clust_validity.R usage

Umer Zeeshan Ijaz

The k-means algorithm is arguably the most fundamental clustering/segmentation algorithm in which the number of clusters to be detected is specified *a priori* (commonly provided as a parameter *k*). The correct choice of k is often ambiguous and difficult to be made with interpretation depending on the prior knowledge about the properties of the data set including the shape and scale of the distribution of points. When the clustering result is evaluated on the data that was clustered itself, the process is called internal evaluation and one can use different types of scoring algorithms to produces clusters with high similarity within a cluster and low similarity between clusters. **clust_validity.R** uses such scoring algorithms to find the optimum number of clusters in a dataset. The script supports k-means and dp-means (a nonparametric Bayesian approximation to k-means using dirichlet process). You can specify either of the two algorithms using **--algo** switch.

The script also has an implementation of evidence accumulation-based clustering, which can be run by specifying "EAC" in **--internalCriteria**. Using this algorithm, the data is split into large number of compact and small clusters by using different decompositions obtained by random initialization of k-means algorithm. After this the data from multiple clusterings is mapped into a co-association matrix which provides a degree of similarity between patterns, and finally using this new similarity, hierarchical clustering gives the final partitions, corresponding to merging of clusters.

The other alternative dp-means is similar to k-means with the difference being the objective function having an additional penalty term based on the number of the clusters with a parameter "Lambda" controlling the tradeoff between traditional k-means and cluster terms. If Lambda is small, you will get many clusters and as you increase the parameter, the number of clusters will reduce. We indirectly optimize the Lambda parameter using internal measures to give the optimum number of clusters in the dataset with range of Lambda values specified using **--paramMin** and **--paramMax** parameters. Note that the same parameters in k-means algorithm are used to specify the range of cluster numbers. dp-means is quite sensitive to the ordering of data and it is suggested by the original authors to try the

algorithms with different orderings (k-means on the other hand is only sensitive to initial clusters). To do so, there is an **--nOrderings** switch, which specifies how many orderings you want to consider for each Lambda value. Furthermore, you can use **--lambdaScale** to scale the Lambda parameter, for example if the range is [2,10], and **--lambdaScale** is 10, you will get the actual range of Lambda as [0.2:0.1:1] with 0.1 being the increment.

The script accepts the file in csv format with the features of individual objects in separate rows. Features should have a header starting with names beginning with capital V followed by a number, i.e., V1, V2, V3, and so on. Furthermore, the file can have "True_Clusters" is an optional column, and if provided, the script uses an external criteria to assess how well the clustering algorithm has performed by comparing predicted clusters against the true clusters. The script will still run if you don't include this column.

Run the script with –h option to get the help on how to use it

```
ijaz$ Rscript clust_validity.R -h
usage:  clust_validity.R [options] file

options:
      --file=FILE
            Input csv file
      --algo=ALGO
            [default kmeans]
            Options:
                  kmeans
                  dpmeans
      --paramMin=PARAMMIN
            kmeans: minimum numbers of clusters, dpmeans:minimum value of lambda [default 2]
      --paramMax=PARAMMAX
            kmeans: maximum numbers of clusters, dpmeans:maximum value of lambda [default 50]
      --nOrderings=NORDERINGS
            dpmeans:number of orderings for each value of lambda [default 5]
      --lambdaScale=LAMBDASCALE
            dpmeans: lambda/scale [default 10]
```

```
--iterations=ITERATIONS
      Iterations in EAC [default 200]
--internalCriteria=INTERNALCRITERIA
      [default EAC]
      Options:
            Ball_Hall
            Banfeld_Raftery
            C_index
            Calinski_Harabasz
            Davies_Bouldin
            Det_Ratio
            Dunn
            Gamma
            G_plus
            GDI11
            GDI12
            GDI13
            GDI21
            GDI22
            GDI23
            GDI31
            GDI32
            GDI33
            GDI41
            GDI42
            GDI43
            GDI51
            GDI52
            GDI53
            Ksq_DetW
            Log_Det_Ratio
            Log_SS_Ratio
            McClain_Rao
            PBM
            Point_Biserial
            Ray_Turi
            Ratkowsky_Lance
```

```
            Scott_Symons
            SD_Scat
            SD_Dis
            S_Dbw
            Silhouette
            Tau
            Trace_W
            Trace_WiB
            Wemmert_Gancarski
            Xie_Beni
            EAC

--externalCriteria=EXTERNALCRITERIA
      [default ARI]
      Options:
            Czekanowski_Dice
            Folkes_Mallows
            Hubert
            Jaccard
            Kulczynski
            McNemar
            Phi
            Precision
            Rand
            Recall
            Rogers_Tanimoto
            Russel_Rao
            Sokal_Sneath1
            Sokal_Sneath2
            ARI
--width=WIDTH
      Width of jpeg files [default 800]
--height=HEIGHT
      Height of jpeg files [default 800]
--fsize=FSIZE
      Font size [default 2]
-h, --help
```

```
            Show this help message and exit
```

