Data Preparation Techniques using PCA,CA,MCA for Hybrid H_K-Means, CLARA, Hybrid K-Means-K_Mediods, Fuzzy methods to efficiently Cluster Bioinformatics Data.

"Katikireddy Srinivas[1],
Research Scholar, Department of CSE,
K L E F, Vaddeswaram,Guntur Dt,AndhraPradesh.

Dr. K V D Kiran[2]
Professor of CSE,Department of CSE,
K L E F, Vaddeswaram,Guntur Dt,AndhraPradesh.

Abstract

Bio-Informatics is dealing with integration of Biotechnology, Biomedical, Biological concepts with Computer science and Information Technology. At this pandemic Covid-19 health emergency scenario, Machine learning is very much help full for Tele-health and tele medicine tools to face the pandemic challenges effectively. As a supporting module to Tele medicine I would like to device an automated drug choosing system for the patients suffering with routine illness conditions from the existing valid drug bank using AI and Machine learning.

In this scenario I would like to use computer science knowledge based on the physio chemical properties and enzyme inhibition properties of drug dataset provided by standard drug bank repository ie www.drugbank.ca and www.malacards.org.Here I applied existing clustering techniques as a blend of k-means, k-medoids, hierarchical methods and Fuzzy k-means to determine an appropriate set of drugs from the given drugbank for different illness conditions of of thyroid patient].In this research work data preparation for drug evaluation is playing a crucial role , we used PCA,CA,MCA methods to make our Cluster system is efficient and effective. We have shown the analysis of blend of cluster methods successful for this cluster system using graphical presentation and derived best hybrid cluster methods as final outcome.

Keywords: Machine Learning, Covid-19,PCA,CA,MCA, Clustering, Thyroid

1.0 Introduction:

In K-means or PAM bunching, the information is partitioned into particular groups, where every component is influenced precisely to one bunch[6]. This sort of grouping is otherwise called hard bunching or non-fluffy bunching. Dissimilar to K-implies, Fuzzy bunching is considered as a delicate grouping, in which every component has a likelihood of having a place with each bunch. At the end of the day, every component has a lot of enrollment coefficients comparing to the level of being in a given group.

Focuses near the focal point of a bunch, might be in the group to a further extent than focuses in the edge of a group. The degree, to which a component has a place with a given bunch, is a numerical incentive in [18].

1.1 Fuzzy C: Fuzzy c-implies (FCM) calculation is one of the most broadly utilized fluffy grouping calculations. It was created by Dunn in 1973 and improved by Bezdek in 1981. It's regularly utilized in design acknowledgment[8].

FCM calculation is fundamentally the same as the k-implies calculation[11] and the point is to limit the target work characterized as follow:

$$\sum j=1k\sum xi\in Cjumij(xi-\mu j)2\sum j=1k\sum xi\in Cjuijm(xi-\mu j)2$$

Where,

- **uijuij is how much a perception xixi has a place with a bunch cjcj**
- **μjμj is the focal point of the group j**
- **uijuij is how much a perception xixi has a place with a bunch cjcj**
- **mm is the fuzzifier.**

The variable umijuijm is characterized as follow:

$$umij=1\sum l=1k(|xi-cj||xi-ck|)2m-1uijm=1\sum l=1k(|xi-cj||xi-ck|)2m-1$$

The level of having a place, uijuij, is connected conversely to the good ways from x to the group community.

The boundary mm is a genuine number more noteworthy than 1 (1.0<m<∞1.0<m<∞) and it characterizes the degree of group fluffiness. Note that, an estimation of mm near 1 gives a group arrangement which turns out to be progressively like the arrangement of hard bunching, for example, k-implies; though an estimation of mm near interminable prompts total fuzzyness[12].

In fluffy grouping the centroid of a bunch is he mean all things considered, weighted by their level of having a place with the bunch[18]:

$$Cj=\sum x\in Cjumijx\sum x\in CjumijCj=\sum x\in Cjuijmx\sum x\in Cjuijm$$

Where,

- **CjCj is the centroid of the group j**
- **uijuij is how much a perception xixi has a place with a bunch cjcj**

Every segment k (for example gathering or group) is demonstrated by the typical or Gaussian dissemination which is portrayed by the boundaries:

- **μkμk: mean vector,**
- **∑k∑k: covariance lattice,**

- **An related likelihood in the blend. Each point has a likelihood of having a place with each bunch.**

1.2. Favorable position of Model Clustering:

The key favorable position of model-based methodology, contrasted with the standard grouping techniques (k-implies, progressive bunching, …), is the recommendation of the quantity of groups and a fitting model[14]

2.0 Materials and Methods: We used R –Software which is a better software development environment and graphical presentation programming language suitable for data science applications which is also a open source language[17].

