Naive Bayes AI for Lung Nodule Cancer Diagnosis

Naive Bayes AI for Lung Nodule Cancer Diagnosis

Noah B. Black, Mathew D. Alexander

Midland, Michigan

Facilitator: Marc Black

12/31/20

Acknowledgement of Contribution:

This work would not be possible without the amazing support from miRcore and its summer camps and Dr. Bradley Sachs. miRcore is an organization affiliated with the University of Michigan that teaches students about bioinformatics, computational biology, and genomics. Without access to their camps over the summer of 2020, this work could not have been done. The camps covered tools and procedures integral to this paper. Additionally, a huge thank you to Dr. Sachs, who, as an oncologist, assisted us in navigating cancer and the diagnostic tools already available.

Table of Contents:

Introduction …………………………………………………………………………… 3

Materials and Methods ……………………………………………………………….. 4-6

Results ………………………………………………………………………………… 6-7

Discussion and Conclusion ………………………………………………………........ 7-8

References/Literature Cited …………………………………………………………….. 9-10

Introduction:

Lung nodules are ovular masses that propagate within the lungs of many patients. 40% of detected nodules are malignant. Detection is typically not deliberate, instead they are normally discovered during chest x-rays or CT scans administered for other reasons.

Lung nodules are no bigger than 3cm and can be caused by a number of factors. Benign nodules are typically caused by inflammation. Inflammation may be caused by infections, which causes the formation of a granuloma. Granulomas are patches of cells that form when tissue is inflamed, and over time the cells may become calcified. Noninfectious nodules also occur and are associated with many disorders including sarcoidosis and rheumatoid arthritis. A third cause for nodules can be either malignant or benign: nodules caused by neoplasms, or abnormal growth of tissues. Many malignant tumor types are associated with nodules including lung cancer, lymphoma, carcinoid, sarcoma, and metastatic tumors.

Understanding cancer is key to understanding the larger context of this research. Cancer is simply a group of diseases characterized by rapid and uncontrolled cell division. Tumors form when cancer proliferates in one area. Lung cancer can be subdivided into two main groups: Non-small Cell Lung Cancer (NSCLC) and Small Cell Lung Cancer (SCLC.) SCLC diagnoses make up a small portion of lung cancer cases, and while significant, it does not concern this research. NSCLC is a name given to most other lung cancers; it makes up the vast majority of lung cancer diagnoses. The cancers and tumors outlined in the previous paragraph are in neither of these main groups and are regarded as entirely different forms of lung cancer. Cancers are treated differently based on their categorization.

Determining the cancer status of a lung nodule can take months to over a year. Typically, CT scans will be administered to the patient over a duration of many months. Depending on the growth rate and other factors, a biopsy may be prescribed. This isn’t just exceedingly costly for the patient, but also threatens their mortality. Malignant lung nodules double in size every 25 days to 15 months. Considering the mortality rate for tumors smaller than 1cm is 30% higher than the average lung nodule cancer diagnoses of 50%, it’s critical that cancer is detected early. Our research aims to accomplish this through alternative methods.

 The ideal pre-biopsy procedure would be cheaper, faster, and less stressful for patients. Using a publicly available dataset information of Gene Expression Omnibus, we set out to accomplish these goals by using artificial intelligence (AI.)

Our dataset, GSE 135304, outlined the protein expression level of peripheral blood of patients with malignant lung nodules, benign lung nodules, and no lung nodules. To collect the data, we used R to download the raw expression data of this dataset. This will be further outlined in the Materials and Methods section. We used the benign group as the control or independent sample and the malignant group as the comparing or dependent sample.

Data collected for GSE 135304 was collected using microarrays. Utilization will be discussed further in the paper, however its function is integral to the theory behind this paper's research. Microarrays are plates with thousands of ssDNA oligonucleotides attached for many sequences. Our microarrays will analyze the intensity of RNA in the peripheral blood of patients. This is accomplished by extracting peripheral blood from patients and isolating the RNA through a system of chemical and physical procedures. The RNA will be washed over the array, where it will hybridize to the corresponding oligonucleotides. The number of hybridizations per sequence adverts the prevalence of the sequence in the blood sample.