You can download the script from
http://userweb.eng.gla.ac.uk/umer.ijaz/bioinformatics/clust_validity.R

You can download example datasets from
http://userweb.eng.gla.ac.uk/umer.ijaz/bioinformatics/clust_validity_datasets.zip

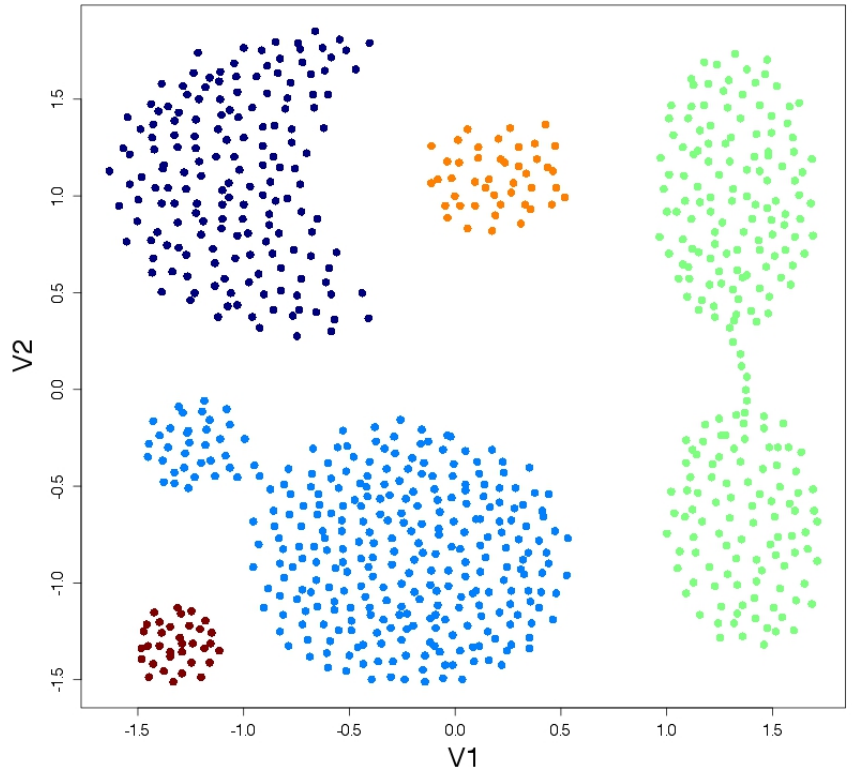
The slides discussing mathematics behind these algorithms are located at
http://userweb.eng.gla.ac.uk/umer.ijaz/bioinformatics/clust_validity.pdf

Look inside the csv files to understand the input format:

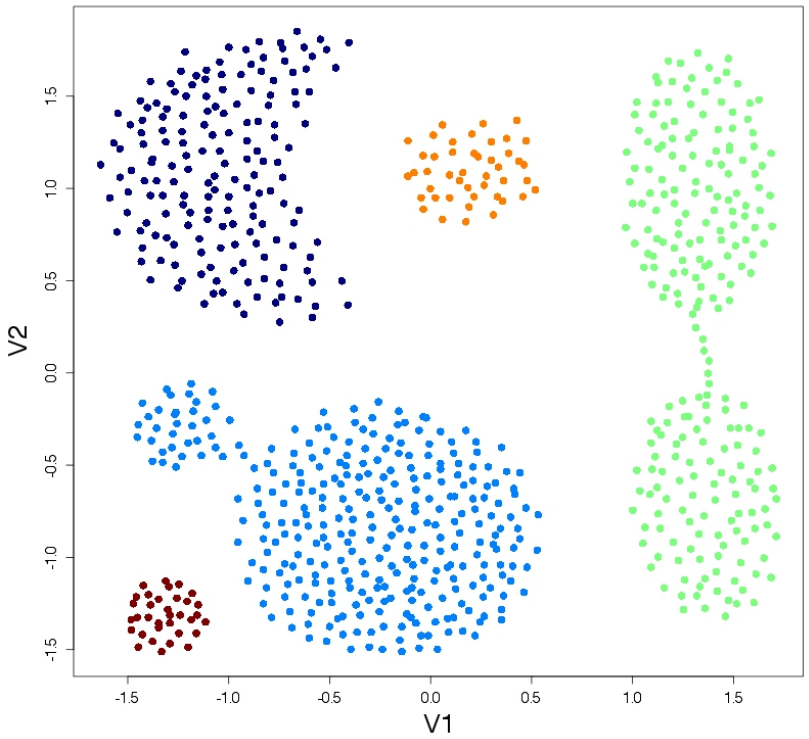
```
ijaz$ head aggregation.csv
,V1,V2,True_Clusters
1,-0.404837489478558,1.78971601133961,2
2,-0.470348195460998,1.65374035747185,2
3,-0.515701761141149,1.75263174210295,2
4,-0.545937471594583,1.80825814595794,2
5,-0.586251752199161,1.71554747286629,2
6,-0.626566032803739,1.76499316518184,2
7,-0.661841028332745,1.85152312673405,2
8,-0.616487462652595,1.64755964593241,2
```

In the next few slides, the results are given for **clust_validity.R** with different values of the parameters (each followed by a plot). You can assess the performance of different parameterizations by looking at "External clustering index" value. ARI stands for Adjusted Rand Index and a higher value is better.

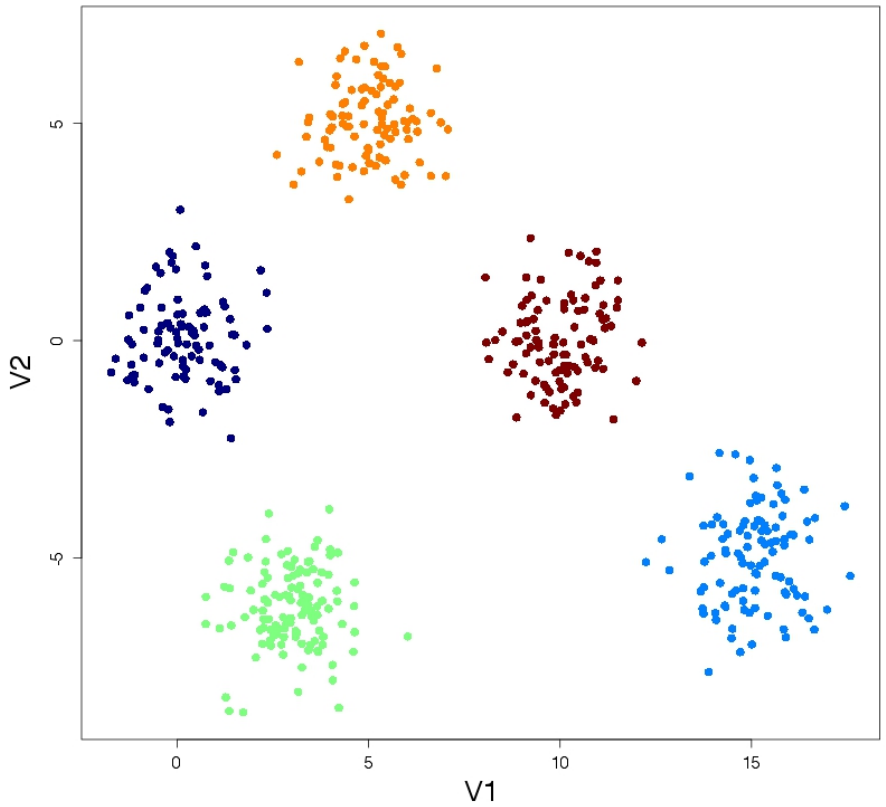
```
ijaz$ Rscript clust_validity.R --paramMin 2 --paramMax 50 --iterations 200 --file aggregation.csv --internalCriteria
EAC --algo kmeans
Computing "Evidence Accumulation-based Clustering" (Iteration:200/200)
External clustering index ("ARI") is 0.808943417036094
Data is 2-dimensional, saving plot as aggregation.jpg

Saving cluster assignments as aggregation_processed.csv
```



