

Comparision Results

Parismita Das

5 March 2017

Comparison between rioja and adjClustBand_heap

To check the validity of the cWard's Heap algorithm that is applied for the function adjClustBand_heap, we compare the clustering results of adjClustBand_heap with the results of rioja chclust function.

We compare whether the merging of clusters happen at same height and for same index for both the functions. Hence analysing below plots and table data we verify the correctness of adjClustBand_heap.

Comparing the Merge Data Table of rioja and adjClustBand_heap.

In rioja, the function chclust takes input as dissimilarity matrix and gives 'hclust' object as output. The 'hclust' object has attributes such as the indices of the clusters that are merged and the height at which clusters are merged.

Considering the dissimilarity matrix as the same given in the file dissimilarity2.txt. We get the merge data as:

	V1	V2	height
1	-30	-31	0.01
2	-81	-82	0.02
3	-28	-29	0.03
4	-96	-97	0.04
5	-3	-4	0.07
6	-127	-128	0.10

From the table we can say that the (30,31) merges at height 0.01, (81,82) at 0.02 and so on.

Considering function adjClustBand_heap, it takes similarity matrix as input such that the diagonal elements are 1 and gives 'hclust' object as output. Hence it also has same attributes as rioja chclust function. Thus we can compare the values of Merge Data.

adjClustBand_heap Data Table with maximum band similarity as shown below:

	V1	V2	gain	height
1	-28	-29	0.001064484	0.001064484
2	-30	-31	0.001307686	0.002372169
3	-40	-41	0.002497098	0.004869267
4	-3	-4	0.002729742	0.007599009
5	-35	-36	0.003671523	0.011270532
6	-23	-24	0.004680510	0.015951043

We realise that the merge data are not same, the reason is that we converted the dissimilarity to similarity using a different approach than rioja does. Hence the output we get are different.

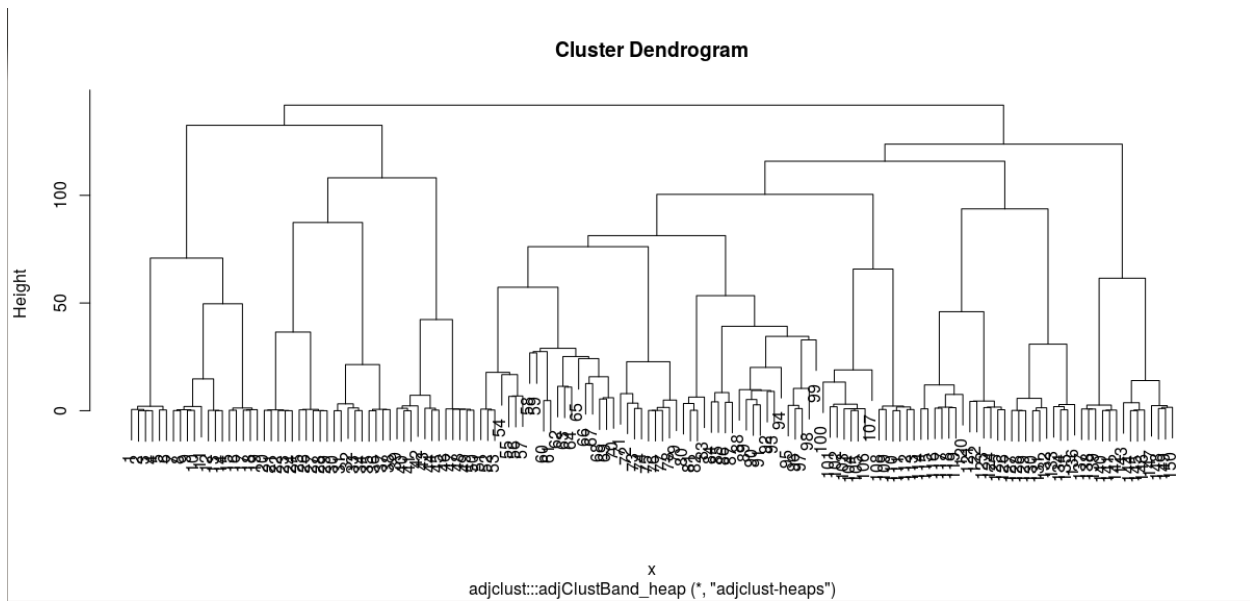
But if we consider the dissimilarity matrix such that it is $2-2*s'$, where s' is scaled similarity matrix (with diagonal as 1). We get:

	V1	V2	height
1	-28	-29	0.001064484
2	-30	-31	0.002372169
3	-40	-41	0.004869267
4	-3	-4	0.007599009
5	-35	-36	0.011270532
6	-23	-24	0.015951043

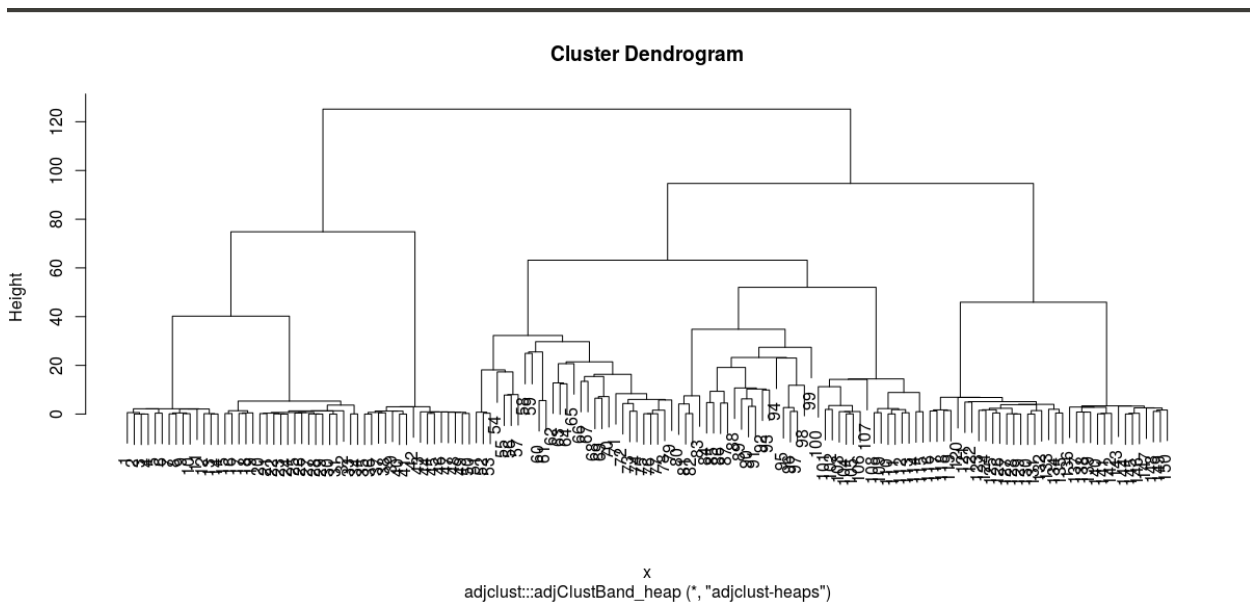
From the table we can say that the merge data of both `chclust` and `adjClustBand_heap` function are same when we take dissimilarity matrix as $2-2*s'$ and maximum band of similarity matrix for `adjBandClust_heap`.