We used a drug dataset choosed from www.malacards.org[9] and www.drugbank.ca[10] for thyroid disease. The data set was pre processed using dimensionality reduction techniques like PCA,CA in addition to normalization methods.

2.1 R Functions for Fuzzy Clustering:

fanny(): Fuzzy examination bunching

The capacity fanny() [in group package] can be utilized to process fluffy bunching. FANNY represents fluffy examination bunching. A disentangled configuration is:

fanny(dataframe, k, memb.exp = 2, metric = "euclidean",

stand = FALSE, maxit = 500)

dataframe: An information network or information edge or uniqueness framework

k: The ideal number of groups to be produced

memb.exp: The enrollment example (carefully bigger than 1) utilized in the fit models. It's otherwise called the fuzzifier

metric: The measurement to be utilized for figuring dissimilarities between perceptions

stand: Logical; assuming valid, the estimations in x are normalized before figuring the dissimilarities

maxit: maximal number of cycles

The capacity fanny() restores an article including the accompanying segments:

membership: lattice containing how much every perception has a place with a given group. Section names are the bunches and columns are perceptions

coeff: Dunn's parcel coefficient F(k) of the grouping,

clustering: the bunching vector containing the closest fresh gathering of perceptions

2.2 Mclust(): R work for figuring model-based bunching

The capacity Mclust() [in mclust package] can be utilized to figure model-based bunching[13].

Introduce and burden the bundle as follow:

 # Install

install.packages("mclust")

 # Load dataset

library("mclust")

The capacity Mclust() gives the ideal blend model estimation as indicated by BIC. An improved arrangement is[2]:

Mclust(data, G = NULL)

data: A numeric vector, framework or information outline. Unmitigated factors are not permitted. In the event that a lattice or information outline, lines relate to perceptions and segments compare to factors[2].

G: A whole number vector indicating the quantities of blend parts (bunches) for which the BIC is to be determined. The default is G=1:9.

The capacity Mclust() restores an object of class 'Mclust' containing the accompanying components:

modelName: A character string indicating the model at which the ideal BIC happens.

G: The ideal number of blend parts (i.e: number of groups)

BIC: All BIV esteems

bic Optimal BIC esteem

loglik: The loglikelihood comparing to the ideal BIC

df: The quantity of evaluated boundaries

Z: A grid whose [i,k]th[i,k]th section is the likelihood that perception ii in the test information has a place with the kthkth class. Segment names are group numbers, and lines are perceptions [13]

classification: The group number of every perception, for example map(z)

uncertainty: The vulnerability related with the order3.5 Cluster analysis using

3.0 Results and Discussion:

Fanny() function Implementation:

```
> res.fanny<-fanny(desc_stats,3, memb.exp = 2, metric = "euclidean",
+       stand = FALSE, maxit = 500)
```

(Where thyroid dataset represented as desc_stats and Number of clusters choosed as 3,membership coefficient as 2 and maximum iterations 500)[17]

Fuzzy Clustering object of class 'fanny' :