Materials and Methods:

Our hypothesis was that an artificial intelligence can be developed that is a “good” predictor of lung nodule malignancy using peripheral blood RNA expression level. A “good” diagnostic device has an efficacy of over 70%. Our independent variable was the expression level of benign lung nodule patients. Our dependent variable was the expression level of malignant lung nodule patients.

A critical part of any research project is the data collection process. We used Gene Expression Omnibus for this purpose. Gene Expression Omnibus (GEO) is a public archive maintained by the National Institute of Health that makes available genomic data to the public. One of those types of data is microarray data, which we utilized in this project.

GEO acts as a search engine to find genomic expression data. A dataset that complimented the intent of our research was GSE 135304. “Non-Small Cell Lung Cancer, Peripheral Immunity, Extending the Tumor Microenvironment” is a dataset that documents the peripheral blood RNA expression level of patients with malignant lung nodules, benign lung nodules, and no lung nodules. This data can be downloaded and analyzed using the programming language R or the integrated tool: GEO2R.

R is a programming language designed for statistical computing. It is commonly used throughout the field of science and especially biology. R has a variety of built-in features that make it very suitable for computational analysis of big data. R also has many third-party packages, or external modules, that extends its use to more specific tasks. As mentioned previously, GEO has a built-in tool for genomic analysis called GEO2R. GEO2R is run by R with specially designed packages for data extraction and analysis. These packages will be explained later on, but were relied upon for our R program.

Another programming language utilized for this project was Python. Python is a high-level programming language extremely common in scientific computing. Python's large library of external modules makes it a key programming language for data analysis and machine learning. These modules will be iterated on later in the paper, but were extremely useful for preparing data and artificial intelligence deployment.

Our first step was to utilize GEO2R on GEO. GEO2R allows users the capability of comparing genomic expression data of one group to another. Our control samples were the 220 benign patients included in our dataset. We compared the benign sample to the 404 malignant patients in the dataset. This yielded a ranking of the top genes most differently regulated between the two groups. This is organized by ranking of lowest p value. P value is a statistical measurement of how likely something is to occur by chance, in this case the genes being differently regulated. The lower a p value is the higher the chance that the gene is differently regulated between the two groups.

Our next step of data analysis, after finding GSE 135304, was to download the raw expression data. This was done in R using the Biobase, GEOquery, and limma libraries. The data collected on microarrays is measured in fluorescent units, and need to be normalized. Data normalization allows for more accurate comparison between different genes. To normalize the data a log2 function is applied to all of the expression measurements. This data frame was saved into an external csv file that could be accessed by our python program.

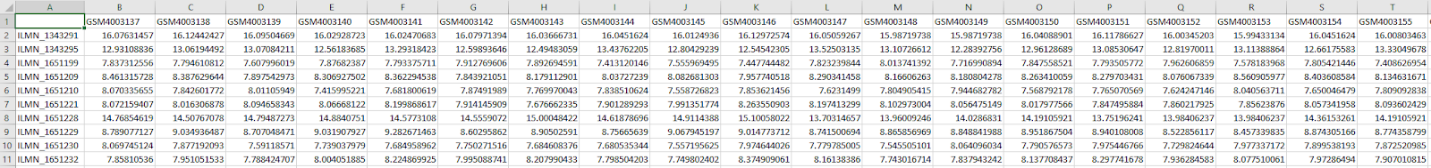


Fig 1 - Part of the expression data frame, samples listed as columns and oligo sequences listed as rows

The third step is creating another dataset with all of the categorization data. The first data frame lists the samples and its corresponding expression data. This new data frame, once again using the aforementioned R libraries, maps the samples with its cancer status and other variables. Once this data frame was extracted, it was saved into a csv file to be accessed by our python program at a later time.

