Cancer subtypes: searching for the signal from within the expression data.

Igor Filipiuk.

Student of Faculty of Mathematics, University of Warsaw. Modelling of Complex Biological Systems.

Abstract

Cancer research is one of the most essential components of medical development in general. Moreover, arguably one of the most important characteristics of disease called cancer is its' variability. This project is aimed at using wide cancer data to tackle the problem of variability and check if unsupervised classification gives any significant or interesting effects against phenotypically established labels.

Type of cancer data that were used to attempt to build classification approach were expression data. Data was obtained from GEMiCCL repository. Arbitrarily I chose 4 types of cancer (lung, pancreas, kidney and thyroid). It included 228 observations with 12 subtypes for lung cancer, 49 observations with 5 subtypes for pancreas cancer, 49 observations and 3 subtypes for kidney cancer and 19 observations with 5 subtypes for thyroid cancer.

In the course of exploratory analysis/preprocessing I attempted to identify and remove batch effects and/or other unwanted latent variable influence on the expression data. Due to the architecture of GEMiCCL platform I decided to design the analysis platform in such a way that it would be possible to compare the results of clustering for the same cancer (e.g. lung) against different microarray platforms they were analyzed (e.g. COSMIC and CCLE platforms).

In the project that was performed by me I tried to check phenotypically classified cancers against unsupervised classification methods. The aim of this project and potentially following ones would be to establish stable approach of using unsupervised classification methods to better identify cancer types based on genetic signals from the data.

The whole pipeline was built using RStudio.

Introduction

Cancer is currently one of the most common causes of death in humans. Among non-communicable diseases, they are the second leading cause of death, immediately after cardiovascular disease. It is estimated that cancer is responsible for ⅛ of all deaths in the world - more than AIDS, malaria and tuberculosis combined. However, if we take into account the differences between regions or countries, the statistics show a clear tendency: infectious diseases such as malaria or AIDS, collect the largest harvests in less developed countries, and in highly developed countries such as the United States of America, a quarter deaths are cancerous. Globally, the number of deaths caused by cancer is about 8.2 million a year and is projected to increase to 13 million a year over the next 20 years. The number of cancer cases detected annually is estimated at around 14 million per year and is expected to increase to 22 million in 20 years.

Cancer is an extremely complicated disease that affects almost all types of tissue. There are currently over 100 different types of cancer, each of which requires appropriate diagnosis and treatment. Characteristics of cancer is a complex and multifaceted problem. So far, 100 clearly defined types of cancers have been distinguished due to specific genotypes. Many current research is directed to a more accurate definition of specific types of cancer in genetic terms, since many of the cancers can be characterized by several genes whose expression has changed. The phenotypic approach seems relatively easier and so far more firmly established in medical community.

With growing importance of molecular diagnostics of biological and medical phenomena I believe that developing tools to group or classify such a dangerous disease as cancer based on molecular signals may be of great value. Following paper will describe my attempt at prototyping a pipeline to process molecular signal available and cluster cancer data into meaningful groups.

Materials and Methods

All the data that was used in this project was from GEMiCCL repository.

Phenotypic data.

All cell line names as downloaded from the Cellosaurus (ftp://ftp.expasy.org/databases/cellosaurus) database. Disease names were selected from the MeSH terms, and tissue names were chosen using information provided by the COSMIC and CCLE databases.

Expression data.

It was specified that GEMiCCL researchers downloaded original microarrays from COSMIC (https://www.ebi.ac.uk/arrayexpress/experiments/E-MTAB-3610, Affymetrix hg U219 array), CCLE (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi? acc=GSE36133, Affymetrixhg U133 plus 2.0 array) and NCI60 (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi? acc=GSE32474, Affymetrix hgU133 plus 2.0 array). Then Affy R package was used for RMA normalization and hg19 genome for gene annotation. Afterwards *ComBat* software was used to remove batch effect. However, due to the

design of the project pipeline, *ComBat* values provided within the dataset was not used.

Dimensionality reduction.