3.1 Print(res.fanny)

```
> fanny(df_Scaled,3, memb.exp = 2, metric = "euclidean",
+       stand = FALSE, maxit = 500)[15]
Fuzzy Clustering object of class 'fanny' :
m.ship.expon.       2
objective     247.8761
tolerance       1e-15
iterations         12
converged           1
maxit             500
n                 188
Membership coefficients (in %, rounded):
      [,1] [,2] [,3]
 [1,]   33   33   33
 [2,]   33   33   33
 [3,]   33   33   33
 [4,]   33   33   33
 [5,]   33   33   33
 [6,]   33   33   33
 [7,]   33   33   33
 [8,]   33   33   33
 [9,]   33   33   33
[10,]   33   33   33
[11,]   33   33   33
[12,]   33   33   33
```

```
[13,] 33 33 33
[14,] 33 33 33
[15,] 33 33 33
[16,] 33 33 33
[17,] 33 33 33
[18,] 33 33 33
[19,] 33 33 33
[20,] 33 33 33
[21,] 33 33 33
[22,] 33 33 33
[23,] 33 33 33
[24,] 33 33 33
[25,] 33 33 33
[26,] 33 33 33
[27,] 33 33 33
[28,] 33 33 33
[29,] 33 33 33
[30,] 33 33 33
[31,] 33 33 33
[32,] 33 33 33
[33,] 33 33 33
[34,] 33 33 33
[35,] 33 33 33
[36,] 33 33 33
[37,] 33 33 33
[38,] 33 33 33
[39,] 33 33 33
[40,] 33 33 33
[41,] 33 33 33
[42,] 33 33 33
[43,] 33 33 33
[44,] 33 33 33
[45,] 33 33 33
[46,] 33 33 33
[47,] 33 33 33
[48,] 33 33 33
[49,] 33 33 33
[50,] 33 33 33
[51,] 33 33 33
[52,] 33 33 33
[53,] 33 33 33
[54,] 33 33 33
[55,] 33 33 33
[56,] 33 33 33
[57,] 33 33 33
[58,] 33 33 33
```

```
[59,]  33  33  33
[60,]  33  33  33
[61,]  33  33  33
[62,]  33  33  33
[63,]  33  33  33
[64,]  33  33  33
[65,]  33  33  33
[66,]  33  33  33
[67,]  33  33  33
[68,]  33  33  33
[69,]  33  33  33
[70,]  33  33  33
[71,]  33  33  33
[72,]  33  33  33
[73,]  33  33  33
[74,]  33  33  33
[75,]  33  33  33
[76,]  33  33  33
[77,]  33  33  33
[78,]  33  33  33
[79,]  33  33  33
[80,]  33  33  33
[81,]  33  33  33
[82,]  33  33  33
[83,]  33  33  33
[84,]  33  33  33
[85,]  33  33  33
[86,]  33  33  33
[87,]  33  33  33
[88,]  33  33  33
[89,]  33  33  33
[90,]  33  33  33
[91,]  33  33  33
[92,]  33  33  33
[93,]  33  33  33
[94,]  33  33  33
[95,]  33  33  33
[96,]  33  33  33
[97,]  33  33  33
[98,]  33  33  33
[99,]  33  33  33
[100,]  33  33  33
[101,]  33  33  33
[102,]  33  33  33
[103,]  33  33  33
[104,]  33  33  33
```

8

```
[105,]  33  33  33
[106,]  33  33  33
[107,]  33  33  33
[108,]  33  33  33
[109,]  33  33  33
[110,]  33  33  33
[111,]  33  33  33
[112,]  33  33  33
[113,]  33  33  33
[114,]  33  33  33
[115,]  33  33  33
[116,]  33  33  33
[117,]  33  33  33
[118,]  33  33  33
[119,]  33  33  33
[120,]  33  33  33
[121,]  33  33  33
[122,]  33  33  33
[123,]  33  33  33
[124,]  33  33  33
[125,]  33  33  33
[126,]  33  33  33
[127,]  33  33  33
[128,]  33  33  33
[129,]  33  33  33
[130,]  33  33  33
[131,]  33  33  33
[132,]  33  33  33
[133,]  33  33  33
[134,]  33  33  33
[135,]  33  33  33
[136,]  33  33  33
[137,]  33  33  33
[138,]  33  33  33
[139,]  33  33  33
[140,]  33  33  33
[141,]  33  33  33
[142,]  33  33  33
[143,]  33  33  33
[144,]  33  33  33
[145,]  33  33  33
[146,]  33  33  33
[147,]  33  33  33
[148,]  33  33  33
[149,]  33  33  33
[150,]  33  33  33
```

[151,] 33 33 33
[152,] 33 33 33
[153,] 33 33 33
[154,] 33 33 33
[155,] 33 33 33
[156,] 33 33 33
[157,] 33 33 33
[158,] 33 33 33
[159,] 33 33 33
[160,] 33 33 33
[161,] 33 33 33
[162,] 33 33 33
[163,] 33 33 33
[164,] 33 33 33
[165,] 33 33 33
[166,] 33 33 33
[167,] 33 33 33
[168,] 33 33 33
[169,] 33 33 33
[170,] 33 33 33
[171,] 33 33 33
[172,] 33 33 33
[173,] 33 33 33
[174,] 33 33 33
[175,] 33 33 33
[176,] 33 33 33
[177,] 33 33 33
[178,] 33 33 33
[179,] 33 33 33
[180,] 33 33 33
[181,] 33 33 33
[182,] 33 33 33
[183,] 33 33 33
[184,] 33 33 33
[185,] 33 33 33
[186,] 33 33 33
[187,] 33 33 33
[188,] 33 33 33

