STAT115/BST282 Lab11

Annoucements

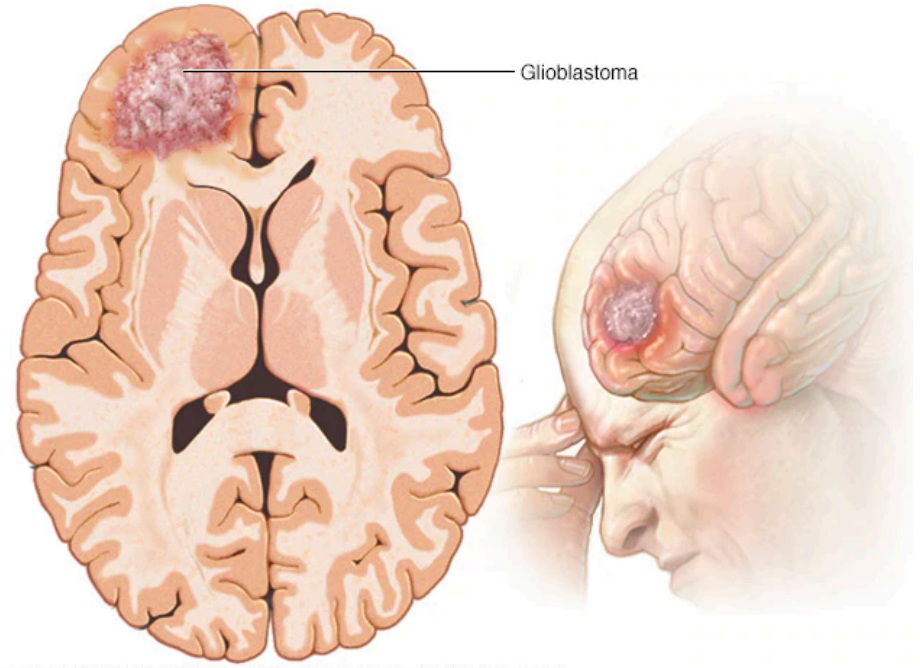
- Homework 6 due Wednesday April 29 @ 11:59pm
- OHs during times of regular lab sessions
- Some extra commented pseudocode will be provided but will not be covered in the lab