As the number of variables subjected to statistical analysis increases, their accuracy increases, but so does the degree of complexity and difficulty in interpreting the results obtained. Too many variables carry the risk of mutual correlation. So the information brought in by some variables may be redundant, i.e. some variables may not add new information to the analysis, but repeat information already provided by other variables. The need to reduce dimensionality (reduce the number of variables) gave birth to a whole group of analyzes devoted to this issue, such as factor analysis, principal component analysis, cluster analysis, and discriminant analysis. These methods allow detection of relationships between variables. Based on these relationships, you can extract groups of similar variables, and for further analysis select only one representative of each group (one variable), or a new variable whose values are calculated based on the other variables in the group. In this way, we can be sure that the information carried by each group is included in the analysis. With a small loss of information, we can reduce the set p variables to the set k variables where k << p.

For high-dimensional datasets (i.e. with number of dimensions more than 10), dimension reduction is usually performed prior to applying a K-nearest neighbors algorithm (k-NN) in order to avoid the effects of the curse of dimensionality. The curse of dimensionality refers to the situation when the correct classification of objects, using the full data set, is almost impossible, and the multiplicity of characteristics in the vector results in an increase in the number of parameters, which results in an increase in the complexity of the classifier [4]. The risk of overfitting and thus the decrease in the generalizing ability of the classifier also increases. This is the reason for the widespread reduction of dimensionality of features. The reason for the problems is the identification of a subset of features that will be used to correctly classify data by the algorithm.

Clustering methods.

K-means

The k-means method is a method belonging to the group of cluster analysis algorithms, i.e. analysis consisting in searching for and isolating groups of similar objects (clusters). It represents a group of non-hierarchical algorithms. The main difference between non-hierarchical and hierarchical algorithms is the need to provide the number of clusters first.

By means of the k-means method, k different clusters as different as possible will be created. This algorithm involves moving objects from clusters to clusters until variations within and between clusters are optimized. It is obvious that the similarity in the cluster should be as large as possible, while the separate clusters should be as different as possible.

The principle of the algorithm is as follows:

1. We set the number of clusters.
2. We set the initial focus measures.
3. We calculate the distance of objects from the centers of clusters.

4. We assign objects to clusters
5. We set new means of concentration
6. We carry out steps 3,4,5 until the stop condition is met.

Mean-Shift Clustering

Mean shift clustering is a sliding-window-based algorithm that attempts to find dense areas of data points. It is a centroid-based algorithm meaning that the goal is to locate the center points of each group/class, which works by updating candidates for center points to be the mean of the points within the sliding-window. These candidate windows are then filtered in a post-processing stage to eliminate near-duplicates, forming the final set of center points and their corresponding groups.

The principle of the algorithm is as follows:

1. We begin with a circular sliding window centered at a point C (randomly selected) and having radius r as the kernel.
2. At every iteration, the sliding window is shifted towards regions of higher density by shifting the center point to the mean of the points within the window.
3. We continue shifting the sliding window according to the mean until there is no direction at which a shift can accommodate more points inside the kernel.
4. This process of steps 1 to 3 is done with many sliding windows until all points lie within a window. When multiple sliding windows overlap the window containing the most points is preserved. The data points are then clustered according to the sliding window in which they reside.

DBSCAN

DBSCAN is a density-based clustered algorithm similar to mean-shift, but with a couple of notable advantages.

The principle of the algorithm is as follows:

1. DBSCAN begins with an arbitrary starting data point that has not been visited. The neighborhood of this point is extracted using a distance epsilon ε (All points which are within the ε distance are neighborhood points).
2. If there are a sufficient number of points (according to minPoints) within this neighborhood then the clustering process starts and the current data point becomes the first point in the new cluster. Otherwise, the point will be labeled as noise.
3. For this first point in the new cluster, the points within its ε distance neighborhood also become part of the same cluster. This procedure of making all points in the ε neighborhood belong to the same cluster is then repeated for all of the new points that have been just added to the cluster group.
4. This process of steps 2 and 3 is repeated until all points in the cluster are determined i.e. all points within the ε neighborhood of the cluster have been visited and labeled.

EM-GMM

Gaussian Mixture Models (GMMs) give us more flexibility than K-Means. With GMMs we assume that the data points are Gaussian distributed; this is a less restrictive assumption than saying they are circular by using the mean. To find the parameters of the Gaussian for each cluster (e.g the mean and standard deviation), we will use an optimization algorithm called Expectation–Maximization (EM). Then we can proceed with the process of Expectation–Maximization clustering using GMMs.