3.2 Fuzzyness coefficients:

dunn_coeff normalized
3.333333e-01 -3.858025e-15

3.3 Closest hard clustering:

[1] 1 1 1 1 2 2 1 2 1 1 1 1 1 1 1 2 2 1 2 2 2 1 2 1 2 1 1 2 2 2 1 2 1 2 2 1 2 1
[39] 2 1 2 2 2 1 2 1 2 1 1 2 1 1 1 2 2 2 1 2 2 2 2 2 2 1 1 2 1 2 2 2 1 1 2 2 1 1
[77] 2 1 1 2 2 2 1 1 2 1 2 2 1 2 2 2 2 2 2 1 2 1 1 2 1 2 1 1 1 1 1 2 1 2 1 1 1 2

[115] 1 2 1 2 1 1 1 1 2 2 1 2 2 1 2 2 1 2 1 1 2 2 1 2 2 2 2 1 2 1 1 2 2 2 1 2 2 2
[153] 1 1 1 1 2 2 2 2 2 2 2 1 1 1 1 2 2 1 1 2 1 2 2 2 2 1 2 2 2 2 2 2 1 1 1 1
k_crisp (= 2) < k !!

3.4 Available components [4]:

```
 [1] "membership"  "coeff"       "memb.exp"    "clustering"  "k.crisp"
 [6] "objective"   "convergence" "diss"        "call"        "silinfo"
[11] "data"
```

3.5 df1$silinfo

$widths

	cluster	neighbor	sil_width
104	1	3	0.2808217748
89	1	2	0.2778023090
173	1	3	0.2575284536
76	1	3	0.2549628910
119	1	3	0.2549524308
178	1	3	0.2495973538
99	1	3	0.2482431600
112	1	2	0.2437275940
27	1	2	0.2434497242
120	1	3	0.2423448060
105	1	3	0.2411886163
142	1	3	0.2388319919
144	1	3	0.2381546382
53	1	3	0.2376134009
188	1	2	0.2331177923
12	1	2	0.2295465546
117	1	2	0.2270606170
128	1	3	0.2253530526
10	1	2	0.2252213146
44	1	3	0.2239349247
11	1	2	0.2230518399
103	1	3	0.2211325900
52	1	3	0.2183596731
121	1	2	0.2176693905
149	1	3	0.2151056436
48	1	2	0.2147487097
7	1	2	0.2138655403
83	1	2	0.2106181748
33	1	3	0.2078660865
84	1	3	0.2069226017
137	1	3	0.2064122786
72	1	3	0.2063532706
4	1	2	0.2014980397
125	1	2	0.1901725166

67	1	3	0.1863656903
115	1	2	0.1852951883
78	1	2	0.1847656555
49	1	2	0.1826183957
155	1	3	0.1820322889
171	1	2	0.1807279246
46	1	2	0.1786163807
186	1	3	0.1785684981
153	1	2	0.1722374724
145	1	2	0.1705178903
111	1	2	0.1615295067
13	1	3	0.1570615458
86	1	2	0.1536790545
131	1	2	0.1524412755
167	1	2	0.1520211750
170	1	2	0.1502934935
101	1	2	0.1497019665
1	1	2	0.1487588534
15	1	2	0.1427890104
71	1	2	0.1218759806
22	1	3	0.1205668357
26	1	2	0.1159499002
122	1	2	0.1114672263
107	1	3	0.1049542979
36	1	2	0.1010632421
31	1	2	0.1010474928
185	1	2	0.0942635927
79	1	2	0.0891393319
113	1	2	0.0852367871
106	1	2	0.0790730856
18	1	2	0.0784235620
187	1	2	0.0723804489
133	1	2	0.0657373800
57	1	3	0.0649526431
109	1	3	0.0596593231
154	1	3	0.0412956767
156	1	2	0.0396302733
164	1	3	0.0146553285
64	1	3	-0.0027082283
165	1	3	-0.0173955577
51	1	3	-0.0204647754
14	1	3	-0.0217174402
75	1	3	-0.0799327754
38	1	3	-0.0822980540
166	1	3	-0.0855611876
65	1	3	-0.0933074732