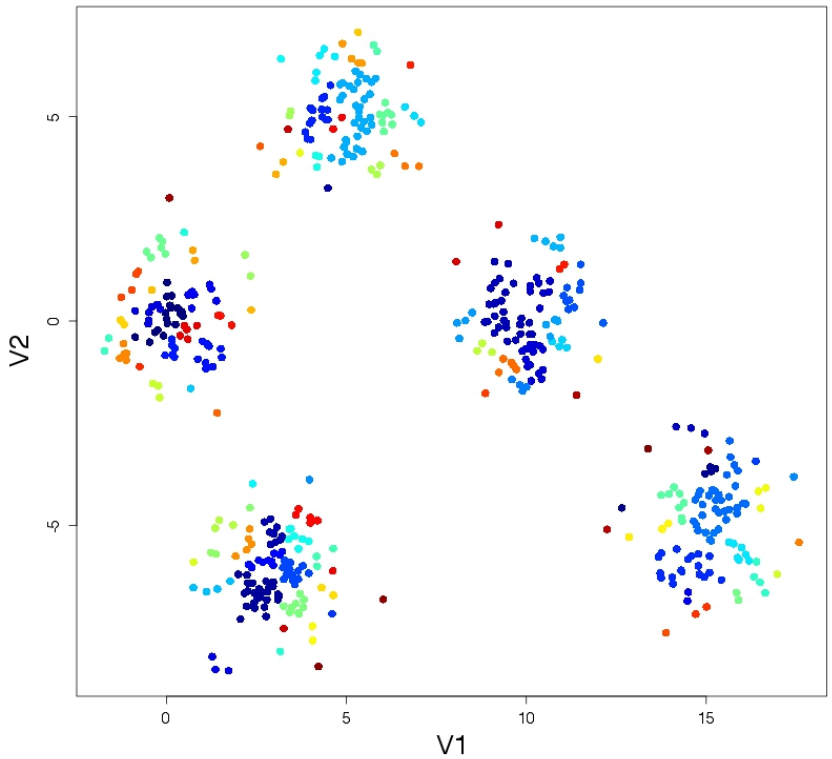
```
ijaz$ Rscript clust_validity.R --paramMin 2 --paramMax 10 --lambdaScale 10 --iterations 5 --file aggregation.csv --
internalCriteria EAC --algo dpmeans
Computing "Evidence Accumulation-based Clustering" (Iteration:5/5)
External clustering index ("ARI") is 0.808943417036094
Data is 2-dimensional, saving plot as aggregation.jpg

Saving cluster assignments as aggregation_processed.csv
```



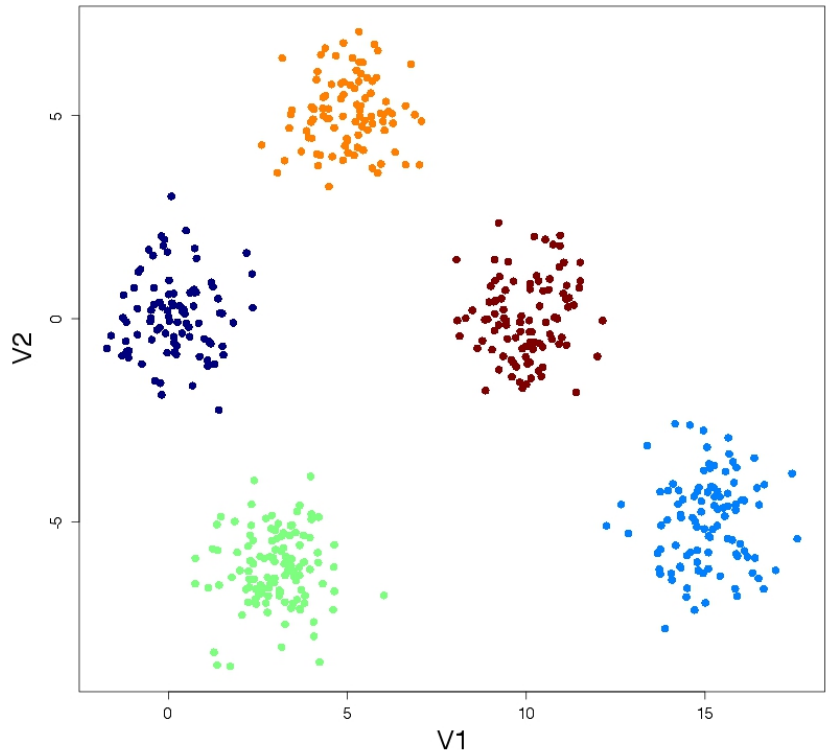
```
ijaz$ Rscript clust_validity.R --paramMin 2 --paramMax 50 --iterations 200 --file fiveclust.csv --internalCriteria
EAC --algo kmeans
Computing "Evidence Accumulation-based Clustering" (Iteration:200/200)
External clustering index ("ARI") is 1
Data is 2-dimensional, saving plot as fiveclust.jpg

Saving cluster assignments as fiveclust_processed.csv
```