The principle of the algorithm is as follows:

1. We begin by selecting the number of clusters (like K-Means does) and randomly initializing the Gaussian distribution parameters for each cluster.
2. Given these Gaussian distributions for each cluster, compute the probability that each data point belongs to a particular cluster. The closer a point is to the Gaussian's center, the more likely it belongs to that cluster.
3. Based on these probabilities, we compute a new set of parameters for the Gaussian distributions such that we maximize the probabilities of data points within the clusters. We compute these new parameters using a weighted sum of the data point positions, where the weights are the probabilities of the data point belonging in that particular cluster.
4. Steps 2 and 3 are repeated iteratively until convergence, where the distributions don't change much from iteration to iteration.

Pipeline Design

In the course of preprocessing the data, exploratory analysis, dimensionality reduction and applying clustering algorithms I used RStudio as it was mentioned before. Firstly I downloaded data from GEMiCCL platform, which was phenotypic data called *Cellosaurus information* and expression data.

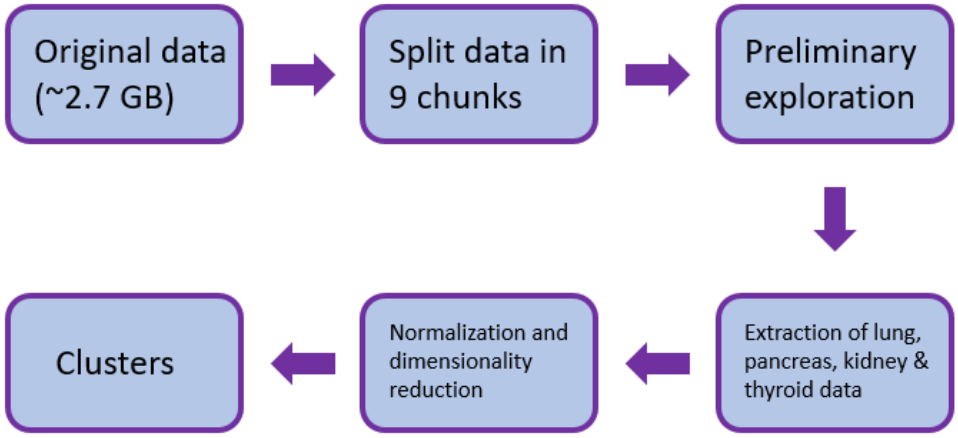
Due to big size of expression data (~2.7 GB) I split the data into 9 smaller chunks of the data that afterwards would be subsetted based on cancer type.

Next, for the exploratory data and subsetting the data by the cancer types I used following expression for extracting corresponding cancers:
Lung cancer <- c('lung', 'Lung', 'Bronchioloalveolar', 'Bronchogenic'), Thyroid cancer <- c('thyroid', 'Thyroid'), Pancreatic cancer <- c('Pancreatic'), Kidney cancer <- c('renal', 'Renal'). Due to design of the further analysis and characteristics of comparison of future results (namely comparing the data based on source i.e. COSMIC, CCLE and NCI60) Combat Expression Data in the dataset named "Combat_Exp_Value" was not taken into further account. After extraction of the data based on cancer types and then subsetting that by the data source (COSMIC, CCLE and NCI60) took place. Despite the fact that in the data there was no obvious technical data, I applied sva::sva() function to check for unknown technical variables, however with no success.

Such data was then fed into a pipeline and expression values were normalized using

norm() and data underwent dimensionality reduction procedure using svd() function.

As the last step of the pipeline I applied clustering methods described in the 'Clustering methods.' Subparagraph using following functions (in the same order as described previously): stats::kmeans(), meanShiftR::meanShift(), dbscan::dbscan(), mclust::Mclust().



Pic.1 Scheme of Project pipeline.

Results

The final results were stored in the summary tables where label informing about original cancer type (taken from original Cellosaurus information) was compared against following variables/columns: raw.kmeans, norm.kmeans, svd.kmeans, norm.meanshift, svd.meanshift, raw.dbscan, norm.dbscan, svd.dbscan, svd.mclust.

