COMP790 Tech Report:

- Analyzing DNA Methylation based on Map-Reduce

ZHI DONG UNC Chapel Hill

1 Introduction:

1.1 TCGA:

The Cancer Genome Atlas (TCGA) is a comprehensive and coordinated effort to accelerate our understanding of the molecular basis of cancer through the application of genome analysis technologies, including large-scale genome sequencing. The mission is to accelerate the understanding of the molecular basis of cancer through the application of genome analysis technologies, including large-scale genome sequencing. TCGA researchers analyze tumor and normal tissue from hundreds of participants for each cancer selected for study. This provides the statistical power needed to produce a complete genomic profile of each cancer, which is crucial to identifying those genomic changes that offer the greatest opportunities for therapeutic development.

The analysis of TCGA is challenged by the fact that traditional analysis tools have difficulty in processing large-scale data from high-throughput sequencing. The open source Apache Hadoop Project, which adopts the Map-Reduce framework and a distributed file system, has recently given bioinformatics researchers opportunity to achieve scalable, efficient and reliable computing performance on Linux clusters and on cloud computing services.

1.2 DNA Methylation Data:

In this proposal, we analyze DNA Methylation Dataset from TCGA. DNA methylation plays a crucial role in the regulation of gene expression and chromatin organization within normal eukaryotic cells. Specifically, the methylation data of cancer cell is important to analyze the cancer type. The sample data is derived from cells suffering the breast cancer.

We use the beta value to evaluate the methylation. The beta value is the ratio of the methylated probe intensity and the overall intensity. The Beta-value statistic results in a number between 0 and 1, or 0 and 100%. Under ideal conditions, a value of zero indicates that all copies of the CpG site in the sample were completely unmethylated (no methylated molecules were measured) and a value of one indicates that every copy of the site was methylated.

The DNA methylation beta value is always followed by the gene position and the gene type. The gene type is defined by several kinds of types, such like. Generally, researchers are interested in which type of gene are more likely to be methylated.

2 Approaches and Implementation:

2.1 One row of data is like this:

cg00000029 0.164466523028052 RBL2 16 53468112

Data explaination:

cg00000029 Sample code:	0.164466523028052 Beta Value	RBL2 Gene Type	16 Chromosome	53468112 Position on Chromosome
Begin with "cg", use it to filter out the heading of the file.	It is "NA" means not available for some reasons, or a value	It is "NA" or some specific strings	It is an integer from 1-22 or a string "X" "Y" means X and Y chromosome.	It is a huge number, so we should use a long variable to stand for.

Surprisingly, this data is extremely regular. We encountered some rows only show 4 attributes as some attributes are simply empty strings. The "NA" condition will be handled in the mapper. The head of the txt will be filtered out by simply checking if it begins with "cg", it is useful when merged several txt files to one big file and get rid of heading lines.

2.2 Data Retrieval:

Researchers would like to retrieve a lot of information from this giant dataset. We list some statistic related data that researchers may concern:

Beta value ranking of all gene types:

Solution: In each row, the beta value of the same gene type is added. The sorting is used to get a ranking list. Since Map-Reduce is able to sort during the mapping and reducing, we don't use the method in assignment 2 which conducts sorting after mapping and reducing.

The distribution of beta value in the range of [0,1].

Solution: 100 intervals has been generated, if the beta value is in the specific interval, then the count for this interval is added.

The probe position clustering of high beta value on specific chromosomes.

Solution: The probe position which has a beta value above a specific value is clustered using K-Mean algorithm, and it is a distributed system version K-Mean.

2.3 Implementation:

The ranking the gene types is based on the sum of all beta values showed in this raw dataset. Map-Reduce provides an existing method for sorting. In our implementation, one job is focus on adding the beta values and the other job is sorting the results. Pipeline in Map-Reduce is leveraged by outputting a file path in the job and then set it as an input filepath for the jobsort.