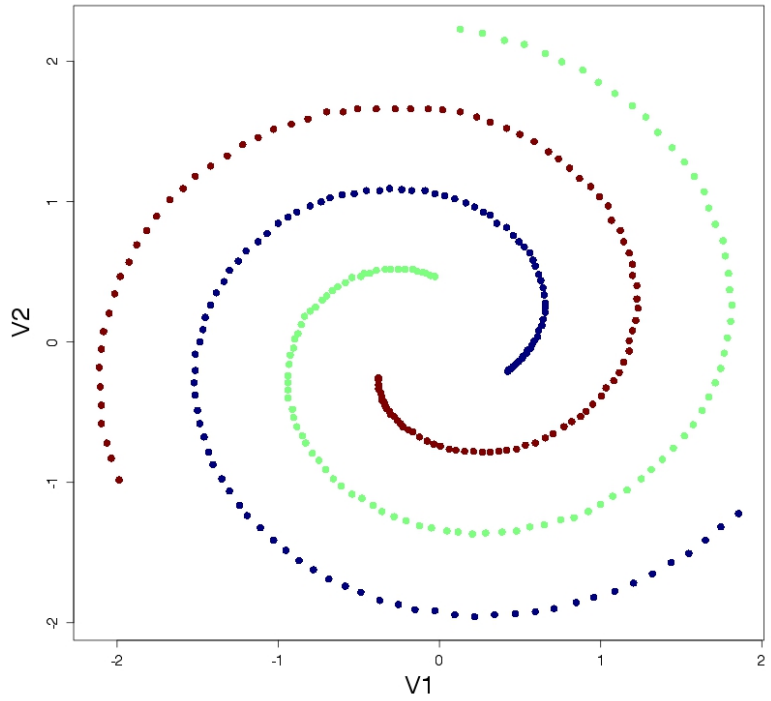
```
ijaz$ Rscript clust_validity.R --paramMin 2 --paramMax 10 --lambdaScale 10 --iterations 5 --file fiveclust.csv --
internalCriteria EAC --algo dpmeans
Computing "Evidence Accumulation-based Clustering" (Iteration:5/5)
External clustering index ("ARI") is 0.171769042005302
Data is 2-dimensional, saving plot as fiveclust.jpg

Saving cluster assignments as fiveclust_processed.csv
```