Outline

- Overview of HW6 questions
  - Concepts
  - Tools

Glioblastoma

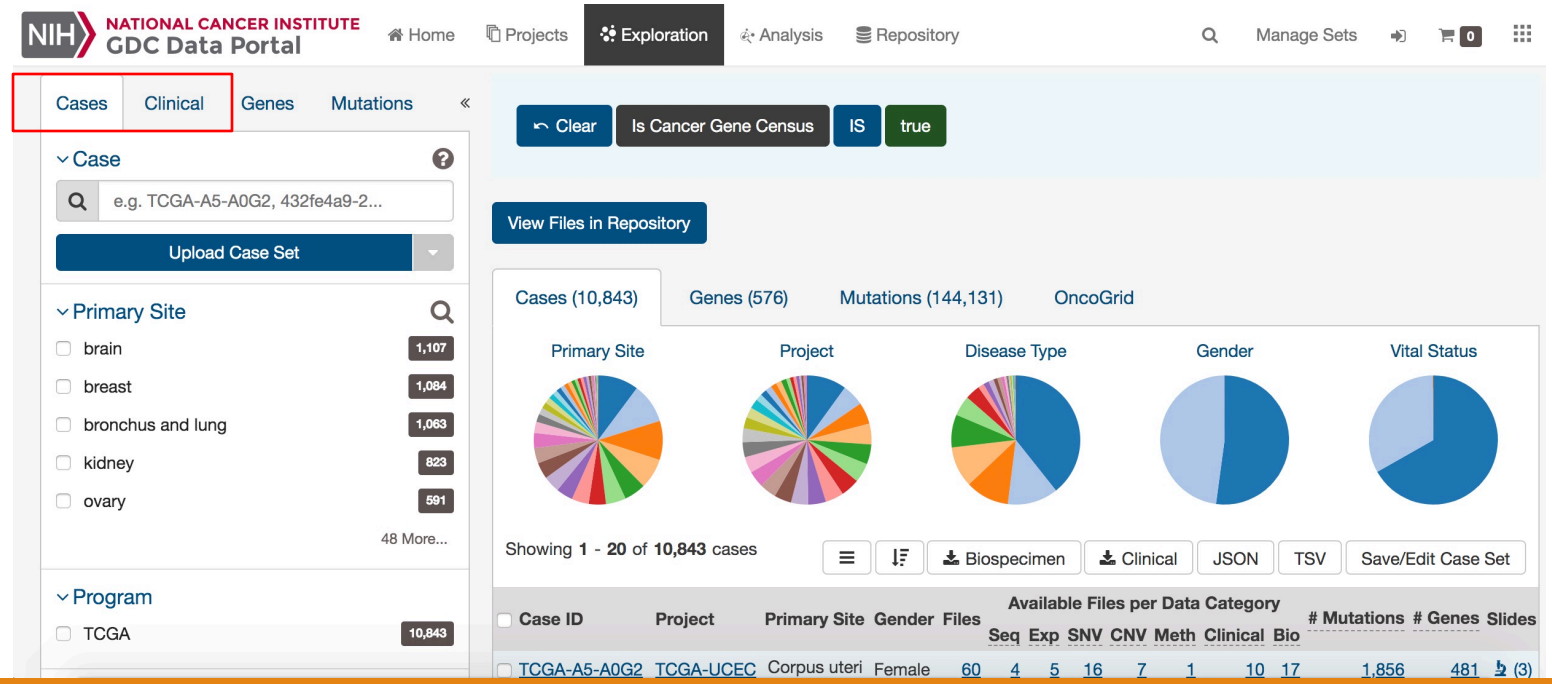
- Aggressive brain cancer
- Starts in astrocytes, a type of cells supporting nerves
- Difficult to treat and cure often impossible



Part I Q1: Exploration on TCGA

https://portal.gdc.cancer.gov/

- Contains raw data from profiling tumor samples in various cancer types
- Go to the homepage -> Exploration



Part I Q2

- Extract processed data from Broad Firehose (http://firebrowse.org/)
- Requires the package “FirebrowseR”
  - Helps us to easily download the processed data
  - Execute devtools::install_github(“mariodeng/FirebrowseR”) to install the package
  - Install devtools via install.packages(“devtools”) if not installed
  - Vignette: https://github.com/mariodeng/FirebrowseR/blob/master/vignettes/FirebrowseR.Rmd

Part I Q2

```{r}
# The method `Metadata.Cohorts` returns all cohort identifiers and their corresponding description
cohorts <- Metadata.Cohorts(format = "csv")
```

```{r}
head(cohorts)
```

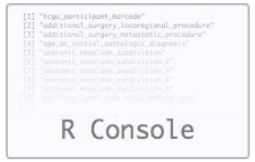
	cohort <chr>	description <chr>
1	ACC	Adrenocortical carcinoma
2	BLCA	Bladder Urothelial Carcinoma
3	BRCA	Breast invasive carcinoma
4	CESC	Cervical squamous cell carcinoma and endocervical adenocarcinoma
5	CHOL	Cholangiocarcinoma
6	COAD	Colon adenocarcinoma

```{r}
cohorts[cohorts$cohort == 'BRCA',]
```

	cohort <chr>	description <chr>
3	BRCA	Breast invasive carcinoma

Part I Q2

```r
#retrieve a list of all patients associated with this identifier
head(Samples.Clinical(cohort = "BRCA", format = "tsv"))
colnames(Samples.Clinical(cohort = "BRCA", format = "tsv"))
```



	tcga_participant_barcode <chr>	additional_surgery_locoregional_procedure <chr>
1	TCGA-E9-A2JT	*NA*
2	TCGA-BH-A0W4	*NA*
3	TCGA-BH-A0B5	*NA*
4	TCGA-AC-A3TN	*NA*
5	TCGA-BH-A0B3	*NA*
6	TCGA-A7-A0CD	*NA*