A brief of overview of the Mapper and Reducer is as follows:

BetaValueAdderMapper	BetaValueAdderReducer	BetaValueRanking

Intermediate Key:
<GeneType, beta Value>
	Add all beta values for specific GeneType.	“job” is the beta adding job which sets the BetaValueAdderMapper as the mapper and BetaValueAdderReducer as the reducer, while jobsort is the sorting job which sets the InverseMapper as the mapper since the value will be compared not the key.
		FileOutputFormat.setOutputPath(job,tempath);
		// do something like adding
		FileInputFormat.addInputPath(jobsort, tempath);
		//sorting.

Number of intervals is large, as we can get the distributions with small interval number easily. The interval number is 100 in our system.

DistributionCountMapper	DistributionCountReducer	DistributionCount

Output key is the interval number
Output Value is 1
	Add all values for specific Interval.	job.setMapperClass(DistributionCountMapper.class);
		job.setReducerClass(DistributionCountReducer.class);

The implementation of clustering need a lot of thoughts. The methodology of K-Means is in data mining, k-means clustering is a method of cluster analysis which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean. This results in a partitioning of the data space into Voronoi cells. The goal is to minimizing the within-cluster sum of squares (WCSS).

In order to cluster the genes with high beta values, in general, we set the threshold as 0.5, if the beta value is larger than 0.5, we marked it as a “hot area” and take this point into the clustering. The goal of the clustering is finding the host areas according to the positions.

The pseudo code is as follows:

```
Pseudo Code of K-means:
Input: K,data[n]
1 select K initial point, say c[n];
2 for data[0]….data[n], compare them with c[n], if it is closest to c[i], marked it as i cluster.
3 for c[i] recalculate the center.
4 repeat 2 and 3, until c[i] is converge.
```

We extend the K-Means to a distributed system version. The architecture and data flow is as below:

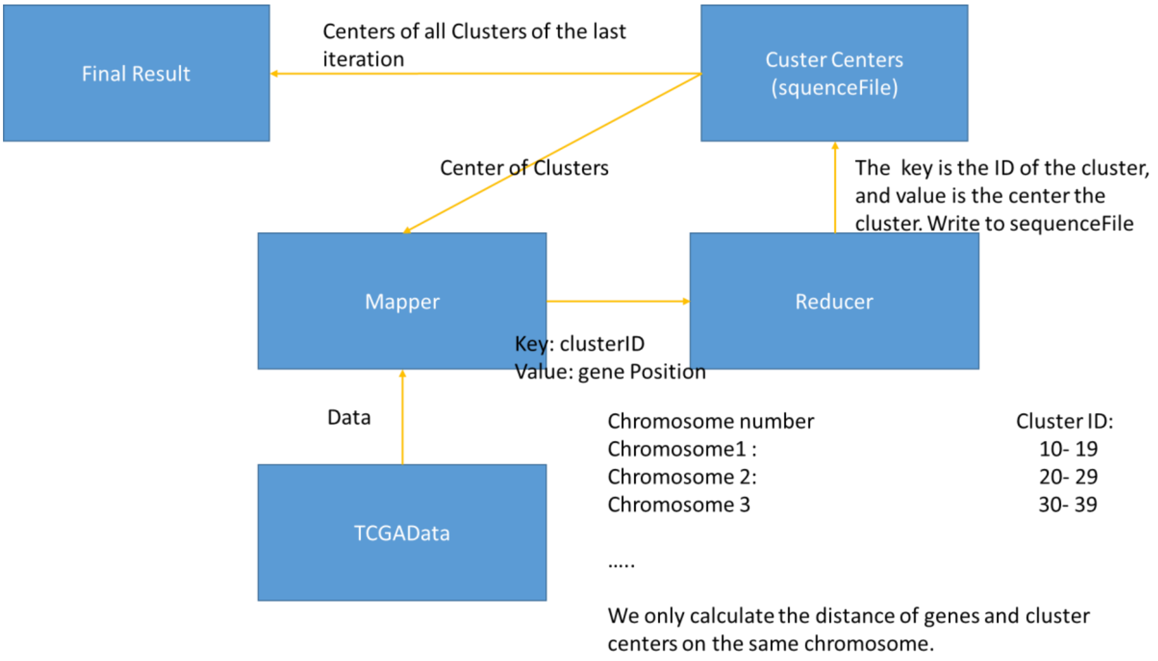
Final Result
Centers of all Clusters of the last iteration
Custer Centers (squenceFile)
Center of Clusters
The key is the ID of the cluster, and value is the center the cluster. Write to sequenceFile
Mapper
Reducer
Key: clusterID
Value: gene Position
Data
TCGAData
Chromosome number
Chromosome1 :
Chromosome 2:
Chromosome 3
.....
Cluster ID:
10- 19
20- 29
30- 39
We only calculate the distance of genes and cluster centers on the same chromosome.