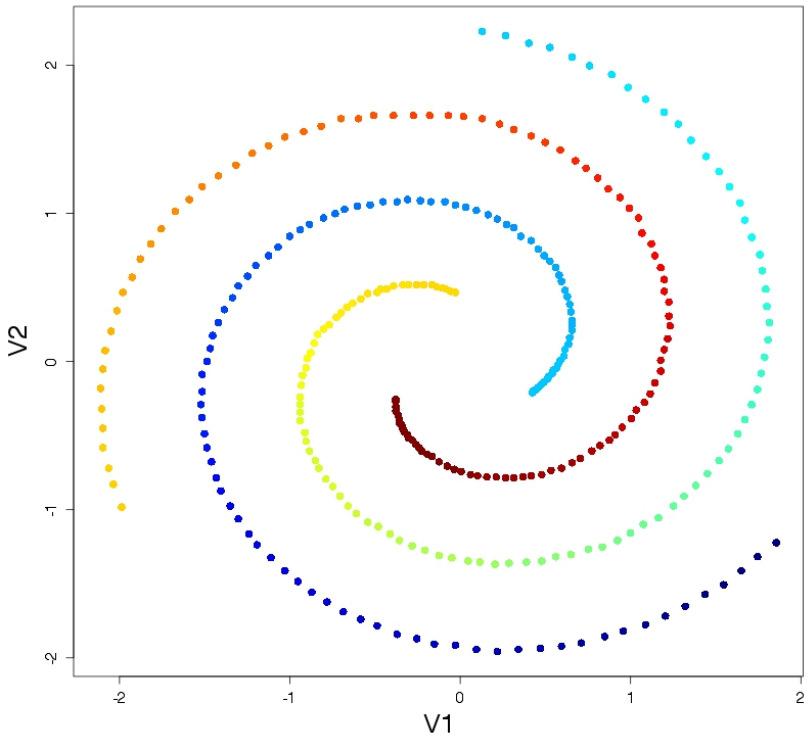
```
ijaz$ Rscript clust_validity.R --paramMin 2 --paramMax 10 --lambdaScale 1 --iterations 5 --file fiveclust.csv --
internalCriteria EAC --algo dpmeans
Computing "Evidence Accumulation-based Clustering" (Iteration:5/5)
External clustering index ("ARI") is 1
Data is 2-dimensional, saving plot as fiveclust.jpg

Saving cluster assignments as fiveclust_processed.csv
```



```
ijaz$ Rscript clust_validity.R --paramMin 2 --paramMax 50 --iterations 200 --file spiral.csv --internalCriteria EAC
--algo kmeans
Computing "Evidence Accumulation-based Clustering" (Iteration:200/200)Warning message:
did not converge in 10 iterations

External clustering index ("ARI") is 1
Data is 2-dimensional, saving plot as spiral.jpg

Saving cluster assignments as spiral_processed.csv
```



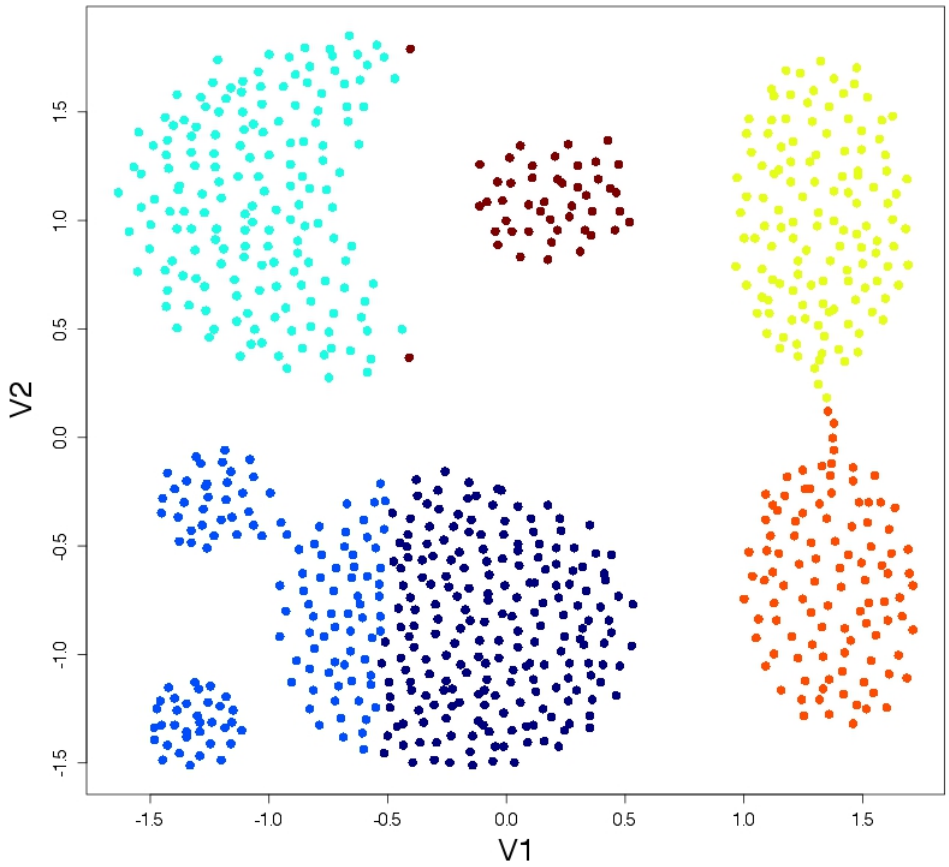
```
ijaz$ Rscript clust_validity.R --paramMin 1 --paramMax 20 --lambdaScale 10 --iterations 5 --file spiral.csv --
internalCriteria EAC --algo dpmeans
Computing "Evidence Accumulation-based Clustering" (Iteration:5/5)
External clustering index ("ARI") is 0.0577962071394242
Data is 2-dimensional, saving plot as spiral.jpg

Saving cluster assignments as spiral_processed.csv
```