6 rows | 1-3 of 111 columns

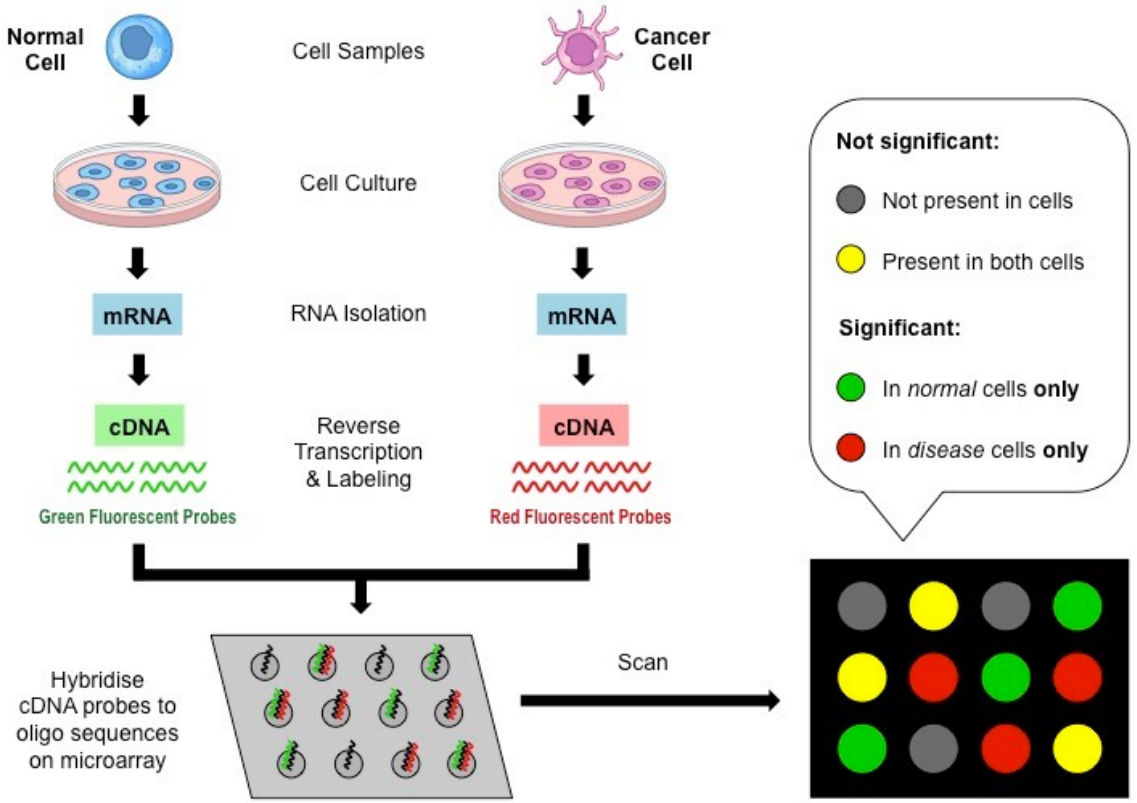
Part I Q2

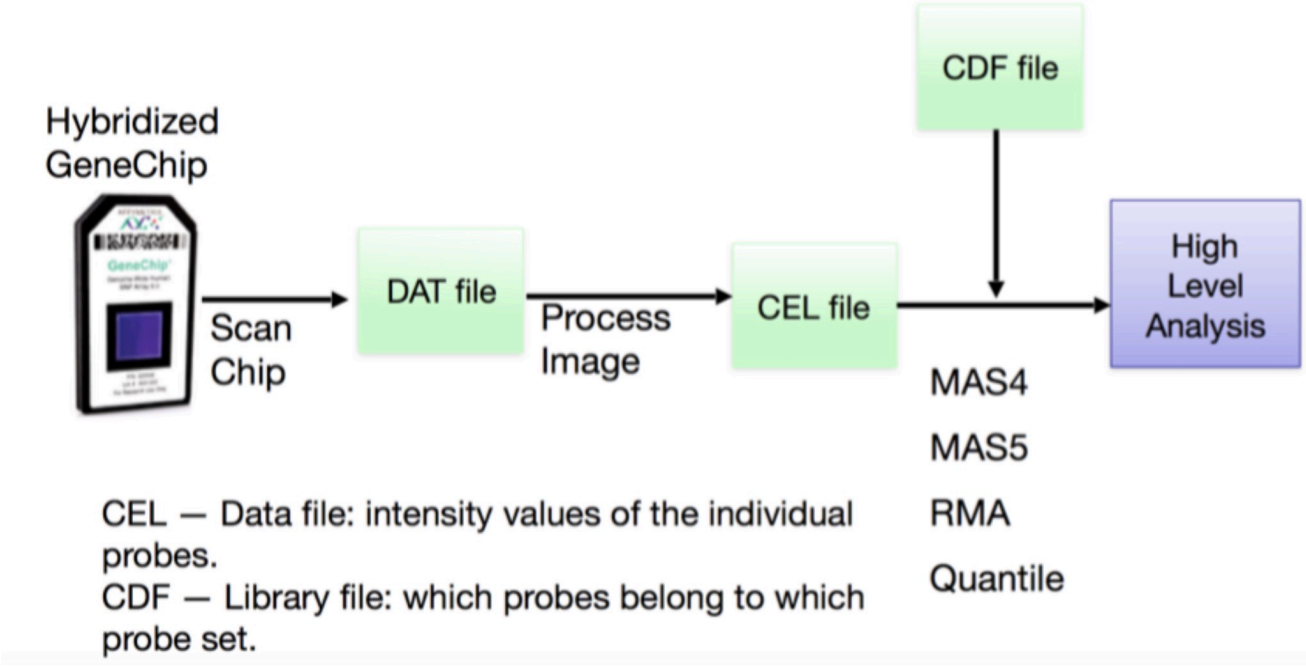
```
# code adapted from FirebrowseR vignette
# https://github.com/mariodeng/FirebrowseR/blob/master/vignettes/FirebrowseR.Rmd
all.Received <- FALSE
page.Counter <- 1
page.size <- 150
brca_pats <- list()

# looping because we can only download 150 patients at a time, we want to keep downloading till
we get all the patients with GBM
while(all.Received == FALSE) {
    brca_pats[[page.Counter]] <- Samples.Clinical(format = "csv",
            cohort = "BRCA", page_size = page.size, page = page.Counter)
    if(page.Counter > 1) {
        colnames(brca_pats[[page.Counter]]) <-
            colnames(brca_pats[[page.Counter-1]])
    }

    if(nrow(brca_pats[[page.Counter]]) < page.size) {
        all.Received = TRUE
    } else {
        page.Counter = page.Counter + 1
    }
}

brca_pats <- do.call(rbind, brca_pats)
dim(brca_pats)

```

The final data frame contains:

- Patients in rows
- Measures in columns
- Find the measure, then calculate the mean

Part II: Microarray data





Part II Q1: GBM data analysis

- Processed microarray data
  - 60 tumor samples with prefix “TCGA”
  - 10 normal samples with prefix “normal”
  - Log2-transformed
- K-means clustering on it(K=3) to detect whether different subtypes present
  - Using 1) all genes and 2) top 2000 most variable genes
  - Recommended settings for parameters using kmeans(): nstart = 10, iter.max = 100

	TCGA.02.0025 <dbl>	TCGA.02.0026 <dbl>	TCGA.02.0080 <dbl>	TCGA.02.0084 <dbl>	TCGA.02.0085 <dbl>
A2BP1	4.964801	5.105801	5.306692	6.061618	7.291034
A2M	10.869589	10.535573	10.554078	11.907391	11.615419
A4GALT	5.237567	4.830462	5.048806	4.839722	4.919547
A4GNT	4.384390	4.074379	4.454094	4.568200	4.244977
AAAS	5.137132	6.118595	6.102693	5.320051	5.683781
AACS	5.835520	6.860231	6.454327	5.899225	7.288678

