters have fewer than 3 members (counts shown below):
##   hierarchical kmeans diana som sota pam clara agnes
## 1           78     77    76  65   93 114   121    46
## 2          137     88   175  79   23  70    92    93
## 3           64     66    53  92   21  76   107    64
## 4           50     65   113 134  103  70    63    87
## 5           76    100    97  62   83  88    28   116
## 6           32     41    18  37   98  66    33    29
## 7           96     96     1  64  112  49    89    98
##
##  Warning: at k= 8, some clusters have fewer than 3 members (counts shown below):
##   hierarchical kmeans diana som sota pam clara agnes
## 1           78     75    76  58   93  94   106    46
## 2           67     78   175  65   23  68    90    56
## 3           64     47    53  59   21  68   111    64
## 4           70     74    96  85  103  68    61    87
## 5           50     55    97  62   98  87    40    37
## 6           76     75    18  68  112  66    43   116
## 7           32     34    17  40   39  33    55    29
## 8           96     95     1  96   44  49    27    98
##
##  Testing for cluster assignments at  40 variables:
##  Testing for cluster assignments at  60 variables:
##  Testing for cluster assignments at  80 variables:
##  Testing for cluster assignments at  100 variables: Final fraction of algs returning clusters
## hierarchical       kmeans        diana          som         sota
##        0.000        0.000        0.011        0.000        0.000
##          pam        clara        agnes
##        0.000        0.000        0.000

algs

## [1] "hierarchical" "kmeans"       "som"          "sota"
## [5] "pam"          "clara"        "agnes"
```

We similarly locally optimize the validity metrics used by eliminating those that are monotone, and those that are highly correlated. The correlation step requires the user to pick a number of measures to keep– the default is four.

```
monotoneClusterRange(BRCA.results)

## average.between            dunn       widestgap           dunn2
##           0.900           0.150           0.600           0.200
##    pearsongamma              g3    max.diameter    avg.silwidth
##           0.400           0.000           0.700           0.425

measuresCorr(BRCA.results)

##                  average.between        dunn   widestgap       dunn2
## average.between       1.00000000  0.77094578  0.95628681  0.12706443
## dunn                  0.77094578  1.00000000  0.72990801 -0.03418530
## widestgap             0.95628681  0.72990801  1.00000000 -0.09085163
## dunn2                 0.12706443 -0.03418530 -0.09085163  1.00000000
## pearsongamma          0.26241156  0.09786953  0.03824156  0.88779324
## g3                    0.18970987 -0.01648310  0.06215495  0.54806174
## max.diameter         0.83864397  0.74652385  0.91673785 -0.36320120
## avg.silwidth         0.02980318 -0.11101399 -0.19293601  0.96910383
##                   pearsongamma          g3 max.diameter avg.silwidth
## average.between   0.26241156  0.18970987   0.83864397   0.02980318
## dunn              0.09786953 -0.01648310   0.74652385  -0.11101399
## widestgap         0.03824156  0.06215495   0.91673785  -0.19293601
## dunn2             0.88779324  0.54806174  -0.36320120   0.96910383
## pearsongamma      1.00000000  0.47881771  -0.23051068   0.87607636
## g3                0.47881771  1.00000000  -0.03866008   0.54450817
## max.diameter     -0.23051068 -0.03866008   1.00000000  -0.45536284
## avg.silwidth      0.87607636  0.54450817  -0.45536284   1.00000000

measures <- getNonCorrNonMonoMeasures(BRCA.results, goodAlgs=algs, numMeasures = 4)

