RNA_Seq data Analysis

Breast cancer Substage classification using Support Vector Machine. The analysis is done in some steps. First the whole RNA_seq and miRNA gene data is downloaded from TCGA web portal. Now TCGA is not working any more. The data base is marged to GDC web portal. Same data can be downloaded from Genomic Data Commons (GDC) web portal.
The link is: https://gdc-portal.nci.nih.gov
After donloading the data it needs to pre process prior to classify. Processing steps are shown in the documents. Two different system of breast cancer classification is evaluated here.
1. Surveillance Epidemiology End Result (SEER)
2. Tumor Node Metastasis (TNM)

Here I used primary and advanced stage of breast-cancer patients sample for both system. Metastasis stage is not included here.

Step1: Download BRCA Clinical data

step 2:

Run Rcode named ***barcode_collection.R*** and Choose the file below.
nationwidechildrens.org_clinical_patient_brca.txt
there are lots of variables / columns. Check pathological stages of breast cancer. The output shows like

```
> table(cli_pat[41])

   [Discrepancy]  [Not Available]   CDE_ID:3203222 pathologic_stage          Stage I
               6                5                1                1               90
        Stage IA         Stage IB         Stage II        Stage IIA        Stage IIB
              86                7                6              358              257
       Stage III       Stage IIIA       Stage IIIB       Stage IIIC        Stage IV
               2              155               27               65               20
         Stage X
              13
>
```

Figure1 : clinical information

Choose stages I, IIA, IIB and IIIA . These stages are chosen because of majority sample size.
Save the barcode of each stage in different txt file. For GDC web portal download all the four stages patients genedata. You can download any kind of data what you like . In this case I downloaded only RNA_Seq data.
Platform -Illumina HiSeq v2,
After you download all the data, save them in four different directory.

Step 3:
Run the module ***datacollection.R*** to fetch the gene data. This module takes only raw_count information. File name of each file is split in two parts. First part represents sample id and second part

defines file extension. The whole matrix column name is replaced by file name as sample id such that we can find the duplicate sample. This module also checks whether any sample is taken 2 times. For example any sample can have more tumor information together with BRCA. Each filename looks like:

"unc.edu.00ee8acd-0841-4c85-a99a-8966a6828dd7.1150156.rsem.genes.results"
" rsem.genes.results" is removed from the filename and taken as sample id.

So sample id looks like:
"unc.edu.00ee8acd-0841-4c85-a99a-8966a6828dd7.1150156"

Each file looks like below table. It has four variables. I took only raw_count .

Filter

	gene_id	raw_count	scaled_estimate	transcript_id
1	?|100130426	0.00	0.000000e+00	uc011lsn.1
2	?|100133144	11.29	3.654012e-07	uc010unu.1,uc010uoa.1
3	?|100134869	11.71	2.769793e-07	uc002bgz.2,uc002bic.2
4	?|10357	103.69	7.431271e-06	uc010zzl.1
5	?|10431	2507.00	8.454465e-05	uc001jiu.2,uc010qhg.1
6	?|136542	0.00	0.000000e+00	uc011krn.1
7	?|155060	330.00	3.948426e-06	uc003wfr.3,uc003wft.3,uc003wfu.2,uc011kup.1
8	?|26823	2.00	1.770213e-07	uc011mlh.1
9	?|280660	0.00	0.000000e+00	uc010nib.1
10	?|317712	0.00	0.000000e+00	uc010ihw.1
11	?|340602	0.00	0.000000e+00	uc004dpj.2
12	?|388795	0.00	0.000000e+00	uc010zub.1
13	?|390284	12.00	2.381338e-06	uc001qoa.2

Showing 1 to 13 of 20,531 entries

Figure 2: data file for each sample looks like this

Step 4:

Run the module ***data_integration.R*** .This module combines all stages of genedata in one big matrix. Where column represents sample id and row represents gene_id. It also creates one data set with equal length of sample id. Later on both of the data set will be used for classification.