6 rows | 1–6 of 70 columns

Part II Q2: Call differential genes using LIMMA

- LIMMA : Linear modeling approach
  - Use p-values obtained to call DE genes
- LIMMA fits the following model for each gene:
  - $y_{ij} = \alpha_j + \beta_j X_i$
  - y_{ij}: Expression index for gene j in sample I
  - α_j : Intercept / “baseline” for gene j
  - X_i : Covariate for subject I
  - β_j : Change in gene expression index
- Vignette: https://kasperdanielhansen.github.io/genbioconductor/html/limma.html

Part II Q2: LIMMA setup for HW6

- Recall that $y_{ij} = \alpha_j + \beta_j X_i$
  - Mean log2 expression of those in one subtype: $\alpha + \beta$
  - Mean log2 expression of those in the reference subtype: α
- To fit LIMMA, we need the data and a design matrix
  - Use the method model.matrix() to create a design matrix
  - Get summary statistics from fitted models
  - Subset the genes that have FDR < 0.05 and logFC > 1.5

	TCGA.02.0025 <dbl>	TCGA.02.0026 <dbl>	TCGA.02.0080 <dbl>	TCGA.02.0084 <dbl>	TCGA.02.0085 <dbl>
A2BP1	4.964801	5.105801	5.306692	6.061618	7.291034
A2M	10.869589	10.535573	10.554078	11.907391	11.615419
A4GALT	5.237567	4.830462	5.048806	4.839722	4.919547
A4GNT	4.384390	4.074379	4.454094	4.568200	4.244977
AAAS	5.137132	6.118595	6.102693	5.320051	5.683781
AACS	5.835520	6.860231	6.454327	5.899225	7.288678

6 rows | 1–6 of 70 columns

	(Intercept)	subtype2
1	1	0
2	1	1
3	1	1
4	1	1
5	1	0
6	1	1

PartII Q3: Methylation and expression

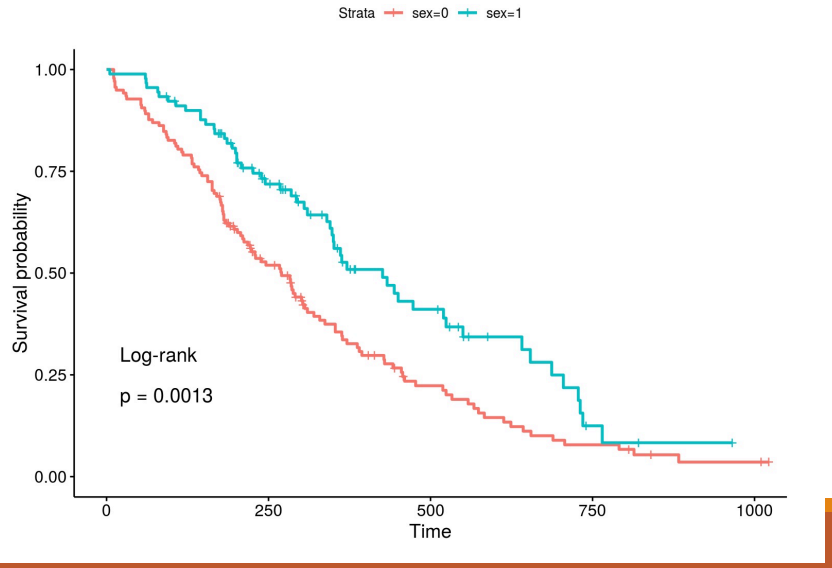
- Methylation data provided
  - Not all samples have methylation data
  - Recorded as a fraction from 0 to 1
  - Need to logit transform it so that it has the range from negative infinity to positive infinity before running LIMMA
- Run LIMMA to get the differentially methylated genes
  - Same threshold applied as in Q2
  - Make sure to create the design matrix for LIMMA correctly
- Plot methylation vs expression to see whether there is a relationship
- Find genes that are both differentially methylated and expressed

Part II Q4: Survival Analysis

- Survival data is unique in that the true survival time may be censored
- T_i: the time to event for the i-th individual
- C_i: the corresponding censoring time.
- We observe $Y_i = \min(T_i, C_i)$ and δ_i=I($T_i \leq C_i$) (i.e. δ_i= 1 if $T_i \leq C_i$ and δ_i= 0 if $T_i > C_i$)
- We also have predictors X_i for each individual.
- Survival function: $P(T_i > t)$