2	1	3	-0.1031216071
13	1	3	-0.1071259664
40	1	3	-0.1072212591
3	1	3	-0.1094636166
98	1	3	-0.1216668318
9	1	3	-0.2214832144
96	1	3	-0.2918220885
141	2	3	0.2108370288
138	2	1	0.2082399021
159	2	1	0.2029071676
91	2	1	0.2008301277
74	2	1	0.1862891857
97	2	1	0.1731573672
95	2	1	0.1700601312
147	2	3	0.1640843782
34	2	1	0.1639429245
160	2	1	0.1638523657
30	2	1	0.1619556659
124	2	1	0.1619200225
148	2	3	0.1570035349
39	2	1	0.1560069899
139	2	1	0.1445490667
45	2	1	0.1423409656
168	2	1	0.1410095137
162	2	1	0.1380553724
174	2	1	0.1335158837
87	2	1	0.1324484543
157	2	1	0.1309605308
77	2	1	0.1301578808
66	2	1	0.1282585339
108	2	3	0.1276319246
23	2	1	0.1271629012
35	2	1	0.1189549826
161	2	1	0.1138578151
102	2	1	0.1120330642
41	2	1	0.1119444075
62	2	1	0.1065866749
47	2	1	0.0995883264
69	2	1	0.0982061954
129	2	3	0.0962197158
118	2	1	0.0936438535
146	2	3	0.0934609576
28	2	1	0.0914896633
116	2	1	0.0880424606
63	2	1	0.0866306794
43	2	3	0.0833570730

123	2	1	0.0771262208
94	2	1	0.0760230385
132	2	1	0.0757786777
136	2	1	0.0689696241
143	2	3	0.0579552087
85	2	3	0.0557577261
81	2	3	0.0450562550
68	2	1	0.0431937022
21	2	1	0.0422612170
29	2	1	0.0407908629
126	2	3	0.0400671907
182	2	1	0.0335963098
54	2	3	0.0322151044
20	2	1	0.0321251814
17	2	3	0.0307737253
181	2	3	0.0307020291
50	2	1	0.0305795451
82	2	3	0.0305414020
19	2	1	0.0304657580
177	2	1	0.0284323790
175	2	1	0.0231584496
158	2	1	0.0194147981
90	2	1	0.0193447732
135	2	1	0.0175481264
70	2	1	0.0120464755
32	2	1	0.0109845034
60	2	1	0.0052755018
73	2	1	0.0048511724
92	2	3	0.0002151293
37	2	1	-0.0066402343
25	2	1	-0.0076092935
172	2	3	-0.0078243371
6	2	1	-0.0096527206
151	2	3	-0.0166127819
110	2	1	-0.0208922197
140	2	1	-0.0246207936
80	2	1	-0.0254616617
127	2	1	-0.0276790262
114	2	1	-0.0386179161
55	2	1	-0.0394324658
5	2	1	-0.0420784560
42	2	1	-0.0434588655
16	2	1	-0.0453570971
183	2	3	-0.0467778552
176	2	1	-0.0497979556
163	2	1	-0.0557941564

24	2	1	-0.0602629996
184	2	3	-0.0644228036
8	2	3	-0.0736807081
130	2	3	-0.0960144587
88	2	3	-0.1091651941
61	2	3	-0.1286480115
100	2	3	-0.1589658438
152	2	3	-0.1657404914
150	2	3	-0.1889818616
56	2	3	-0.2416773454
58	2	3	-0.2655623741
180	2	3	-0.2862732749
179	2	3	-0.2885538606
59	2	3	-0.3041267755
93	3	1	0.0751194911
169	3	1	-0.2425422529

3.5.1 $clus.avg.widths

[1] 0.12750958 0.03460667 -0.08371138

3.5.2 $avg.width

[1] 0.07634027

3.6 Cluster plot using clusplot():

```
clusplot(fanny(x = df_Scaled, k = 3, memb.exp = 3, metric = "euclidean
                clusplot(    stand = FALSE, maxit = 500))
```