Step 5:

Run module ***Zeros_remove.R*** . this module removes zero value column. That means gene raw_count zeros in all of the sample is meaningless. So removed from the data set. This module process 2 sets of data and save them in a different file such that we can get back original one if necessary.

Step 6.
Run ***split_gene_name/id.R*** to split gene name in 2 columns gene_id(numerical) and gene name. We need gene_id for Gene Ontology(GO) analysis(i.e separate protein coding and non-coding genes). It also makes 2 dataset for further use one with gene name and gene id . Other one without gene name. This one we will use to GO analysis.

Filter

	unc.edu.059c8834-efd5-447c-97c6-49fd68c09c0d.1140716	unc.edu.08236177-a046-4cfe-a11c-a24292a1f777.1134035	unc.edu.0ef20000-2c8e-47bb-86e5-ffa2814099a7.1139467	unc.edu.14404fa2-adc4-4ecd-b4ee-5f4a75b54ff2.1116678	unc.edu.3599-4ae6e28c
?|100130426	0.00	0.00	0.00	0.00	
?|100133144	11.56	15.73	8.76	1.63	
?|100134869	8.44	7.27	2.24	1.37	
?|10357	277.79	262.98	169.38	128.27	
?|10431	3136.00	1395.00	1863.00	1285.00	
?|155060	248.00	313.00	284.00	247.00	
?|26823	0.00	0.00	0.00	0.00	

Figure3 : Original data

Filter

	Gene_Name	1	2	3	4
100130426	?	0.00	0.00	0.00	0.00
100133144	?	11.56	15.73	8.76	1.63
100134869	?	8.44	7.27	2.24	1.37
10357	?	277.79	262.98	169.38	128.27
10431	?	3136.00	1395.00	1863.00	1285.00
155060	?	248.00	313.00	284.00	247.00
26823	?	0.00	0.00	0.00	0.00
280660	?	0.00	0.00	0.00	0.00

	1	2	3	4	5
100130426	0.00	0.00	0.00	0.00	
100133144	11.56	15.73	8.76	1.63	
100134869	8.44	7.27	2.24	1.37	
10357	277.79	262.98	169.38	128.27	
10431	3136.00	1395.00	1863.00	1285.00	2
155060	248.00	313.00	284.00	247.00	1
26823	0.00	0.00	0.00	0.00	
280660	0.00	0.00	0.00	0.00	

Figure 4 : 2 data set created

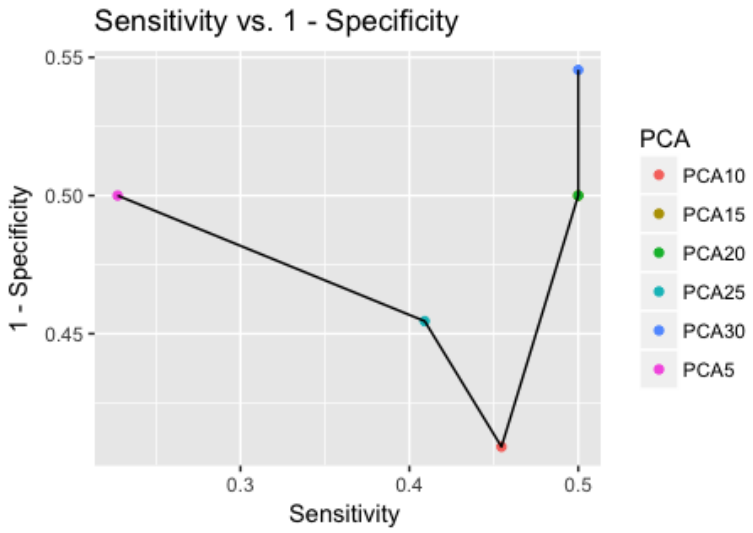
Step 7:
Run ***GO_Analysis.R*** to make 2 data set protein-coding and Non-coding RNA_seq data.
This module makes total 4 data set 2 for each data set since we have now 2 data sets created.
Module now load *filtered_Gname_ID_data.Rdata* to get protein coding and noncoding data. Later it will assign gene name as column name and sample number as row name. This way it will create four data sets. Namely : PCodingRNA and nonCodingRNA for each data sets.

