Adenoid Cystic Carcinoma Risk Visualizer

Introduction

Adenoid Cystic Carcinoma (ACC) is a rare form of cancer characterized by malformed mucus membranes and clustered fibrous structures. It most commonly develops in oral and throat tissues. It can also spread to the bronchial tubes, where growths can dangerously obstruct breathing. It generally grows slowly but can co-opt the lymphatic system, at which point it quickly becomes metastatic.

Adenoid cystic carcinoma has an etiology that is unclear. However, current research reveals that genetic alterations (mutations) are the primary cause of cellular malignancy in many malignancies, including ACC. And the activation of the oncogenic transcription factor gene MYB is the most common genetic event associated with ACC. MYB regulates genes involved in cell cycle control, DNA replication and repair, and RNA processing, as well as driving ACC cell proliferation. As a result, the MYB oncogene could be used to diagnose and treat ACC.[1]

In order to visualize the landscape of ACC mutations, the total number of mutations, the fraction of genome altered, and tumor mutational burden (TMB) are specifically interpreted in this program. The number of total mutations refers to the number of mutations detected in the tumor genome. Fraction of genome altered refers to the percentage of the genome that has been modified by copy number gains or losses. [2] And TMB represents the number of proteins possessing a non-synonymous mutation, which is a measure of the number of gene mutations (changes) within cancer cells. [3] These characteristics are beneficial to genetic researchers since they provide more detailed information on the genomic make-up of malignancies.

Patients may survive for years with metastases because this tumor is generally well-differentiated and slow-growing. According to a 1999 study of a cohort of 160 ACC patients (Pisani), adenoid cystic carcinoma patients have an 89 percent five-year survival rate overall. Less than 70% of people survive for ten years. And death from late-occurring metastatic illness accounted for only 40% of deaths at 15 years. [4]

Therefore, the goal of this project is to design a shiny app that can visualize the morbidity and mortality of 1049 Adenoid Cystic Carcinoma patients as well as determine the ACC mutational landscape. Because of this capacity to abruptly change rate and type of spread, it’s particularly important to understand the different forms of this type of cancer to form an accurate prognosis.

Methods

Shiny App

Shiny is an R package that enables web apps with an R back end. It blends R's computational capability with a clean and simple UI. In order to create the shiny app, two scripts are required: UI and server. UI is a user-interface script that allows you to customize the appearance of your program. The computer is communicated using a server script. For our project, we all did both UI and server for two web pages. Our project proposal is to visualize the ACC mutational landscape and therefore understand various forms of this cancer better.

R version 3.6.1 was used to conduct all of the analyses.

Library

Shiny, shinythemes, shinydashboard, shinywidgets, readr, readxl, dplyr, tidyr, janitor, and ggplot2 were used in this project. Readr and readxl were first used to read imported data. Janitor and tidyr were then used to examine and clean dirty data. After obtaining tidy data, dplyr was used for data manipulation. Shiny, shinythemes, shinydashboard, shinywidgets were used to build and design the shiny app. And ggplot2 were used to build scatter plots in the web page.

Dataset

The data were acquired from cBioPorta, an exploratory analytic platform for investigating large-scale cancer genomic data sets that contains data from big consortium projects. (<https://www.cbioportal.org/study/summary?id=acc_2019>)

The original data set was break down into subgroups with interested variables. This result to following data set “Cnancer\_type, KM\_Plot Overall\_Survival\_(months), Overall\_Survival\_Status, Somatic\_Status, Mutated\_Genes, Mutation\_Count,” and “Fraction\_Genome\_Altered”. The first four datasets focused on the morbidity and mortality of 1049 Adenoid Cystic Carcinoma patients, while the rest of the datasets were further combined into “Mutation\_Count\_VS\_Fraction\_Genime\_Altered” to provide interrelated variables that could be used to plot graphs and determine the ACC mutational landscape.

Morbidity and Mortality

First, tables of cancer type, somatic status, and survival status of the patients with selections of different visualization methods. Data on overall survival in months was also included for further reference. Functions to provide users the choice of filtering and comparing between groups are also added. There are six dropdown menus in total, which are “Data source”, “Cancer type”, “Sex”, “Oncotreecode”, “Somatic status”, and “Sample type”. The keyword search bar is also available, which makes it easy to search study id, patient id and sample id in the dataset and return to the specific search results.