Fig.1 Architecture and data flows of K-Mean implementation

For each pair of chromosomes, we set the maximum number of clusters is 10. So we have maximum 230. We set the first cluster for the first chromosome pair is 10, 10-19 clusters are assigned to the first chromosome, and sequentially, the rest cluster IDs is allocated to other clusters as Fig.2 shows.

Mapper retrieve data from TCGA raw data, Beta value is checked and compared with the threshold to see if it is a valid gene with high beta value. The gene with high beta value is compared with all clusters center on the same chromosome and generate a <key, value> pair where clusterID is the key and gene position is the value. The cluster centers are stored in a sequence file where the key is the cluster and the value is center. We override the setup method to load the file into the mapper. Hashmap is used to store the cluster ID and center from the file at runtime.

The reducer receive all the clusterID corresponds to the gene positions. Make the average of the gene positions and a new cluster center is generated. The new centers is written into the sequence file for the next iteration.

The main class for the K-Means needs a lot of thoughts. We first guarantee that the file we write in could replace the older one, and therefore the remove/delete file operation is implemented. In the main class has a loop which has a fixed number of iterations, the intermediate results are also stored. At each iteration, a new job is created, and mapper reducer also set to the job. The configuration of the job as the file path name of previous cluster center results. The output of reducer is a set of new centers written into the cluster center sequence file.

We briefly summarize the algorithm:

```
Input. k, data[n] (data should be in dfs)

1 select k initial center points c[], c[] is saved in file: clusterlist(sequence file).
2 start mapreduce, send cluserlist to each node. Input is dfs data, output is dfs_clusterlist
3 Mapper:    input is data[k1…k2], and as for data[k], compared with c[0]…c[n-1], it is marked to i cluster if it is closest to cluster i , output is <i, data[k]> , i is the key and data[k] is the value.

4 reducer:    Since cluster is marked as key, then all data belongs to the same cluster will be inputed to the same reducer. Then we can recalculate c[i], which is { data[j] sum/(number of data marked by i. Then output the result to dfs_clusterlist.

5 check dfs_clusterlist and compared it with original clusterlist. If it coverage, it is terminated, if not, jump to step2. Or used a fixed iterations to do it.
```

3 Results

3.1 Beta Value Distribution

We use our Map-Reduce framework analysis the beta value percentage distribution in Fig.2. The figure 2 shows some interesting findings. There are two peaks, one is around 1%-4%, and the other one is roughly between 94%-97%. The distribution out of the two peaks are fairly smooth and surprisingly no spikes detected. It probably reveal a fact that the DNAs in the breast cancer are either very easy to be methylated or they are very hard to be methylated. This polarization in the beta value will give a fairly useful pattern to predict some diseases.