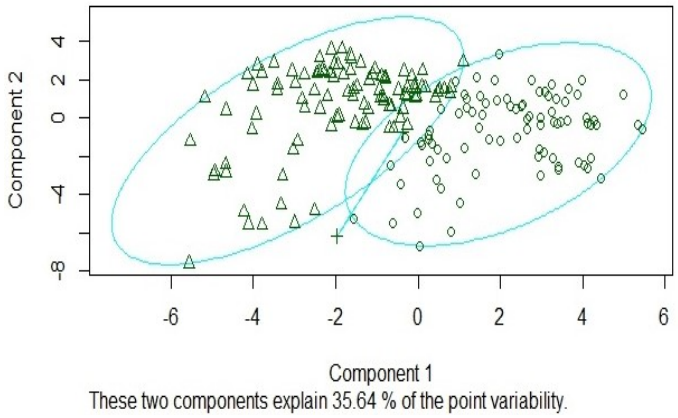

Figure 1: Graph for Fuzzy Clusters for Complete dataset

3.6.1 Fuzzy Cluster plot for a subset of a given dataset

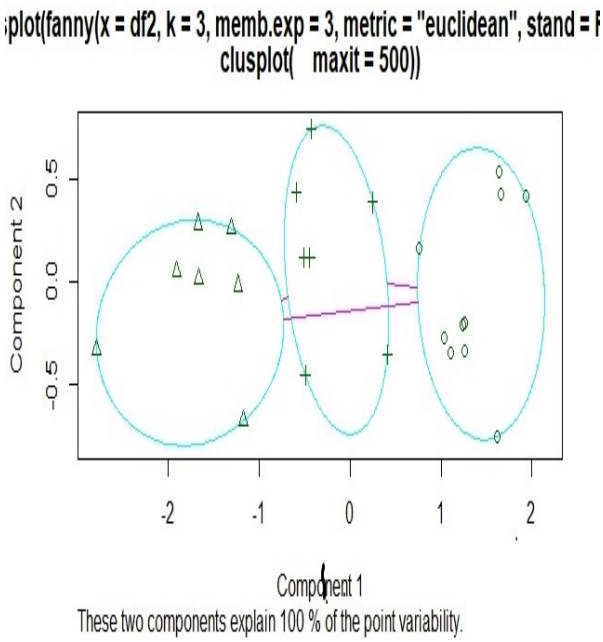

Figure 2: Graph for Fuzzy Clusters for a sample small subset

3.6.2 Correlation Graph for Sample Subset:

```
corrplot(res.fanny$membership, is.corr = FALSE)
```

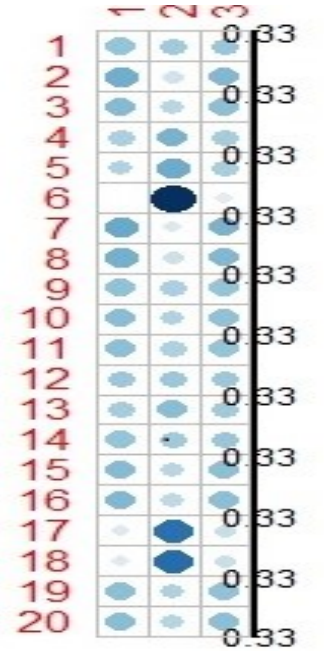

Figure 3: Graph for correlation plot for a sample small subset

3.7: Model based clustering Implementation: We used MClust() function for Clustering its

R-Syntax is:

>Mclust(data,G=Null,Model names = Null, Proir = Null, ….), Mclust() is the optimal model according to BIC(Bayesian Information Criterion) for EM initialized by hierarchical clustering for parameterized Gaussian mixture models[17].

3.6.1>df2.res <-Mclust(df_Scaled,3) fitting ...

```
|=========================================================================
=| 100%
> summary(mc)
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust EEE (ellipsoidal, equal volume, shape and orientation) model with 3
components:

 log.likelihood   n  df      BIC       ICL
     -6450.496 188 662 -16367.52 -16369.83

Clustering table:
 1  2  3
92 11 85
3.6.2 # Values returned by Mclust()
names(mc)
 [1] "call"           "data"           "modelName"      "n"
 [5] "d"              "G"              "BIC"            "bic"
 [9] "loglik"         "df"             "hypvol"         "parameters"
[13] "z"              "classification" "uncertainty"
> mc$modelName
[1] "EEE"

> mc$G
[1] 3
> head(mc$z)
            [,1]        [,2]        [,3]
[1,] 9.999947e-01 1.052048e-09 5.335908e-06
[2,] 1.273719e-05 9.999871e-01 1.246669e-07
[3,] 8.075497e-06 9.999916e-01 3.445118e-07
[4,] 9.999999e-01 4.310888e-08 4.577186e-08
[5,] 9.994202e-01 1.276561e-07 5.796828e-04
[6,] 2.162908e-07 2.532855e-10 9.999998e-01
> head(mc$classification, 10)
 [1] 1 2 2 1 1 3 1 3 2 1
> head(mc$uncertainty)
[1] 5.336960e-06 1.286186e-05 8.420009e-06 8.888074e-08 5.798104e-04
[6] 2.165441e-07
```

3.8 Model-based clustering results can be drawn using the function plot.Mclust():
plot(x, what = c("BIC", "classification", "uncertainty", "density"),
xlab = NULL, ylab = NULL, addEllipses = TRUE, main = TRUE, **...**)[5]

3.8.1# BIC values used for choosing the number of clusters[15]
plot(mc, "BIC")