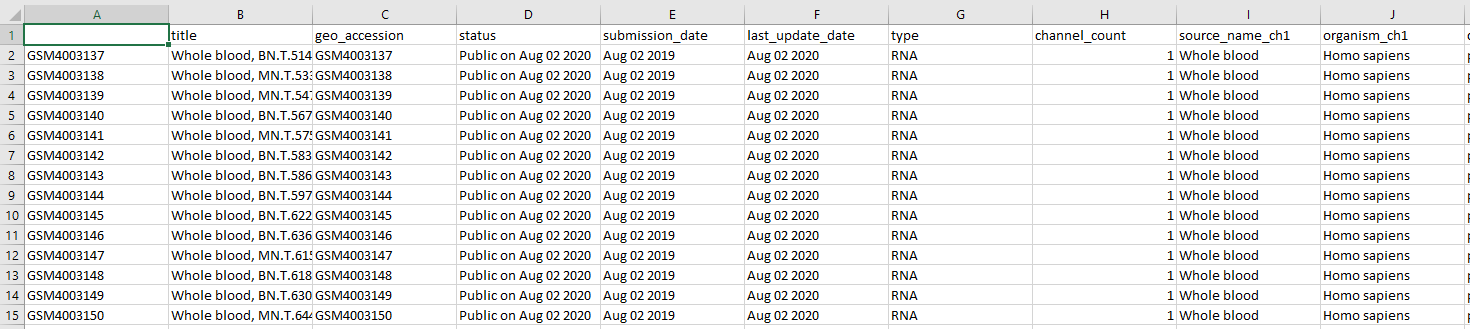


Fig 2 - Part of the categorization data frame, samples listed as rows and descriptors listed as columns

After R was utilized, further data analysis and artificial intelligence was developed using python. Our data manipulation was through a library called pandas, a popular library for data frame manipulation. Our GEO2R dataset had gaps that needed to be corrected. Values that were not existent or not numerical were removed. The dataset, already numerically ordered by smallest to highest p value, was reduced to the top 50 genes according to smallest p value.

Once the top 50 genes were isolated, we loaded in our expression data csv and reduced that to only include the top 50 genes per individual. Afterwards, we loaded in the descriptors csv.  This information was used to create a binary list of all of the nodule patients in the study. Malignant patients were labeled one and benign were labeled zero.

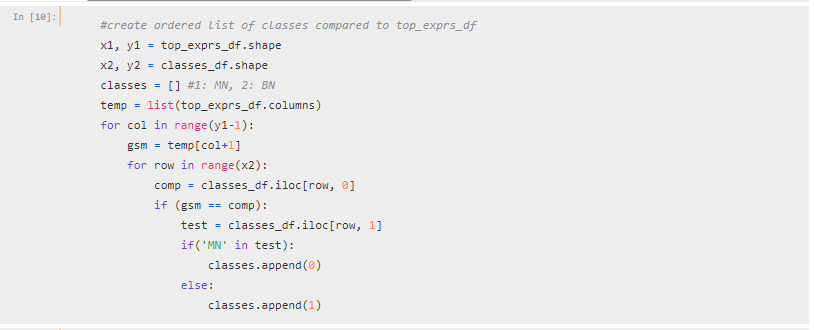


Fig 3 - Python code used to make a binary list of patients in the study

The final step before the artificial intelligence could be developed was to create a final data frame that was formatted for the task. First, we transposed the expression data csv so the oligo sequences were the columns and the patients were the rows. Then we created a new column which inherited the binary cancer status of each patient.

Finally, we created the artificial intelligence. There are many different kinds of artificial intelligence that calculate an outcome differently by using different equations. We decided to implement a Naive Bayes artificial intelligence. We created the AI using an external library called Scikit Learn, which is designed for artificial intelligence development.

The first step in AI development was to determine a training split. A training split determines the proportion of data points used to train the AI vs the proportion of data points used to verify the accuracy of the AI. We decided on the relatively standard split of 70/30, which means 70% of the data points are used to train the program and 30% of the data points are used to measure the accuracy of the artificial intelligence. We also had to determine a random state or seed that the program used to determine how to split up the data. The random state ensures the experiment is replicable. After testing we found the best random state was 5155. The artificial intelligence was then created by fitting a Gaussian Naive Bayes model to the data.