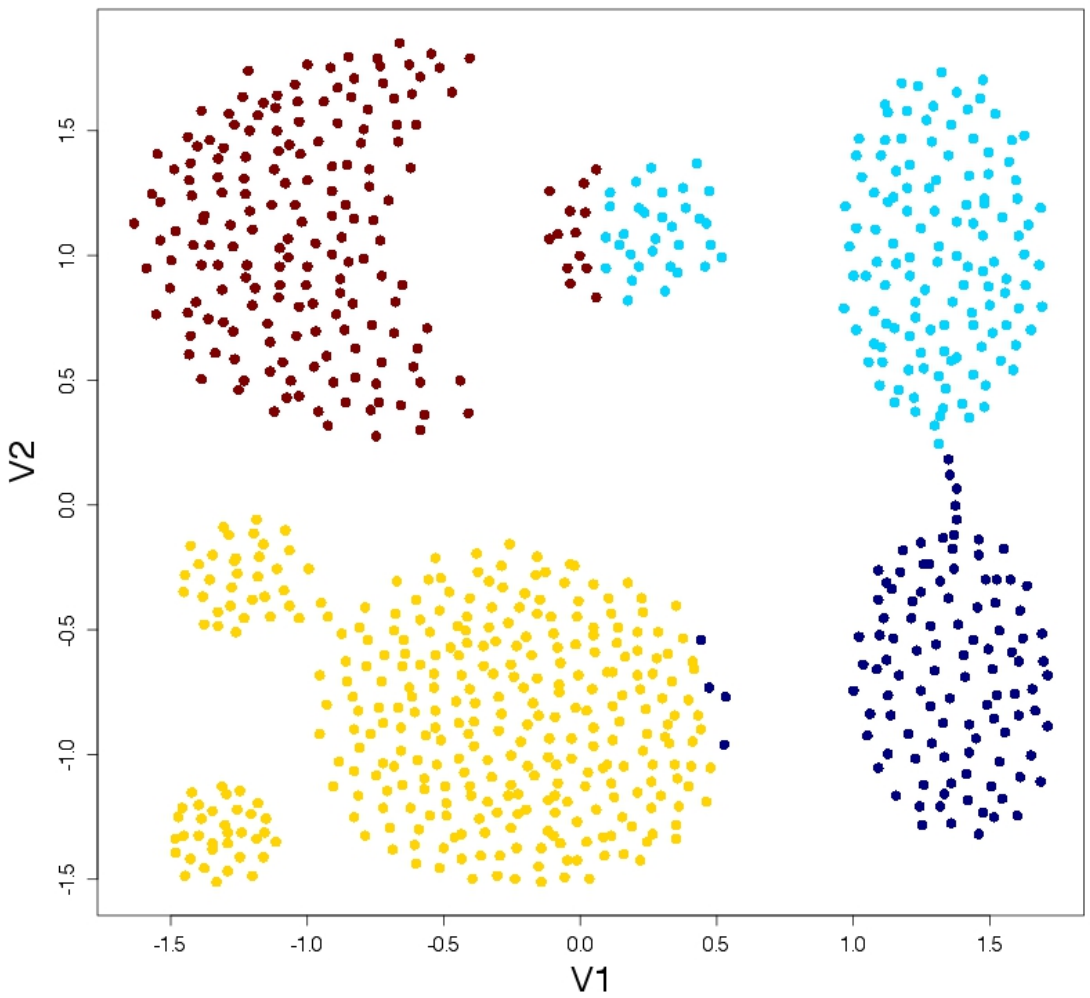
```
ijaz$ Rscript clust_validity.R --paramMin 2 --paramMax 20 --lambdaScale 10 --file aggregation.csv --internalCriteria
Silhouette --algo dpmeans
Computing "Silhouette" (Lambda:2,Ordering:5)5)
Internal clustering index ("Silhouette") for different values of Lambda is as follows (Inf for cases when only one
cluster is obtained):
 Lambda Silhouette
    0.2  0.3700269
    0.2  0.3936386
    0.2  0.3760552
    0.2  0.3892554
    0.2  0.3699375
    0.3  0.3994360
    0.3  0.4271194
    0.3  0.3898639
    0.3  0.3827509
    0.3  0.3959035
    0.4  0.4245877
    0.4  0.4353942
    0.4  0.4342159
    0.4  0.4536903
    0.4  0.4339835
    0.5  0.4428555
    0.5  0.4505622
    0.5  0.4601076
    0.5  0.4635115
    0.5  0.4641889
    0.6  0.4632556
    0.6  0.4842043
    0.6  0.4393602
    0.6  0.4543664
    0.6  0.4687018
    0.7  0.4790235
    0.7  0.4972620
    0.7  0.4448603
    0.7  0.4743781
    0.7  0.4806873
    0.8  0.4880329
```