As a naming convention I assumed that prefixes raw, norm and svd correspond to the values obtained from as follows: raw data, normalized data and data that underwent svd::svd() function. Per analogy suffixes kmeans, meanshift, dbscan, mclust correspond to the values obtained from following functions: stats::kmeans(), meanShiftR::meanShift(), dbscan::dbscan(), mclust::Mclust().

As it is easy to notice number of columns in the final output is not a result of simple Cartesian multiplication ({raw, norm, svd} x {kmeans, meanshift, dbscan, mclust}) which would sump up to 12. There is 9 columns. The reason for that is that not all of the functions used for clustering can be fed with either very big raw or norm data.

Result tables had varying number of observations within one cancer type. This situation is caused because of different platforms (i.e. COSMIC, CCLE and NCI60) had varying number of cell lines used for analyses.

Through measuring number of unique cluster group that each clustering method produced I conclude that a few methods did not produce potentially interesting results. Namely meanShift() function applied to normalized data produced as many or nearly as many clusters as there was observation in given dataset. At the same time this very function produced only one cluster for data that underwent svd() function.

kmeans() function through all cancers and all datatypes generated number of unique clusters is same or similar to the number of unique original labels. dbscan() function on the other hand generated consistently through all the datasets 2 clusters.

```
> table(lung.CCLE$Diseases_extracted, lung.CCLE$norm.kmeans)

                                 1  2  3  4  5  6  7  8  9 10 11 12
  Adenosquamous lung carcinoma   1  0  3  0  0  0  0  0  0  0  1  0
  Bronchioloalveolar carcinoma   0  0  1  0  0  0  4  0  0  0  1  0
  Bronchogenic carcinoma         0  0  0  0  0  0  1  0  0  0  0  0
  Large cell lung carcinoma      1  0  1  0  1  0  1  2  0  0  6  0
  Lung adenocarcinoma            1  0  7  1  0  9 14  0  0  0 22  0
  Lung carcinoid tumor           0  0  1  0  0  0  0  0  0  0  0  0
  Lung giant cell carcinoma      0  0  0  0  0  0  0  0  0  0  2  0
  Lung mucoepidermoid carcinoma  0  0  0  0  0  0  0  0  0  0  0  1
  Non-small cell lung carcinoma  0  2  1  0  0  0  5  0  0  0  3  1
  Papillary lung adenocarcinoma  0  0  0  0  0  1  0  0  0  0  0  0
  Small cell lung carcinoma      0  1  0  0  3  0  0 24  7  4 10  0
  Squamous cell lung carcinoma   1  0  0  0  0  1  5  0  0  0 12  1
table(lung.CCLE$Diseases_extracted, lung.CCLE$raw.kmeans)

                                 1  2  3  4  5  6  7  8  9 10 11 12
  Adenosquamous lung carcinoma   0  4  0  0  0  1  0  0  0  0  0  0
  Bronchioloalveolar carcinoma   1  3  0  0  0  2  0  0  0  0  0  0
  Bronchogenic carcinoma         0  0  0  0  0  1  0  0  0  0  0  0
  Large cell lung carcinoma      5  1  1  0  1  1  1  0  1  0  0  1
  Lung adenocarcinoma            7 16  0  0  0 11 10  0  0  0  1  9
  Lung carcinoid tumor           0  0  0  0  0  0  0  0  0  0  1  0
  Lung giant cell carcinoma      1  1  0  0  0  0  0  0  0  0  0  0
  Lung mucoepidermoid carcinoma  0  1  0  0  0  0  0  0  0  0  0  0
  Non-small cell lung carcinoma  1  4  0  1  1  2  2  0  0  0  0  1
  Papillary lung adenocarcinoma  0  0  0  0  0  0  0  0  0  0  0  1
  Small cell lung carcinoma      5  0 10  0  1  0  1 21  7  2  2  0
  Squamous cell lung carcinoma   4 10  0  0  1  2  1  0  0  0  1  1
> table(lung.CCLE$Diseases_extracted, lung.CCLE$svd.kmeans)

                                 1  2  3  4  5  6  7  8  9 10 11 12
  Adenosquamous lung carcinoma   0  0  0  4  0  0  1  0  0  0  0  0
  Bronchioloalveolar carcinoma   0  0  0  2  2  0  0  0  0  2  0  0
  Bronchogenic carcinoma         0  0  0  0  0  0  0  0  0  1  0  0
  Large cell lung carcinoma      0  0  0  8  1  0  0  0  0  3  0  0
  Lung adenocarcinoma            0  4  0 24  3  1  3  1  2 14  2  0
  Lung carcinoid tumor           0  0  0  1  0  0  0  0  0  0  0  0
  Lung giant cell carcinoma      0  0  0  0  0  0  1  0  0  0  1  0
  Lung mucoepidermoid carcinoma  0  0  0  1  0  0  0  0  0  0  0  0
  Non-small cell lung carcinoma  0  0  0  5  2  0  0  0  0  5  0  0
  Papillary lung adenocarcinoma  0  0  0  1  0  0  0  0  0  0  0  0
  Small cell lung carcinoma      3  7  2 12  3  0  1  0  1  9 10  1
  Squamous cell lung carcinoma   0  1  0  9  4  1  1  0  0  4  0  0
```

```
> table(thyroid.COSMIC$Diseases_extracted, thyroid.COSMIC$raw.kmeans)

                                                  1 2 3 4 5
  Hereditary thyroid gland medullary carcinoma//NCIt 1 0 0 0 0
  Thyroid gland follicular carcinoma                3 0 1 1 1
  Thyroid gland papillary carcinoma                 0 1 1 0 0
  Thyroid gland sarcoma                             0 0 1 0 0
  Thyroid gland undifferentiated (anaplastic) carcinoma 0 1 2 0 3
> table(thyroid.COSMIC$Diseases_extracted, thyroid.COSMIC$norm.kmeans)

                                                  1 2 3 4 5
  Hereditary thyroid gland medullary carcinoma//NCIt 0 0 1 0 0
  Thyroid gland follicular carcinoma                3 1 2 0 0
  Thyroid gland papillary carcinoma                 1 0 0 1 0
  Thyroid gland sarcoma                             1 0 0 0 0
  Thyroid gland undifferentiated (anaplastic) carcinoma 0 0 1 4 1
> table(thyroid.COSMIC$Diseases_extracted, thyroid.COSMIC$svd.kmeans)

                                                  1 2 3 4 5
  Hereditary thyroid gland medullary carcinoma//NCIt 0 0 0 0 1
  Thyroid gland follicular carcinoma                2 3 1 0 0
  Thyroid gland papillary carcinoma                 0 1 0 1 0
  Thyroid gland sarcoma                             1 0 0 0 0
  Thyroid gland undifferentiated (anaplastic) carcinoma 0 2 0 4 0
```