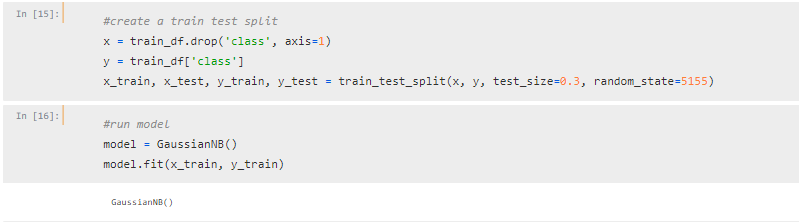


Fig 4 - Python code used to fit the Gaussian Naive Bayes to the final data frame

Results (Data and Findings):

Diagnostic tools with a 70-80% accuracy are characterized as “good,” and 80-90% are “very good.” Our artificial intelligence has an accuracy of about 80.3%, which categorizes this as a “good” to “very good” diagnostic tool. We believe, based on the tool's efficacy, this tool can be utilized by doctors to expedite the time from nodule recognition to treatment for many patients.

Our testing data set included 188 samples. Of those samples we had 25 false positives (zero represented as one) and 12 false negatives (one represented by zero.) That’s 13.3% and 6.4% of all results respectively. This makes false positives far more likely than false negatives in this particular set of testing data. This will be further discussed in the discussion section and could have certain research implications.

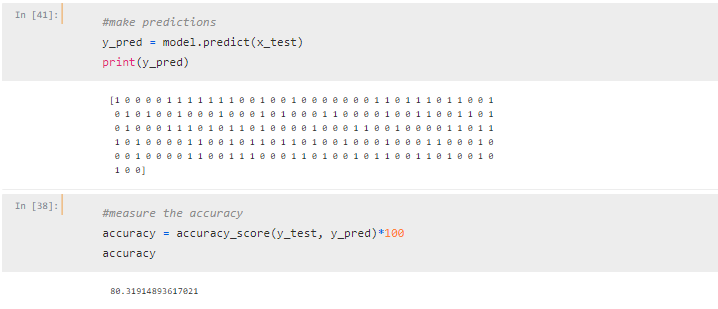
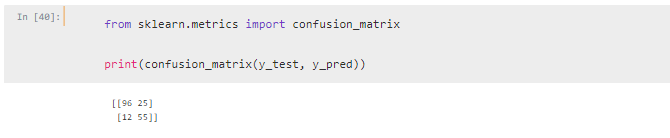


Fig 5 - Python code for the prediction of the testing a data and the accuracy associated with the result

Fig 6 - shows the confusion matrix that discloses (from right to left) the number of true negative, false positive, false negative, and true negative results

Discussion and Conclusion:

The top 50 genes were determined based on p value. P value is a statistical measurement that determines the percent of chance of an outcome. This is as opposed to difference in LogFC value, or the normalized absolute expression value of the gene. In the context of our data the p value determines whether the difference of gene regulation between the benign and malignant groups is determined by random chance. The smaller a p value is the lower the chance of randomness. This is an important statistic and methodology to mention because the genes used for artificial intelligence development were not chosen based on their actual absolute numerical differences in regulation. This could have an impact on artificial intelligence prediction results because, while p value is a very reliable statistical method, the values represented on the microarray could be extremely similar.

The tool used to gather protein expression levels of patients is a microarray. Microarrays, as previously mentioned, are plates with thousands of attached single stranded oligonucleotide sequences. The RNA (single stranded) is isolated from the peripheral blood of patients and washed over the microarray. The RNA and ssDNA will hybridize and release a certain illuminosity that can be measured with laboratory equipment. This technology is already in use for cancer diagnostics and is a prevalent technology in genomics.

The cost to run this technology is dependent on the number of genes being analyzed. Because our array only targets 50 RNA sequences out of many thousands, an array designed to complement our artificial intelligence is a targeted array. Targeted arrays, per sample, are between 10-100 dollars. And though dependent on the lab, it is a fair estimate that patients can expect to receive their results within a few weeks of submitting to a test.