0.8	0.4798123
0.8	0.4823118
0.8	0.4487939
0.8	0.4868553
0.9	0.4739541
0.9	0.4488826
0.9	0.4543610
0.9	0.4509979
0.9	0.4789557
1.0	0.4674565
1.0	0.4604248
1.0	0.4795943
1.0	0.4610976
1.0	0.4610976
1.1	0.4610976
1.1	0.4758925
1.1	0.4610976
1.1	0.4547046
1.1	0.4812989
1.2	0.4476079
1.2	0.5017307
1.2	0.4677169
1.2	0.4556616
1.2	0.4689936
1.3	0.5130317
1.3	0.4688110
1.3	0.4760593
1.3	0.4545682
1.3	0.4747375
1.4	0.4389001
1.4	0.4470038
1.4	0.4984796
1.4	0.4850132
1.4	0.4151299
1.5	0.4842267
1.5	0.4617279
1.5	0.4844400

```
    1.5  0.4850132
    1.5  0.4821641
    1.6  0.4842566
    1.6  0.4281352
    1.6  0.4850132
    1.6  0.4630158
    1.6  0.4842566
    1.7  0.4344730
    1.7  0.4842566
    1.7  0.4630158
    1.7  0.4631491
    1.7  0.4842566
    1.8  0.4629551
    1.8  0.4847829
    1.8  0.4617279
    1.8  0.4617279
    1.8  0.4389027
    1.9  0.4630158
    1.9  0.4630158
    1.9  0.4629551
    1.9  0.4630158
    1.9  0.4630158
    2.0  0.5132339
    2.0  0.4986008
    2.0  0.4984796
    2.0  0.4993289
    2.0  0.4844104

Optimum cluster size is 6 for lambda=2
External clustering index ("ARI") is 0.792671799302784
Data is 2-dimensional, saving plot as aggregation.jpg

Saving cluster assignments as aggregation_processed.csv
```


V2
1.5
1.0
0.5
0.0
-0.5
-1.0
-1.5
-1.5
-1.0
-0.5
0.0
0.5
1.0
1.5
V1

```
ijaz$ Rscript clust_validity.R --paramMin 2 --paramMax 20 --file aggregation.csv --internalCriteria Silhouette --
algo kmeans
Computing "Silhouette" (K:20)
Internal clustering index ("Silhouette") for different values of K is as follows:
  K Silhouette
  2  0.4324140
  3  0.5188809
  4  0.5606036
  5  0.4984796
  6  0.4632183
  7  0.4907981
  8  0.4497794
  9  0.4500851
 10  0.4729748
 11  0.4259923
 12  0.4249264
 13  0.4764234
 14  0.4680052
 15  0.3884708
 16  0.3972855
 17  0.4278382
 18  0.4040233
 19  0.4157585
 20  0.4091827

Optimum cluster size is 4
External clustering index ("ARI") is 0.762285233288582
Data is 2-dimensional, saving plot as aggregation.jpg

Saving cluster assignments as aggregation_processed.csv
```


V2
1.5
1.0
0.5
0.0
-0.5
-1.0
-1.5
-1.5
-1.0
-0.5
0.0
0.5
1.0
1.5
V1