measures

## [1] "dunn"         "dunn2"        "g3"           "avg.silwidth"
```

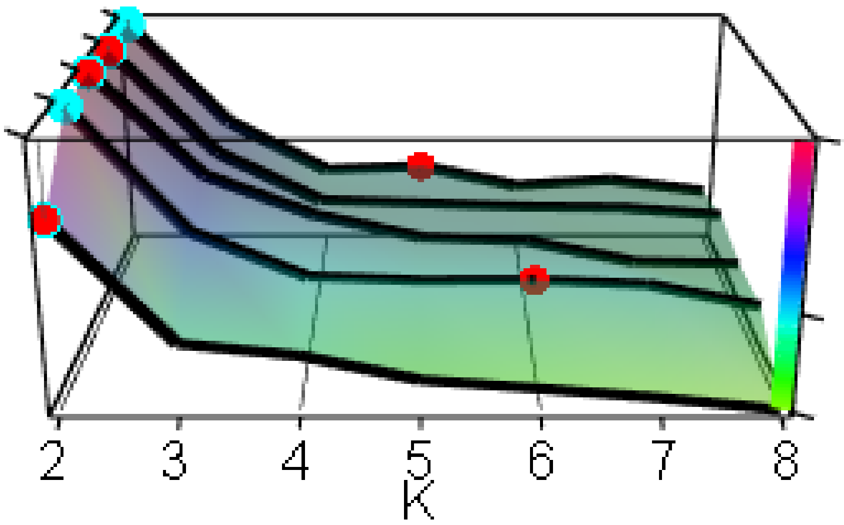
We can generate the plots using `plotRange3D`. These indicate that two clusters is best in this very limited toy example. First will be a 2d plot; this is a view of mean values collapsed for all variable subsets. Second will be a snapshot of the 3D plot; this is the main output, and is normally interactive and pops up in an extra screen. Plot3D has been set to false for this example, but a snapshot of example output is provided.

```
plot.data <- plotRange3D(BRCA.results, ks, algs, measures, plot3D=F)

## [1] "Count of clusters < minSize for the algorithms tested (over all K):\n"
##          hierarchical kmeans som sota pam clara agnes
## vars_20             0      0   0    0   0     0     0
## vars_40             0      0   0    0   0     0     0
## vars_60             0      0   0    0   0     0     0
## vars_80             0      0   0    0   0     0     0
## vars_100            0      0   0    0   0     0     0

     ## Warning in plot.window(...):  "plot3D" is not a graphical parameter
   ## Warning in plot.xy(xy, type, ...):  "plot3D" is not a graphical parameter
## Warning in axis(side = side, at = at, labels = labels, ...):  "plot3D" is not
                              a graphical parameter
## Warning in axis(side = side, at = at, labels = labels, ...):  "plot3D" is not
                              a graphical parameter
         ## Warning in box(...):  "plot3D" is not a graphical parameter
        ## Warning in title(...):  "plot3D" is not a graphical parameter
```



```
## print the values from the 3D plot
plot.data

##           20         40          60          80        100
## 2  1.1239858  1.5595275  1.53665562  1.4246362  1.4062952
## 3 -0.2370076  0.2042595  0.32887190  0.2115819  0.1800708
## 4 -0.3746233 -0.1565756 -0.01102347 -0.2072879 -0.2484114
## 5 -0.5636209 -0.1666047 -0.21326891 -0.1868370 -0.1711399
## 6 -0.6077898 -0.1412145 -0.19124310 -0.1899594 -0.3462682
## 7 -0.6879133 -0.2353223 -0.31531332 -0.2270715 -0.3301034
## 8 -0.7539750 -0.3229215 -0.37932110 -0.3194059 -0.3916615
```

Below is a snapshot of the 3D plot. This shows the results before aggregation into the 2D plot above. The red dots mark the steepest non-edge peak, if a peak exists. The blue points mark the overall maximum for each variable set. The z-axis has labels for Tukey's five number summary of all the values. For an interactive 3D example of `plotRange3D`, you may run this code yourself or see the help page for `plotRange3D`.



Extracting Core Clusters

After looking at these results, we are satisfied that $k = 2$ is optimal. However, solely for demonstration purposes, we will use $k = 3$, since it allows for more interesting results below. The next step is to extract the cluster assignments for $k = 3$ with `getClustering`. It returns a data frame whose rows are the samples and columns are the clustering algorithm names. Each entry is the cluster assignment of a sample from the respective algorithm. This is the input used in identifying 'core' clusters.

However, we first need to choose a variable subset which is optimal. One might choose the subset with the fewest variables where the target clustering is obtained (here 20 variables).