There are many different varieties of artificial intelligence. Different types can be deployed for different kinds of analysis and prediction depending on the application. We decided to implement a Gaussian Naive Bayes artificial intelligence model for this data. The Naive Bayes model is unique because it operates off of conditional probabilities and therefore interprets each factor as an individual predictor. This means the artificial intelligence will assume mutual exclusivity for all input categories. This is not true for protein regulation, as the web of protein interactions is populated by thresholds and feedback loops. However, we decided Naive Bayes would be the best predictor for our purposes because the complexity of genome regulation is not embedded into the logic of other artificial intelligence models. We concluded that operating under mutual exclusivity, though not true, would be the best way to avoid false or indirect correlations that the program might find.

As stated in the previous section we had a total of 37 wrong predictions out of a possible 188. These results break down to 25 false positives and 12 false negatives. These results could be due to many reasons. In the testing data there were 67 benign samples and 121 malignant samples. If errors are evenly distributed that makes the chances for false positives far more likely than false negatives - about 1.81 times more likely. However, in actuality the rate of false positives was about 2.08 times higher.

We think there could be a few probable reasons for these skewed results. In absolute terms, the volume of malignant to benign samples made false positives much more likely. The test, proportionally, did skew more negative (malignant) than positive (benign.) This could be because of lack of training data, the more data points available to train the AI makes it more accurate. The training data also skewed negative, which means the artificial intelligence was exposed to more malignant data points than benign. This could make it more sensitive to malignant data, which would explain the negative results in the testing data.

Overall, the results of this experiment suggest the artificial development and deployment was effective. With an efficacy of 80.3%, the program's diagnostic viability is very high as it is a “good” to “very good” diagnostic tool. Considering this test can be administered and returned in just a few weeks, the benefits far outweigh the cost. Most importantly, it could save patient lives by expediting the typical time lung nodules are recognized to being diagnosed as malignant. There are also other benefits such as lower cost and mental health. Repeated CT scans over months to a year can be expensive for many, and considering just one microarray test could avoid this entire process, many patients would benefit economically from this test. Additionally, waiting months on end and constant presence in medical institutions just to diagnose a nodule may cause distress in patients.