Pic.2 Picture shows frequencies of clusters generated by kmeans against original disease labels (table to the right: cancer type – lung, source type – CCLE; table to the left: cancer type – thyroid, source type – COSMIC).

```
> table(lung.COSMIC$Diseases_extracted, lung.COSMIC$raw.dbscan)

                                 0  1
  Adenosquamous lung carcinoma   0  4
  Bronchioloalveolar carcinoma   0  6
  Bronchogenic carcinoma         0  1
  Large cell lung carcinoma      1  9
  Lung adenocarcinoma            0 51
  Lung carcinoid tumor           0  4
  Lung carcinoma                 0  1
  Lung giant cell carcinoma      0  2
  Lung mucoepidermoid carcinoma  0  2
  Non-small cell lung carcinoma  0 17
  Papillary lung adenocarcinoma  0  1
  Small cell lung carcinoma      0 59
  Squamous cell lung carcinoma   0 19
> table(lung.COSMIC$Diseases_extracted, lung.COSMIC$norm.dbscan)

                                 0  1
  Adenosquamous lung carcinoma   0  4
  Bronchioloalveolar carcinoma   0  6
  Bronchogenic carcinoma         0  1
  Large cell lung carcinoma      1  9
  Lung adenocarcinoma            0 51
  Lung carcinoid tumor           0  4
  Lung carcinoma                 0  1
  Lung giant cell carcinoma      0  2
  Lung mucoepidermoid carcinoma  0  2
  Non-small cell lung carcinoma  0 17
  Papillary lung adenocarcinoma  0  1
  Small cell lung carcinoma      0 59
  Squamous cell lung carcinoma   0 19
> table(lung.COSMIC$Diseases_extracted, lung.COSMIC$svd.dbscan)

                                 0  1
  Adenosquamous lung carcinoma   0  4
  Bronchioloalveolar carcinoma   0  6
  Bronchogenic carcinoma         0  1
  Large cell lung carcinoma      1  9
  Lung adenocarcinoma            1 50
  Lung carcinoid tumor           1  3
  Lung carcinoma                 0  1
  Lung giant cell carcinoma      0  2
  Lung mucoepidermoid carcinoma  0  2
  Non-small cell lung carcinoma  0 17
  Papillary lung adenocarcinoma  0  1
  Small cell lung carcinoma      1 58
  Squamous cell lung carcinoma   0 19
```

```
> table(pancreas.COSMIC$Diseases_extracted, pancreas.COSMIC$raw.dbscan)

                                 0  1
  Pancreatic adenocarcinoma      0 13
  Pancreatic adenosquamous carcinoma 0  1
  Pancreatic carcinoma           0  3
  Pancreatic ductal adenocarcinoma 0 13
  Pancreatic somatostatinoma     1  0
> table(pancreas.COSMIC$Diseases_extracted, pancreas.COSMIC$norm.dbscan)

                                 0  1
  Pancreatic adenocarcinoma      0 13
  Pancreatic adenosquamous carcinoma 0  1
  Pancreatic carcinoma           0  3
  Pancreatic ductal adenocarcinoma 1 12
  Pancreatic somatostatinoma     1  0
> table(pancreas.COSMIC$Diseases_extracted, pancreas.COSMIC$svd.dbscan)

                                 0  1
  Pancreatic adenocarcinoma      3 10
  Pancreatic adenosquamous carcinoma 0  1
  Pancreatic carcinoma           2  1
  Pancreatic ductal adenocarcinoma 3 10
  Pancreatic somatostatinoma     1  0
```