Filter

	AAA1	AACSL	ABCA17P	ABCC6P1	ABCC6P2	ADAM21P1	ADAM3A
2	0.00	2.00	15.00	51.00	68.00	4.00	0.00
3	0.00	15.00	8.00	0.00	52.00	5.00	0.00
4	0.00	0.00	13.00	3.00	15.00	1.00	0.00
5	0.00	5.00	7.00	25.00	24.00	1.00	0.00
6	0.00	0.00	29.00	5.00	16.00	10.00	0.00
7	0.00	0.00	12.00	28.00	178.00	1.00	0.00

Figure 5: sample data of Non-codingRNA

Step 8:
Run ***S_S_plot_PCA.R.*** module. This module perform PCA and classify the data in 5 different PCA sets then it will check which PCA set is better perform in SVM classifier. Select the number of PCA. Here PCA 15 is better . Since sensitivity and1-specificity are same.



Step 9: Classification Results
unbalanced sample
gene sampled:
ProteinCoding gene = 4488
non-coding gene = 351

TNM system performance

5 fold LOOCV	Accuracy	Sd_error/Acc	sensitivity	Sd_error/sen	specificity	Sd_error/spc
ProteinCoding	42.11 %	0.017	26.88%	0.003	76.06%	0.001
non-coding	42.42 %	0.015	27.24%	0.008	76.31 %	0.001
miRNA	45.14 %	0.036	25.00%	0.0000	75.00 %	0.0000

SEER system performance

5 fold LOOCV	Accuracy	Sd_error/Acc	sensitivity	Sd_error/sen	specificity	Sd_error/spc
ProteinCoding	57.41%	0.020	59.00%	0.031	55.97%	0.028
non-coding	51.91%	0.015	54.44%	0.024	49.21%	0.015
miRNA						

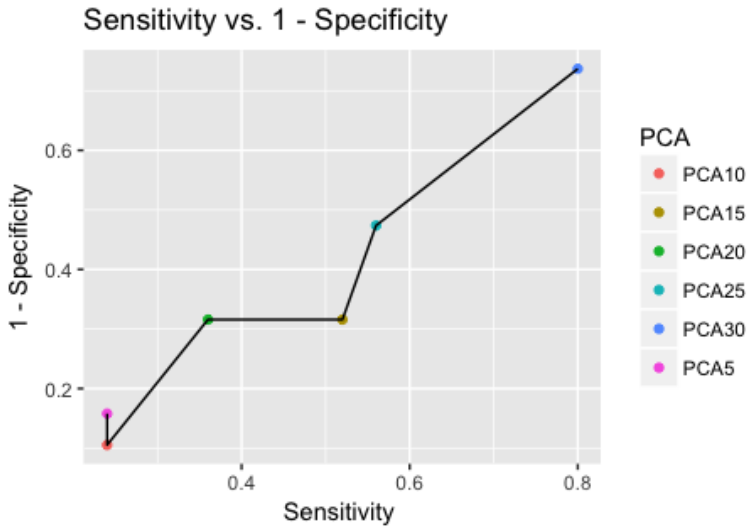
Balanced sample

TNM system performance

5 fold LOOCV	Accuracy	Sd_error/Acc	sensitivity	Sd_error/sen	specificity	Sd_error/spc
ProteinCoding	29.48%	0.0130	30.76%	0.017	76.83%	0.005
non-coding	25.24%	0.019	26.36%	0.024	75.36%	0.007
miRNA	48.27 %	0.019	29.20%	0.009	77.91%	0.004

SEER system performance

5 fold LOOCV	Accuracy	Sd_error/Acc	sensitivity	Sd_error/sen	specificity	Sd_error/spc
ProteinCoding	58.78%	0.012	56.67%	0.025	60.83%	0.032
non-coding	53.40%	0.018	56.93%	0.0223	49.80%	0.018
miRNA	62.12%	0.016	71.68%	0.040	52.49%	0.042



N.B: Data can be sent by email upon request
email: facihul@yahoo.com