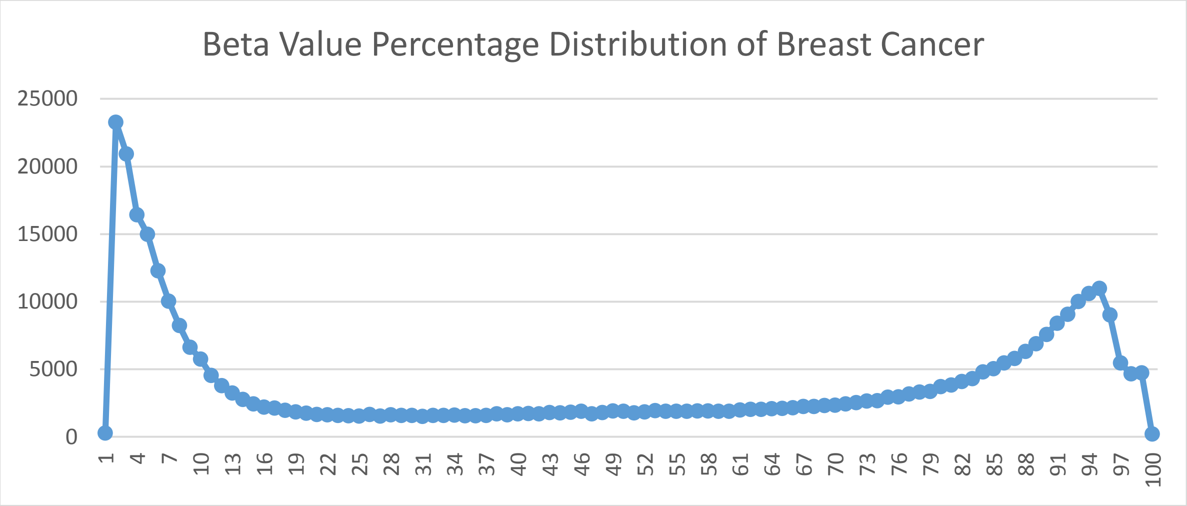

Fig 2. Beta value percentage distribution of breast cancer.

3.2 Beta value ranking of all gene types

We move on to analysis the most active and least active gene types. We first look at the top active gene types. It is obvious that PRDM 16 is pretty important in the DNA methylations, several papers also confirm this judgment such like [1], published on Cell journal.

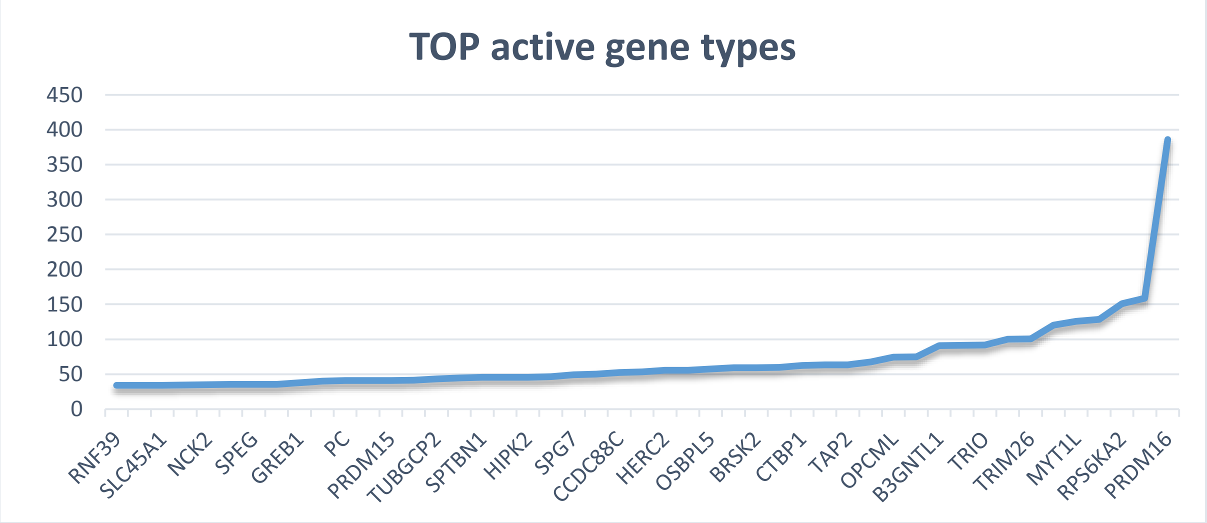

Fig3 Top active gene types.