Pic.3 Picture shows frequencies of clusters generated by dbscan against original disease labels (table to the right: cancer type – lung, source type – COSMIC; table to the left: cancer type – pancreas, source type – COSMIC).

Discussion

Pic.2 and Pic.3 are representative to other functions that were applied to the data during analysis. Although I cannot firmly state that functions produced results that could be corresponding to the original labels, it is visible that kmeans() function produced clusters that are skewed towards most numerous cancer subtypes (in case of lung it would be: lung adenocarcinoma, small cell lung carcinoma and squamous cell lung carcinoma). In case of dbscan(), the function produced clustering that are very skewed towards one of the clustering value in practice not giving any reasonable signal from the data. Similar situation is in case of Mclust() function which either generates one cluster with insignificant signals from the second cluster or generates a few clusters which in number are close to the original number of labels, however centered around most numerous cancer types.

Based on the results from efficacy of finding significant clusters checked against original labels informing about cancer subtypes of a particular sample, it is inevitable to state that efficient tool to find such clustering was not found.

Following idea to find significant clusters is to apply one of Deep Learning techniques, namely Variational Autoencoder and generate simulated values for corresponding samples. As the second step would be to try and cluster the simulated data and check if it produces any reasonable outcome.

References

1. Jeong, I., Yu, N., Jang, I.et al., ‘GEMiCCL: mining genotype and expression data of cancer cell lines withelaborate visualization.Database’ Vol. 2018: article ID bay041;
2. Jacob, L., Gagnon-Bartsch, J., Speed, P. T, ‘Correcting gene expression data when neither the unwanted variation nor the factor of interest are observed.’,Biostatistics 2016 Jan; 17(1): pp. 16–28;
3. Wall, M. E., Rechtsteiner, A., Rocha L. M., ‘Singular value decomposition and principal component analysis In A Practical Approach to Microarray Data Analysis (D.P. Berrar, W. Dubitzky, M. Granzow, eds.)’ Kluwer: Norwell, MA, 2003. pp. 91-109.
4. Eraslan G., Avsec Ž., Gagneur J., Theis F. J., ‘Deep learning: new computational modelling techniques for genomics’, Nature Reviews Genetics, 2019, Vol. 20, pp. 389–403