```
result <- BRCA.results$all.results$vars_20
clusters <- result$getClustering(k=3)
apply(clusters, 2, table)

##   hierarchical kmeans diana som sota pam clara agnes
## 1          154    211   189 210  396 180   200   162
## 2          283    216   229 221   44 251   230   273
## 3           96    106   115 102   93 102   103    98
```

From the table, you can see that there are disagreements between clustering algorithms, but in general the clusters fall into three groups of about equal size (just as expected). Now we combine the results to get final cluster assignments. This shows that the algorithms agree on all but 21 of the assignments (which end up in cluster 0, meaning 'unassigned')

```
# re-key cluster labels to most frequent assignments
mat.key <- clusterKeys(clusters)
examineCounts(mat.key)

##      percent.agreement sample.counts percent.remaining.if.removed
## [1,]              50.0            20                         0.96
## [2,]              62.5            54                         0.86
## [3,]              75.0            83                         0.71
## [4,]              87.5           164                         0.40
## [5,]             100.0           212                         0.00

# find 'core' clusters
core <- returnCore(mat.key, agreement.thresh=50) # find 'core' clusters

## A total of 20 samples were rejected as not robustly clustered.
## 96.2 % samples remain.

table(core) # the 'core' clusters

## core
##   0   1   2   3
##  20 177 233 103

head(core) # the cluster assignments

## TCGA.AO.A03P.01 TCGA.A8.A06T.01 TCGA.A8.A07F.01 TCGA.A8.A081.01
##             "1"             "1"             "2"             "2"
## TCGA.A8.A08C.01 TCGA.A8.A08T.01
##             "1"             "0"
```

Now let's consider a more involved example of how `clusterKeys` and `returnCore` are useful. Consider the following cluster assignments. Overall the algorithms agree that there are three clusters, but differ in how they label the clusters. They disagree about the cluster of the last point.

```
clusters.example <- data.frame(
  alg1=as.integer(c(1,1,1,1,1,2,2,2,2,2,3,3,3,3,1)),
  alg2=as.integer(c(1,1,1,1,1,3,3,3,3,3,2,2,2,2,1)),
  alg3=as.integer(c(3,3,3,3,3,1,1,1,1,1,2,2,2,2,2))
)
```

`clusterKeys` reindexes the labels for each algorithm to make the agreement more apparent.

```
mat.key <- clusterKeys(clusters.example)
mat.key # cluster indices are relabeled

##       alg1 alg2 alg3
##  [1,]    1    1    1
##  [2,]    1    1    1
##  [3,]    1    1    1
##  [4,]    1    1    1
##  [5,]    1    1    1
##  [6,]    2    2    2
##  [7,]    2    2    2
##  [8,]    2    2    2
##  [9,]    2    2    2
## [10,]    2    2    2
## [11,]    3    3    3
## [12,]    3    3    3
## [13,]    3    3    3
## [14,]    3    3    3
## [15,]    1    1    3
```

The next step is to synthesize these into "core" clusters. The clusters are assigned by majority vote. If not enough algorithms agree, based on a user-defined threshold, the cluster is left undetermined. `examineCounts` shows how many samples would be undetermined at various threshold levels.

```
examineCounts(mat.key)

##      percent.agreement sample.counts percent.remaining.if.removed
## [1,]          66.66667             1                         0.93
## [2,]         100.00000            14                         0.00
```

Now we use a threshold to retrieve the "core" clusters. The default threshold is 50%, meaning that more than 50% of the algorithms must agree. In this case, if we use the 50% threshold, then all points are assigned to some cluster.

```
core <- returnCore(mat.key, agreement.thresh=50) # find 'core' clusters

## A total of 0 samples were rejected as not robustly clustered.
## 100 % samples remain.

table(core) # the 'core' clusters

## core
## 1 2 3
## 6 5 4
```

However, if we require all algorithms to agree, then one point is undetermined (hence labeled as cluster 0).

```
core <- returnCore(mat.key, agreement.thresh=99)

## A total of 1 samples were rejected as not robustly clustered.
## 93.3 % samples remain.

table(core) # 0 is undetermined

## core
## 0 1 2 3
## 1 5 5 4
```