3.3 Position clustering of gene with high beta value

We look at the high beta value areas across chromosomes, after 20 iterations of K-Means clustering, the figure can give an intuitive view of the space distribution of genes with high beta value:

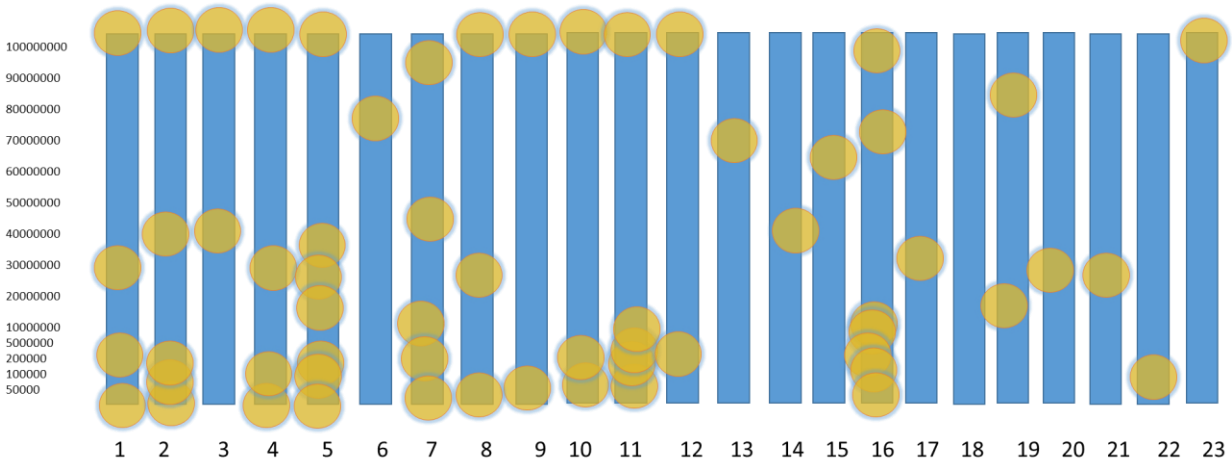

Fig.3 Clustering of high beta values, the horizon axis is the chromosome number. The vertical axis is the gene position.

Intuitively, the more we go to the end of the chromosome, the more easily we can get the DNA methylated. The reason may be DNA at the end of chromosome is not as stable as in the middle.

4 Conclusion:

In this paper, we provide a solution to large bio-data processing based on map-reduce framework. Map-Reduce is powerful in statistic usage such like get the distribution of specific value. It also offers a straightforward way to get items sorted with high efficiency. We also implemented a sophisticated solution to K-Mean launched on Map-Reduce, which is useful if the data is drastically big. We also show some interesting findings in terms of results which are in line with the research of current years.

5 References:

[1]Ineˆs Pinheiro,1,2 Raphae¨l Margueron,3,6 Nicholas Shukeir,1 Michael Eisold,1 Christoph Fritzsch,1 Florian M. Richter,1 Gerhard Mittler,1 Christel Genoud,4 Susumu Goyama,5,7 Mineo Kurokawa,5 Jinsook Son,3 Danny Reinberg,3 Monika Lachner,1 and Thomas Jenuwein1,2, “Prdm3 and Prdm16 are H3K9me Methyltransferases Required for Mammalian Heterochromatin Integrity” Cell 2012.

http://211.144.68.84:9998/91keshi/Public/File/42/150-5/pdf/1-s2.0-S0092867412009385-main.pdf