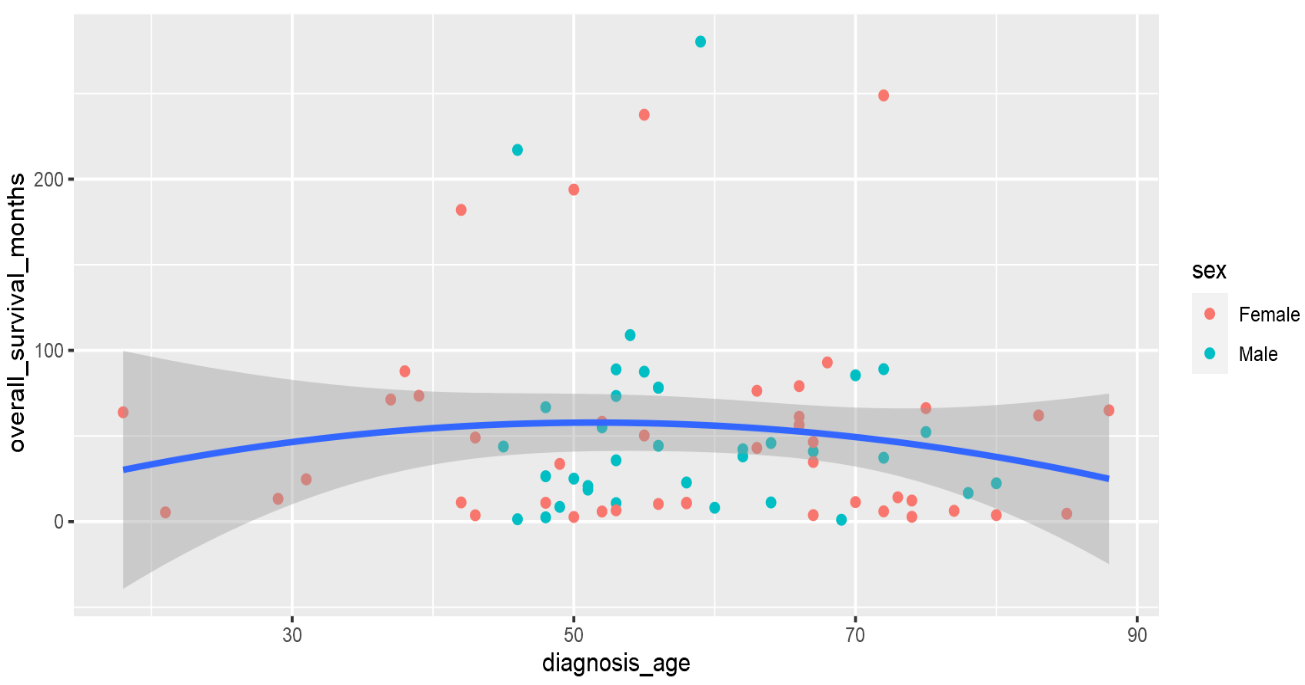
ACC Mutational Landscape

Second, a three-variable scatter plot was created from the table. The total number of mutations and frequency of each genotype were added to relate to tumor mutational burden (TMB), which is also merged in a plot visualizer of Mutation counts vs. fraction of genome altered with the option of various scales to find correlations with specific alterations or clinical attributes, such as the survival or somatic stage.

Overall, we used this program to understand the relationship between genetic burden and prognosis in salivary cancer, across different groups such as sex and age at diagnosis.  This could help us better understand the risks posed by different types of cancer and better differentiate cancers in the same tissue.  The user can also fit linear, polynomial, and logarithmic curves to the data to test different predictions.  The X and Y axis can be any combination of variables, with a third variable acting as the point color.

Results & Discussions

We found that survival, in months, decreased with both age at onset and fraction genome altered. The former effect was much stronger in males than in females, and the latter effect was very faint, and obscured by small overall sample size and variance.



Most of the other relationships within this dataset were either trivially true, such as that between tumor mutational burden and fraction genome altered, or the dataset was too small to determine their validity.

This app could be used for other cancer data from the cBioPortal website, to easily create and alter visualizations for cancer data and better understand how different phenomena are related within specific cancers. While the prognosis for salivary cancers has an atypical range, as they change their growth rate so quickly, late in their development, it is nonetheless useful for all types of cancer.

It could be further modified to fit specific models and test them on different data sets and at different sample sizes. The current simple equations primarily act as a proof of concept for this application’s curve-fitting and error bands. This would make it more useful for testing specific hypotheses or changing the model for the same dataset repeatedly with few changes to the underlying R code.

This could also help educate people who would otherwise be averse to writing their own R Scripts about model fitting. Being able to quickly change the x and y axes, as well as sample size shows how those impact how results appear.

References

1. Dillon PM, Chakraborty S, Moskaluk CA, Joshi PJ, Thomas CY. Adenoid cystic carcinoma: A review of recent advances, molecular targets, and clinical trials. Head Neck. 2016;38(4):620-627. doi:10.1002/hed.23925
2. Fraction of Genome Altered and Total Mutations added to cBioPortal Plots tab, The Thyve, https://www.thehyve.nl/articles/fraction-of-genome-altered-total-mutations-cbioportal
3. Tumor Mutational Burden (TMB), CARISMI, https://www.carismolecularintelligence.com/tumor-mutational-burden-tmb/
4. Pisani, P., Airoldi, M., Allais, A., Aluffi Valletti, P., Battista, M., Benazzo, M., Briatore, R., Cacciola, S., Cocuzza, S., Colombo, A., Conti, B., Costanzo, A., Della Vecchia, L., Denaro, N., Fantozzi, C., Galizia, D., Garzaro, M., Genta, I., Iasi, G. A., Krengli, M., … Zigliani, A. (2020). Metastatic disease in head & neck oncology. *Acta otorhinolaryngologica Italica : organo ufficiale della Societa italiana di otorinolaringologia e chirurgia cervico-facciale*, *40*(SUPPL. 1), S1–S86. https://doi.org/10.14639/0392-100X-suppl.1-40-2020