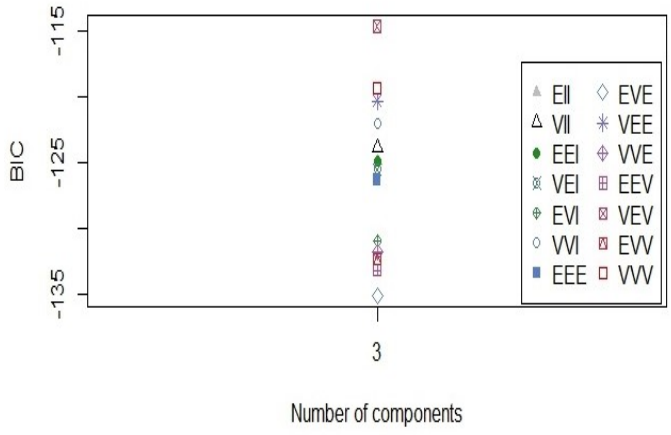

Figure 4: Graph for plot(mc,"BIC")

3.8.2 Classification:
plot(mc,"classification")

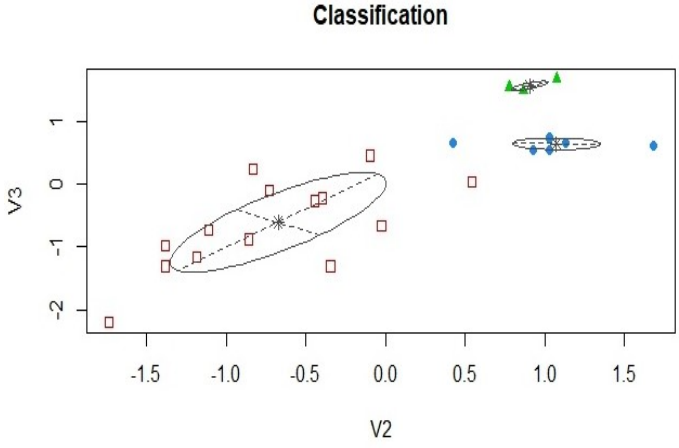

Figure 5: Graph for plot(mc,"classification")

3.8.3 Classification uncertainty
plot(mc, "uncertainty")

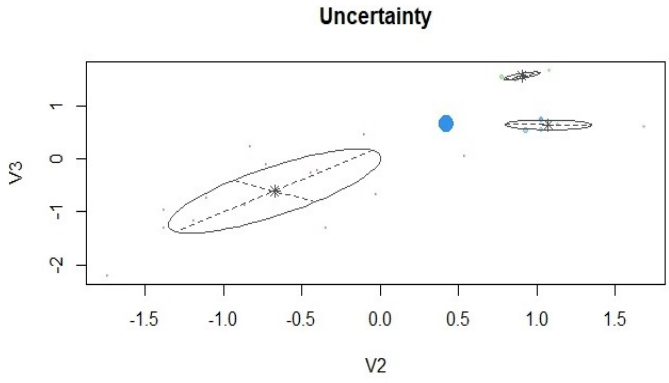

Figure 6: Graph for plot(mc, "uncertainty")

3.8.4 Estimated density. Contour plot
plot(mc, "density")

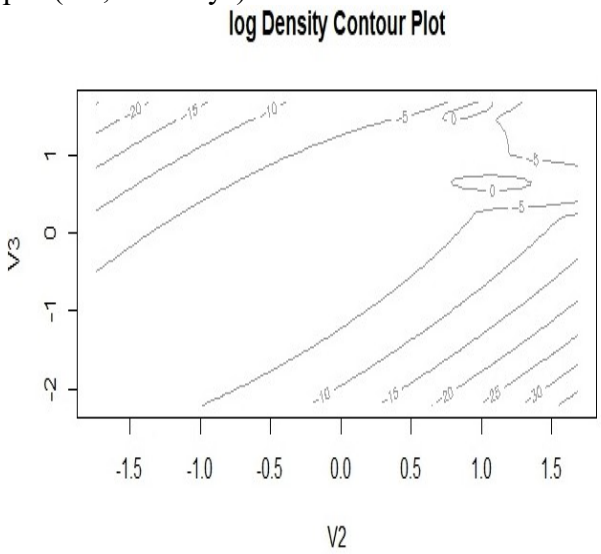

Figure 7: Graph for plot(mc, "density")

4.0 Conclusion

Both Fuzzy and Model based clustering methods are better approaches for accurate and efficient clusters. Here our thyroid drug data set with associated physio chemical and enzyme inhibition properties are clustered and then analyzed the quality and appropriateness of each cluster as shown in the below table[16].

