SWIB24: Leukemia Data Analysis

Thanks to Professor David Harrington and Julie Vu (Harvard) for material.

Reading on graphical summaries

– *OI Biostat*: Section 1.6 and earlier material on numerical and graphical summaries.

Golub Leukemia Data. Gene expression data for 7,129 genes were collected from 72 children with acute leukemia, of which 47 had acute lymphoblastic leukemia (ALL) and 25 had acute myeloblastic leukemia (AML). The goal of the experiment was to identify genes that are differentially expressed between ALL versus AML, in order to develop a strategy for diagnosing leukemia type based on gene expression data.

Can childhood leukemia be diagnosed using molecular measurements and software?

In other words, are there two sets of genes in a leukemia sample: one that is highly expressed in ALL patients but not in AML, and the other highly expressed in AML patients but not ALL?

Some questions to think about:

1. *Differential genes* Can we identify a set of individual genes that are expressed in ALL cancers when compared to AML cancers?
2. *Classification* Can we distinguish AML from ALL using a subset of genes by building a prediction algorithm? Some algorithms to try include Lasso, Elastic Net, the SuperLearner library.
3. *Visualization* Can we visualize the results using methods such as a heatmap?
4. *Interpretation* How do your findings compare to those published in the paper by Golub, T et al (1999).

Variables in the dataset:

– `Samples`: Sample or chip number. The material from each patient was examined on a separate chip and experimental run.
– `BM.PB`: Type of patient material analyzed. BM denotes bone marrow; PB denotes a peripheral blood sample.
– `Gender`: `F` for female, `M` for male.
– `Source`: Hospital where the patient was treated.
– `tissue.mf`: A variable showing the combination of type of patient material and sex of the patient. For example, `BM:f` indicates a bone marrow sample from a female patient.

– `cancer`: The type of leukemia, with a notation for subtype within ALL. `aml` is AML, `allB` is ALL which started in B-cells (cells that mature into plasma cells), and `allT` is ALL with T-cell origin (T-cells are a type of white blood cell).

Some code to help you get started

1. After loading the Golub data, execute the following code to create a matrix called `gene.matrix` that only contains gene expression values:

```
## load data -- please change below to specify the directory where you data is stored
load(here::here("data/golub_exprs_pheno.Rdata"))
Golub <- golub.exprs.pheno

## Check the dimensions of the Golub matrix
dim(Golub)
```

```
## [1]   72 7135
```

```
## View the frcolumn names of the Golub matrix
colnames(Golub)[1:10]
```

```
##  [1] "Samples"        "BM.PB"          "Gender"         "Source"
##  [5] "tissue.mf"      "cancer"         "AFFX-BioB-5_at" "AFFX-BioB-M_at"
##  [9] "AFFX-BioB-3_at" "AFFX-BioC-5_at"
```

```
#create gene.matrix, trimmed version of Golub dataset
gene.matrix = as.matrix(Golub[,-(1:6)])
```

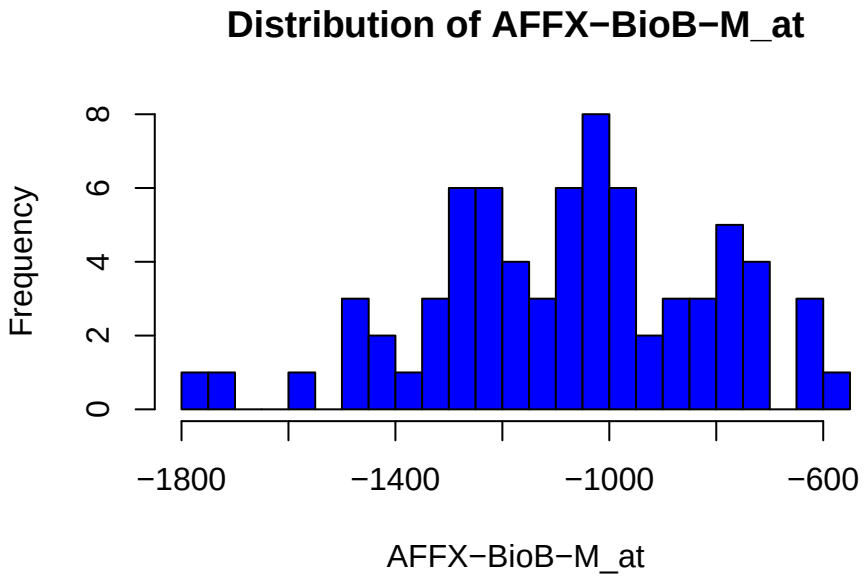
Matrix notation is used to specify rows and columns. For a dataframe `A`, `A[i, j]` refers to the element in row `i` and column `j`. `A[i,]` refers to row `i` and `A[,j]` refers to column `j`.

How does `gene.matrix` differ from the original dataset?

While the original dataset contains phenotype information in the first six columns, `gene.matrix` only consists of the gene expression data from the last 7,129 columns of `Golub`.

Plot a histogram showing the distribution of the expression levels of the second gene across patients. Describe the distribution.

```
hist(gene.matrix[,2], breaks=40, xlab=colnames(gene.matrix)[2], col="blue", main=paste("Distributio
```



Create a logical variable, `leuk.type`, that has value `1` for AML and value `0` for anything that is not AML (i.e., `allT` and `allB`).

```
#create logical variable
leuk.type = (Golub$cancer)

#view summary of leukemia types
table(leuk.type)
```

```
## leuk.type
## allB allT  aml
##   38    9   25
```

How many patients are there with AML? How many with ALL?

Some ideas for developing a project

These are some ideas. Feel free to come up with your own

Comparing classifiers

- Try different classification algorithms and compare AUC.

Comparing methods to control FDR

- Compare methods to control the false discovery rate (FDR): understand why in the presence of multiple tests, the FDR increases. Two methods to control FDR are (1) Storey's q value approach (2) Simulation (see code). Suggested steps:
- Filter the 7100 genes and select the top 20% of most variable genes.
- Randomly select 10 AML and 18 ALL to leave out as a test set.
- In the remaining 15 AML and 29 ALL, use a t test to test difference between ALL and AML
- Control the FDR using each of the two approaches and select the genes that meet FDR < 0.05
- Train a classifier using the selected genes using each of the FDR approach above.
- Test the classifier using the test set defined above.