Part II Q4: Survival Analysis

- Survival data
  - Need to format the sample names so that it will be consistent with previous datasets
  - Pass in Y_i and δ_i into Surv(), then apply survfit()
    - **survfit(Surv**(Y_i, δ_i) ~ X_i, data = your_data_frame)
    - Plot the Kaplan-Meier Curve
  - Logrank test compares the survival curves across the observed time frame. Significant p-value means the two curves are different.



	vital.status <int>	days.to.death <int>	days.to.last.followup <int>
TCGA-08-0509	1	382	17
TCGA-08-0510	1	130	107
TCGA-12-0620	1	318	181
TCGA-12-0772	1	1638	1615
TCGA-12-0775	1	232	167
TCGA-12-0818	1	2791	2791
TCGA-12-0827	1	1179	1179
TCGA-12-1090	1	231	190
TCGA-14-0783	1	189	189
TCGA-14-1456	0	*NA*	1246

Part II Q5: Cox Regression

Hazard function $\lambda(t)$ is defined as :

$$\lambda(t) = \lim_{\delta \to 0} \frac{1}{\delta} P(t \leq T < t + \delta | T \geq t).$$

- Interpretation: instantaneous rate at time t, given that the event has not occurred prior to time t.

Cox proportional hazards model is defined as:

$$\lambda(t_i) = \lambda_0(t_i) \exp(X_1 \beta_1 + \cdots + X_p \beta_p)$$

- $\lambda_0(t_i)$ is the baseline hazard when all $X_i = 0$.
- The likelihood ratio/Score/Wald test output shows the predictive power of the model, compared to a model without any covariates.
  - Significant p-value indicates model is performing better than model without any covariates.

Part II Q5: Cox Regression

- Pass in Y_i and δ_i into Surv(), then apply coxph()
  - **coxph**(**Surv**(Y_i, δ_i) ~ X_i, data = your_data_frame)
  - Use summary() to see the summary statistics

```
## Call:
## coxph(formula = Surv(time, death) ~ sex, data = lung2)
##
##   n= 228, number of events= 165
##
##        coef exp(coef) se(coef)      z Pr(>|z|)
## sex -0.5310    0.5880   0.1672 -3.176  0.00149 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

- In Q5, we want to model with genes as covariates, but there is a potential problem:
  - Way too many genes comparing to samples
  - Need to select relevant genes so as to shrink the number of predictors before performing the regression
  - LASSO!
  - How can we do this?

Part II Q5: Cox Regression

- Data wrangling to get a data frame in a similar format

	Time	Death	Diff_gene1	Diff_gene2	…	Diff_gene3
Sample_1	…	…	…	…	…	…

- Run LASSO on the current dataset, perform cross-validation to select the best lambda for selecting the set of relevant genes that have the strongest predictive ability
  - cv.glmnet(all_covariates, surv_object, family = "cox")
- Using the optimal lambda('lambda.min'), get the non-zero "active" coefficients after shrinkage
- The covariates with non-zero coefficients will be the covariates for running cox regression
  - **coxph**(**Surv**(Y_i, δ_i) ~ X_i, data = your_data_frame)
  - Look at its summary statistics to see whether the model has significant predictive power

Part II Q6: Cox Regression

- Do the differentially expressed genes really outperform the other genes?
- Randomly parse out a number of genes 100 times
  - Of your choice, but would be more relevant if the number of covariates is close to your optimal model
  - Run run cox regression 100 times to get 100 cox models
- How do we compare the different models?
  - One possible way is AIC
- **AIC**: With difference in parameters = k, the AIC is $2k-2\log(L)$.
  - log(L) : log-likelihood
  - Smaller is better
  - If a model has too many useless covariates(large k), then will be penalized
- A possible way to approach the problem is to evaluate the AIC of your model in Q5 and compare it to the AICs from the 100 randomly selected models
  - Does any random model happen to have smaller AIC than our optimized model?