Comparing the dendrograms formed by `chclust` and `adjClustBand_heap`

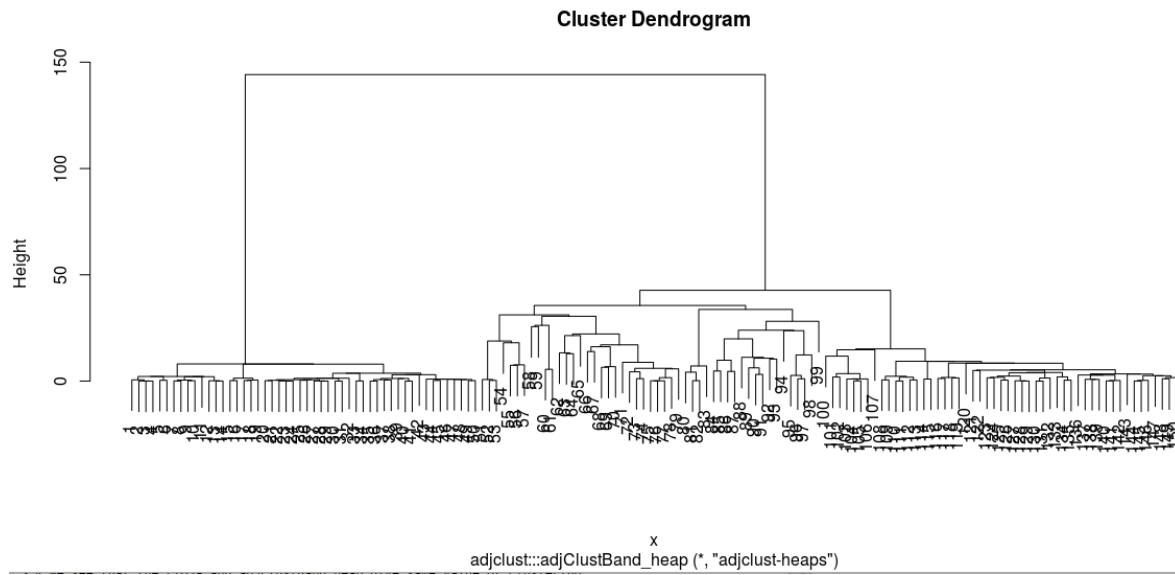
Considering `adjClustBand_heap` dendrograms for various values of the band constrain h , so that we can compare the difference on what happens when we take only values which are in located in the neighbourhood of the diagonal.



adjClustBand_heap Dendrogram for $h = 5$



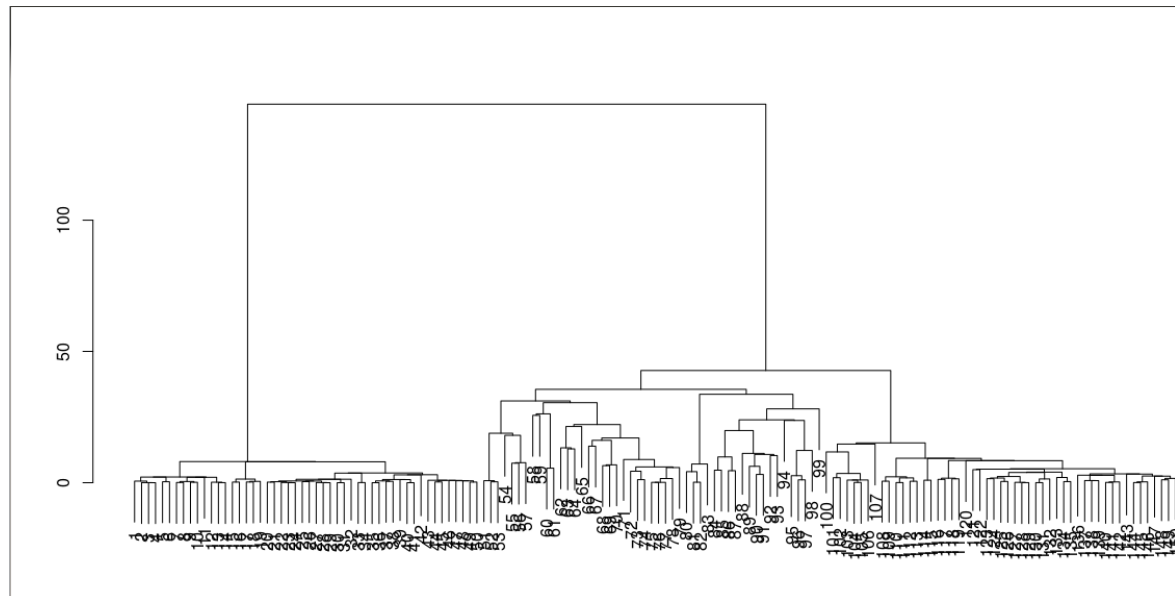
adjClustBand_heap Dendrogram for $h = 20$