References/Literature Cited

1. National Cancer Institute. “What is Cancer?” National Cancer Institute, 2015, <https://www.cancer.gov/about-cancer/understanding/what-is-cancer>.
2. National Human Genome Research Institute. “DNA Microarray Technology Fact Sheet.” National Human Genome Research Institute, 2020, <https://www.genome.gov/about-genomics/fact-sheets/DNA-Microarray-Technology>
3. Fedak, Vladimir. “Top 10 Most Popular AI models.” DZone, 2018, <https://dzone.com/articles/top-10-most-popular-ai-models>
4. K, Dhiraj. “Naive Bayes Classifier in Python Using Scikit-learn.” Heartbeat, 2020, <https://heartbeat.fritz.ai/naive-bayes-classifier-in-python-using-scikit-learn-13c4deb83bcf>
5. The American Cancer Society Medical and Editorial Content Team. “What is Lung Cancer?” The American Cancer Society, 2019, <https://www.cancer.org/cancer/lung-cancer/about/what-is.html>
6. Mayo Clinic Staff. “Lung Cancer.” Mayo Clinic, 2020, <https://www.mayoclinic.org/diseases-conditions/lung-cancer/symptoms-causes/syc-20374620>
7. Smith, Gordon, et al. “limma: Linear Models for Microarray and RNA-Seq Data User’s Guide.” Bioconductor, 2020, <https://www.bioconductor.org/packages/devel/bioc/vignettes/limma/inst/doc/usersguide.pdf>
8. Davis, Sean. “Using the GEOquery Package.” Bioconductor, 2014, <https://www.bioconductor.org/packages/release/bioc/vignettes/GEOquery/inst/doc/GEOquery.html>
9. Falcon, Seth, et al. “An Introduction to Bioconductor’s *ExpressionSet* Class.”, Bioconductor, 2007, <https://bioconductor.org/packages/release/bioc/vignettes/Biobase/inst/doc/ExpressionSetIntroduction.pdf>
10. Šimundić, Ana-Maria. “Measures of Diagnostic Accuracy: Basic Definitions.” National Institute of Health, 2009, <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4975285/>
11. Cleveland Clinic. “Pulmonary Nodules.” Cleveland Clinic, 2016, <https://my.clevelandclinic.org/health/diseases/14799-pulmonary-nodules>
12. The American Cancer Society Medical and Editorial Content Team. “Lung Nodules.” The American Cancer Society, No Date, <https://www.cancer.org/cancer/lung-cancer/detection-diagnosis-staging/lung-nodules.html>
13. University of Rochester Medical Center. “Pulmonary Nodules.” University of Rochester, No Date, <https://www.urmc.rochester.edu/encyclopedia/content.aspx?contenttypeid=22&contentid=pulmonarynodules>
14. Radiologyinfo.org. “Needle Biopsy of the Lung.” Radiologyinfo.org, 2020, <https://www.radiologyinfo.org/en/info.cfm?pg=nlungbiop>
15. Sharma, Nikita. “Understanding the Mathematics Behind Naive Bayes.” HeartBeat, 2020, <https://heartbeat.fritz.ai/understanding-the-mathematics-behind-naive-bayes-ab6ee85f50d0>
16. Hoopes, Laura. “Genetic Diagnosis: DNA Microarrays and Cancer.” Nature, 2008, <https://www.nature.com/scitable/topicpage/genetic-diagnosis-dna-microarrays-and-cancer-1017/>
17. Biocompare. “DNA Microarrays: A Trusted Tool Keeps Evolving.” Biocompare, 2015, <https://www.biocompare.com/Editorial-Articles/172195-DNA-Microarrays-A-Trusted-Tool-Keeps-Evolving/#:~:text=A%20targeted%20DNA%20microarray%20runs,add%20both%20cost%20and%20time>
18. Nationwide Children’s. “Microarray Test Analysis.” Nationwide Children’s, 2020, <https://www.nationwidechildrens.org/family-resources-education/health-wellness-and-safety-resources/helping-hands/microarray-test-analysis#:~:text=This%20test%20compares%20the%20patient's,to%20get%20the%20test%20results>
19. American Health Imaging. “How Much Does a CT Scan Cost?” American Health Imaging, No Date, <https://www.americanhealthimaging.com/blog/how-much-does-a-ct-scan-cost/#:~:text=In%20general%2C%20you%20can%20expect,or%20bill%20your%20insurance%20provider>
20. Gene Expression Omnibus. “GEO Overview.” National Institute of Health, 2020 <https://www.ncbi.nlm.nih.gov/geo/info/overview.html>
21. The R Foundation. “What is R?” The R Foundation, No Date, <https://www.r-project.org/about.html>
22. Python Software Foundation. “What is Python? Executive Summary.” Python Software Foundation, No Date, <https://www.python.org/doc/essays/blurb/>
23. Illumina, Inc. “Gene Expression Microarray Data Quality Control.” Illumina, Inc., 2010, <https://www.illumina.com/Documents/products/technotes/technote_gene_expression_data_quality_control.pdf>
24. Ferreira, Hugo. “Confusion Matrix and Other Metrics in Machine Learning.” Medium, 2018, <https://medium.com/hugo-ferreiras-blog/confusion-matrix-and-other-metrics-in-machine-learning-894688cb1c0a>
25. Ruoslahti, Erkki. “How Cancer Spreads.” JSTOR, 1996, <https://www.jstor.org/stable/24993350?casa_token=0J3ITnU3EGcAAAAA%3ABNNZHITRqey5OUoTqYy_BRYCVE4fivzvkVmaOc4yIa9hMEC5u9omJbI8rCELYQ-CIxTFm7wYpGirpihhVCme2xMTumuRlQNjJ_1aC19LYMxp66f9NV3J&seq=1>