Summary of this week

Recently, I've been tired of tempting different gene selection methods, different clustering methods, different parameters, etc., to obtain appropriate consensus clustering results. Finally something interesting was found, and I'm trying to summarize them as follows.

PAC score

It's not trival to select a plausible estimated number of clusters for any clustering methods. There exist some works offering some heuristic methods to do this. **PAC**(Proportion of Ambiguously Clustered pairs), especially the rounded PAC score, is a relatively reasonal standard for consensus clustering methods.

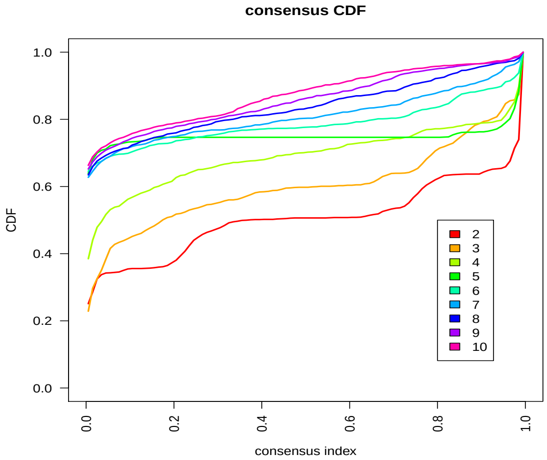
Each element of the consensus matrix represents the proportion of a sample pair being clustered into the same cluster during the re-sampling procedure, with 0 meaning this pair never appears in the same cluster, and 1 meaning this pair always appear in the same cluster.

So we set two thresholds,

$$u_1 = 0.1, \qquad u_2 = 0.9$$

and PAC is the proportion of sample pairs locating in the region $[u_1, u_2]$. So samller PAC means more stable clustering results.

Finally, we use the K with the samllest PAC score. For example, in the following graph, $K = 5$ will be selected.



Implementation

Next, I implement consensus clutering on the BREAST cancer datasets. Some packages I use are listed as follows.

- **'ConsensusClustePlus'**. This is a Bioconductor-based package, which implements all regular inner clustering methods for consensus clustering. Something strange exists in this package, although. It offers interface for users-defined clusterint methods, which, however, are only allowed to accept distance matrix and number of clusters as input. This, of course, can fit most clustering methods, but not all, such as the basic kmeans methods. So why do we modify the basic kmeans, just to look for troubles? Unfortunately, the kmeans embedded inside the package is the default parameters of **R**'s kmeans function, which sets $iter.max = 10$, $nstart = 1$, using only one random start, and 10 iterations. This has great possibilies not to converge.

- **'curatedBreastData'**. This package collects 34 public available breast cancer datasets. This package contains also a function filterAndImputeSamples to impute NAs in the gene expression matrix.

During the debug process, I find the following parameters may have some substantial effects:

- The number of genes selected and the selection strategy

- The inner clustering algorithms

- The implementation of clustering algorithms

- ...

Before I elaborate on some experiment results, I must admit by adjusting the above parameters alone or all, I didn't obtain the expected results. It may seem weird to me,that by accident, I checked the detailed implementations of **COINCIDE** package, and they use $pItem = .9$ instead of the 0.8 I used. This parameter controls the proportion of re-sampled samples, i.e. there will be $pItem \times N$ samples sampled from the original datasets. I don't know why this is, but I obseved in my experiments.

Now, for others, I should summarize some heuristic strategies. The main inner clustering methods include kmeans and hierarchical clutering. The former is good, but we should use 100 iterations. The latter should be used carefully. The 'complete' linkage method is should not be used, which usually clusters most patients in one cluster and leaves extreme small number of patients.
I will only use kmeans in the following discussions. I summarize results in the table.

NO.	NAME	Samples	Gene	Optimal K	PAC
1	GPL1223_all(1379)	60-2	0.05	4	0.08
			0.1	5	0.006
			0.5	5	0.17
2	GPL96_all(2034)	286-5	0.05	2	0.05
			0.1	7	0.11
			0.5	7	0.09
3	GPL3558_all(4913)	50-2	0.05	7	0.11
			0.1	6	0.17
			0.5	2	0.08
4	GPL3883_all(6577)	88-4	0.05	4	0.11
			0.1	4	0.13
			0.5	4	0.14
5	GPL_5049_all(9893)	155-30	0.01	10	0.27
			0.1	10	0.26
			0.5	4	0.19
6	GPL5186_all(12071)	46-2	0.05	-	-
			0.1	10	0.14
			0.5	3	0.12
7	GPL96_all(12093)	136-10	0.05	2	0.107
			0.1	2	0.107
			0.5	2	0.05
8	GPL570_all(16391)	48-2	0.05		
			0.1	10	0.16
			0.5	2	0
9	GPL570_aLL(16446)	114-6	0.05		0
			0.1	2	0
			0.5	2	0.02

10	GPL_JBI(17705)	103-2	0.05	6	0.16
			0.1	10	0.18
			0.5	2	0.17
11	GPL_MDACC(17705)	195-11	0.05	2	0.07
			0.1	3	0.08
			0.5	3	0.09
12	GPL570_all(18728)	21	-	-	-
13	GPL570_all(19615)	115-5	0.05	2	0.02
			0.1	2	0.06
			0.5	2	0.09
14	GPL570_all(19697)	24	-	-	-
15	GPL96_all(20181)	53-2	0.05	4	0.19
			0.1	10	0.17
			0.5	10	0.16
16	GPL96_all(20194)	261-11	0.05	7	0.10
			0.1	2	0.07
			0.5	6	0.08
17	GPL6480_all(21974)	32	-	-	-
18	GPL1390_all(21997)	35	-	-	-
19	GPL5325_all(21997)	28	-	-	-
20	GPL7504_all(21997)	31	-	-	-
21	GPL1708_all(22226)	128-9	0.05	4	0.12
			0.1	3	0.11
			0.5	7	0.12
22	GPL4133_all(22226)	20	-	-	-
23	GPL5325_all(22358)	122-8	0.05	4	0.08
			0.1	4	0.07
			0.5	5	0.12
24	GPL5325_all(23428)	16	-	-	-
25	GPL96_MDACC(25055)	221-12	0.05	2	0.06
			0.1	3	0.06
			0.5	3	0.03
26	GPL96_MDACC(25055)	6	-	-	-

27	GPL96_LBJ(2506 5)	17	-	-	-
28	GPL96_MDACC(25065)	71-2	0.05	2	0.03
			0.1	2	0.03
			0.5	2	0
29	GPL96_MDACC_MDA(25065)	15-1	-	-	-
30	GPL96_PERU(25065)	25-1	-	-	-
31	GPL96_Spain(25065)	16-2	-	-	-
32	GPL96_US0(25065)	54-1	0.05	2	0.04
			0.1	2	0.04
			0.5	2	0.07
33	GPL570_all(32646)	115-7	0.05	3	0.02
			0.1	2	0.07
			0.5	10	0.16
34	GPL570_all(33658)	11	-	-	-

Basically, we can say there's no significant effects of the number of genes selected if PAC is not very large.

Note: I check the consensusClusetrPlus.R and find that they adopt a replace-free re-sampling strategy. So stange...

All my results are stored on the server: /data/home/dwang/project/conceptInstance/ConsensusClusterPlusImplementation