Mclust()	BIC	Classification	Uncertainity	Density
Parameter	Yes	Yes	Yes	Yes
Fanny()	“Membership”	“Dunn Coefficient”	“memb.exp”	“clustering”
Values	0.333333	0.333333	3	Yes

Fanny()	"objective and tolerance"	"convergence"	"dissi"	"call"
Values	8.262558 & 1.000000e-15	11 Iterations, I Convergence, 500 Max Interations	Null	Syntax Mclust()
Fanny()	"k.crisp"	"silinfo"	"data"	
Values	3	Avg:0.07634077	Thyroid data	

Table:1 Fuzzy Vs MClust Parameters

By analyzing the data using BIC, Classification, Uncertainity and density parameters the appropriateness of correct assignment to a relevant cluster/class is more qualitative approach.

Fuzzy clustering is a soft clustering which shows most likelihood based on the probability and it is also good method of clustering by evaluating dunn coefficient, silinfo, k,crisp,membership, objective etc function parameters.

Future extension is possible to evaluate multiple antiviral drugs available to predict to control effectively the pandemic corona virus(novel Sars2 Virus-2019) which dangerously collapsing the entire world both human life and economy also.

5.0 References:

[1] Chris Fraley, A. E. Raftery, T. B. Murphy and L. Scrucca (2012)." mclust Version 4 for R: Normal Mixture Modeling for Model-Based Clustering, Classification, and Density Estimation",Technical Report No. 597, Department of Statistics, University of Washington.

[2]. Chris Fraley and A. E. Raftery (2002). "Model-based clustering, discriminant analysis, and density estimation", Journal of the American Statistical Association 97:611:631.

[3]. http://www.sthda.com/english/wiki/factoextra-r-package-quick-multivariate-data-analysis-pca-ca-mca-and-visualization-r-software-and-data-mining.

[4]. www.cran.r-project.org
[5].www.rstudio.com
[6].Hartigan, J. A.; Wong, M. A. (1979), "Algorithm AS 136: A K-Means Clustering Algorithm". Journal of the Royal Statistical Society, Series C 28 (1): 100–108

[7].Milligan GW, Cooper MC,"An examination of procedures for determining the number of clusters in a data set", Psychometrika. 1985 Jun 27;50(2):159-79

[8].J. C. Bezdek, "*Pattern Recognition with Fuzzy Objective Function Algorithms ",*(Plenum Press, New York, 1981.

[9]. http://www.malacards.org/
[10] .www.drugbank.ca

[11]. Gustafson E.E., Kessel W.C., 1978,"Fuzzy clustering with a fuzzy covariance matrix", Proceedings of the IEEE Conference on Decision and Control, pp. 761-766

[12].Ferraro M.B., Giordani P., 2013."A new fuzzy clustering algorithm with entropy regularization", Proceedings of the meeting on Classification and Data Analysis (CLADAG)

[13]. "The art of R Programming", Norman Matloff, William Pollack, 2013,I edition .

[14]. Data mining Concepts and Techniques, Han & Kamber, Morgan Kaufmann(Elsevier)

[15]. www.sthda.com

[16].www.datanovia.com

[17].www.rstudio-pubs-static-s3.amazonaws.com

[18]. Katikireddy Srinivas, Dr K V D Kiran,"Computational Approach to Overcome Overlapping of Clusters by Fuzzy k-Means" International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-7 Issue-4S2, December 2018."

[19].Katikireddy Srinivas, Dr K V D Kiran "A novel hybrid k-means-k-medoids algorithm as an efficient method of Clustering for Thyroid disease drug database using R "in International Journal of Sciences and Resarch(IJSR),Volume no:73 and Issue no 8,August 2017 (Doi:10.21506/j.ponte.2017.8.51), Ponte Publishers,Italy.

[20].Katikireddy Srinivas, Dr K V D Kiran ,Performance Analysis of Hybrid Hierarchical K-Means Algorithm Using Correspondence Analysis for Thyroid Drug Data"in Journal of Advanced Research in Dynamical and Control Systems, Volume 10,12-Special Issue,August 2018

[21].Katikireddy Srinivas, Dr K V D Kiran "A Novel Hybrid Clustering System using k-means, k-medoids, hierarchical, Fuzzy C Means Algorithms on Thyroid Drug Data using R , International Journal of Advanced Science and Technology Vol. 29, No. 5, (2020), pp. 9480-9492."