Appendix: R code

---

title: "Final Project"

author: "Joshua Witten & Yufeng Zhu"

date: "4/22/2022"

output: html\_document

---

```{r setup, include=FALSE}

knitr::opts\_chunk$set(echo = TRUE)

```

```{r}

library(ggplot2)

library(dygraphs)

library(dplyr)

library(tidyr)

library(readr)

library(readxl)

library(dplyr)

library(shiny)

library(shinyWidgets)

library(shinydashboard)

library(shinythemes)

library (janitor)

```

```{r}

cancer <- read.delim("Cancer\_Type.txt")%>% clean\_names()

fga <- read.delim("Fraction\_Genome\_Altered.txt")%>% clean\_names()

survivalm <- read.delim("KM\_Plot\_\_Overall\_Survival\_\_(months).txt")%>% clean\_names()

mugenes <- read.delim("Mutated\_Genes.txt")%>% clean\_names()

mucount <- read.delim("Mutation\_Count.txt")%>% clean\_names()

muvsfra <- read.delim("Mutation\_Count\_vs\_Fraction\_Genome\_Altered.txt")%>% clean\_names()

survival <- read.delim("Overall\_Survival\_Status.txt")%>% clean\_names()

somatic <- read.delim("Somatic\_Status.txt")%>% clean\_names()

```

```{r}

options(max.print = 15000)

gene<-read.table(file = 'acc\_2019\_clinical\_data.tsv', sep = '\t', header = TRUE) %>% clean\_names()

```

```{r}

overall <- read\_tsv("acc\_2019\_clinical\_data.tsv")

overall<-overall%>%drop\_na()%>%clean\_names()%>%select(-c(cancer\_type,cancer\_type\_detailed, study,study\_id,oncotree\_code, patient\_id, sample\_id, data\_source))

overall

```

```{r}

ui <- fluidPage(

titlePanel("Basic Dataset"),

# Create a new Row in the UI for selectInputs

fluidRow(

column(4,

selectInput("data\_source",

"Data source:",

c("All",

unique(as.character(gene$data\_source))))

),

column(4,

selectInput("cancer\_type",

"Cancer type:",

c("All",

unique(as.character(gene$cancer\_type))))

),

column(4,

selectInput("sex",

"Sex:",

c("All",

unique(as.character(gene$sex))))

),

column(4,

selectInput("oncotree\_code",

"oncotree\_code:",

c("All",

unique(as.character(gene$oncotree\_code))))

),

column(4,

selectInput("somatic\_status",

"Somatic status:",

c("All",

unique(as.character(gene$somatic\_status))))

),

column(4,

selectInput("sample\_type",

"Sample type:",

c("All",

unique(as.character(gene$sample\_type))))

)

),

# Create a new row for the table.

DT::dataTableOutput("table"),

#1. Select 1 of 3 continuous variables as y-variable and x-variable

selectInput("y\_varb", label="Y-axis variable",choices=names(overall)),

selectInput("x\_varb", label="X-axis variable", choices=names(overall)), #2.

selectInput("cat\_colour", label="Select point color variable", choices=names(overall)), #3. Select sample size

selectInput("sample\_sz", label = "Select sample size", choices = c(50:137)),

#4. Three different types of linear regression plots

selectInput("formula", label="Formula", choices=c("y~x", "y~poly(x,2)", "y~log(x)")),

#5. Reset plot output after each selection

plotOutput("plot", dblclick = "plot\_reset")

)

server <- function(input, output) {

# Filter data based on selections

output$table <- DT::renderDataTable(DT::datatable({

data <- gene

if (input$cancer\_type != "All") {

data <- data[data$cancer\_type == input$cancer\_type,]

}

if (input$oncotree\_code != "All") {

data <- data[data$oncotree\_code == input$oncotree\_code,]

}

if (input$sex != "All") {

data <- data[data$sex == input$sex,]

}

if (input$data\_source != "All") {

data <- data[data$data\_source == input$data\_source,]

}

if (input$somatic\_status != "All") {

data <- data[data$somatic\_status == input$somatic\_status,]

}

if (input$sample\_type != "All") {

data <- data[data$sample\_type == input$sample\_type,]

}

data

}))

#1. Register the y-variable selected, the remaining variables are now options for x-variable

remaining <- reactive({

names(overall)[c(-1,-3,-4,-match(input$y\_varb,names(overall)))]

})

observeEvent(remaining(),{

choices <- remaining()

updateSelectInput(session = getDefaultReactiveDomain(),inputId = "x\_varb", choices = choices)

})

output$plot <- renderPlot({

#Produce scatter plot

subset\_data<-overall[1:input$sample\_sz,]

ggplot(subset\_data, aes\_string(input$x\_varb, input$y\_varb))+

geom\_point(aes\_string(colour=input$cat\_colour))+

geom\_smooth(method="lm",formula=input$formula)}, res = 96)

}

# Run the application

shinyApp(ui = ui, server = server)

```

Also available on GitHub: https://github.com/YUZ132/final-project-R