adjClustBand_heap Dendrogram for h = maximum height of the matrix

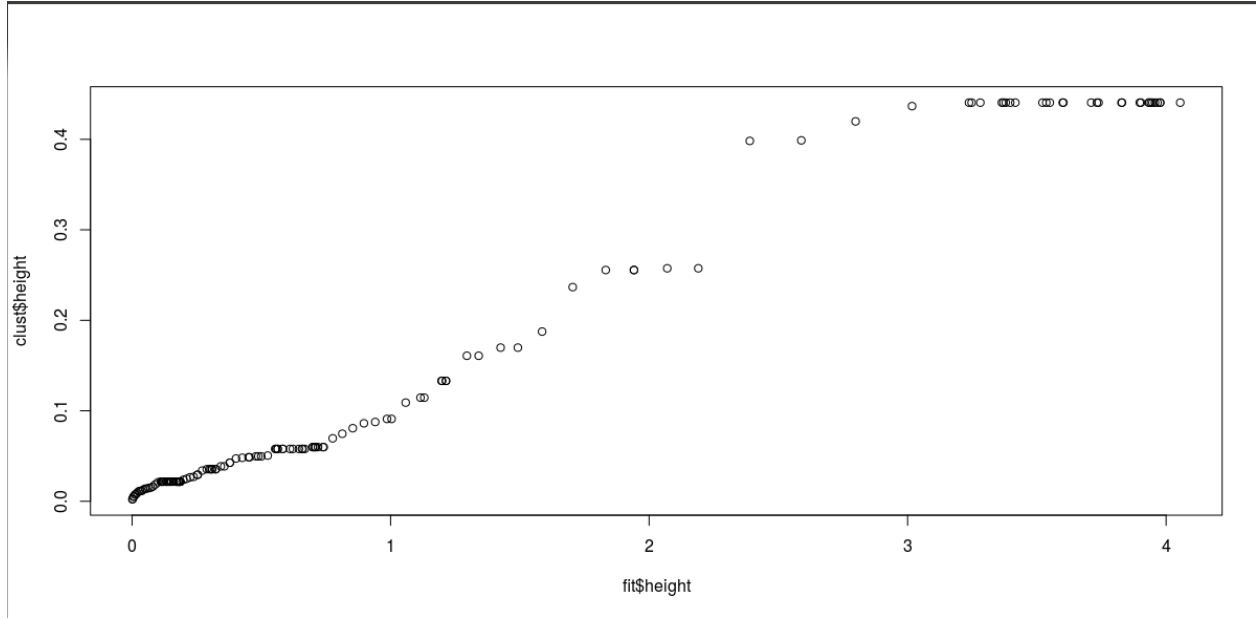
We get to know from the dendrograms that larger the value of h , more similar it is to clusters formed by rioja.

Considering dendrograms formed by chclust with dissimilarity as $2 \cdot 2^s$:



We can see that we get exact same result from rioja and adjClustBand_heap when full band is considered.

However there is slight difference between the heights of clusters that are merged via chclust and adjBanc-Clust_heap. To show the relation between the heights we see the scatter plot between there heights:



From the plot , as we can see its almost a straight line till a certain height, thus the adjBancClust_heap single linkage method is working fine and the height obtained is proportional to that obtained by rioja.

But after a certain height ie, 3 for adjBancClust_heap and 0.4 for chclust. The chclust function of rioja approximates all height above 0.4 to be at 0.4, thus we get a constant slope plot as seen.

In adjClustBand_heap that approsimation is not done hence we get a bit different result than rioja as seen in dendrograms formed.

Hence we can conclude that the adjClustBand_heap gives same result as that of rioja thus proving its correctness and uses a more optimised algorithm.