香港中文大學
The Chinese University of Hong Kong

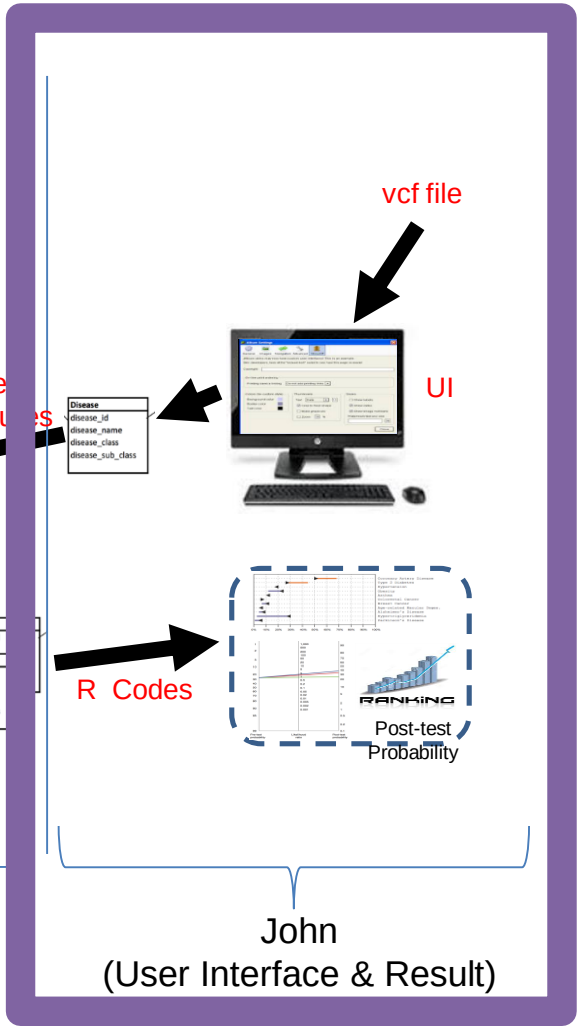
Database Construction and Knowledge Discovery for Personalized Medicine - Clinical Assessment Aids-

MSc in Genomics & Bioinformatics
By
Ho Ping Chong (Student ID 1155057016)

April 2016

Focus/Effort Allocation

Objective: Design and implement user interface, data visualization and interfacing with data mart.



Ken
(Data Acquisition)

Edward
(Database Design)

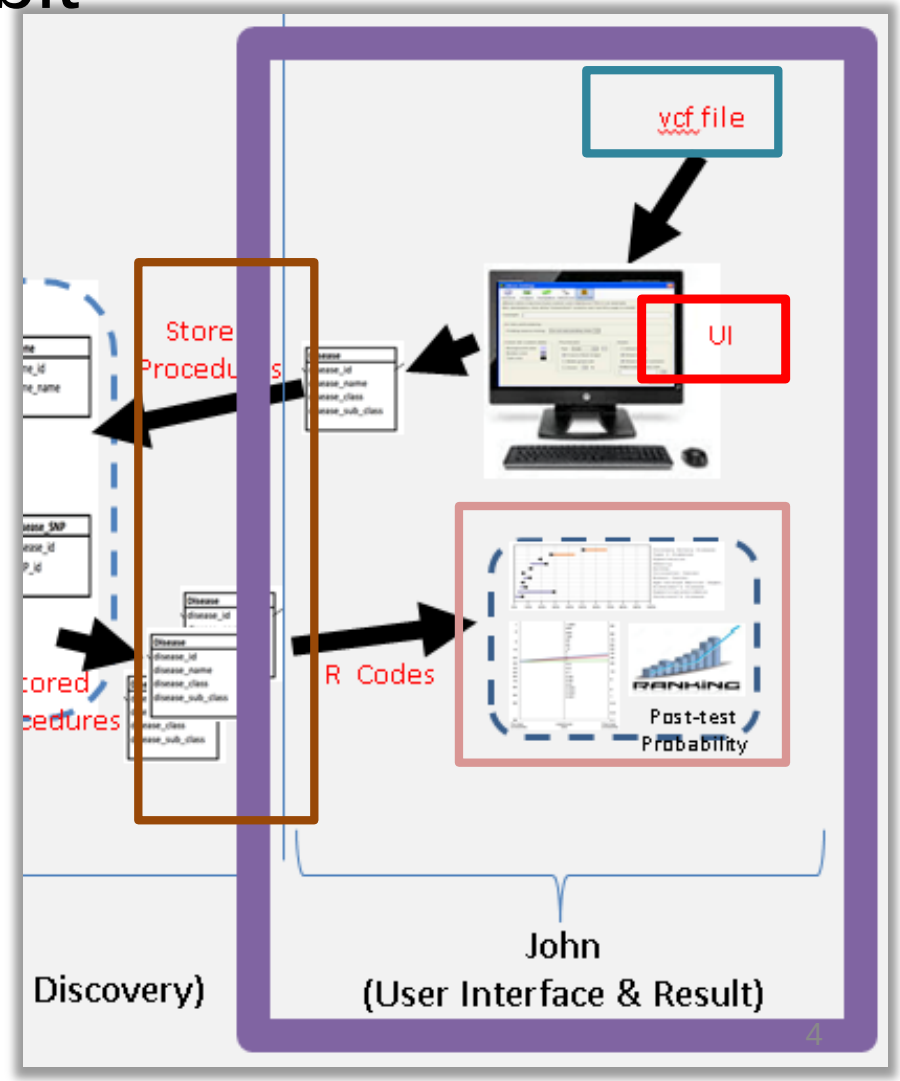
Tony
(Data Mart Design & Knowledge Discovery)

Agenda

- Cloud Based Technology Platform
- Workflow
  - Input, interfacing, presentation, screenshot
- Discussion
  - Evaluation, limitation & difficulty, finding

Technology platform

- Ubuntu 14.04 LTS 64-bit
- MySQL 5.7
- R studio
  - R 3.2.3
  - Package
    - Shiny
    - RMySQL
    - plotly
    - vcfR
    - tidyr



Technology platform

server.R

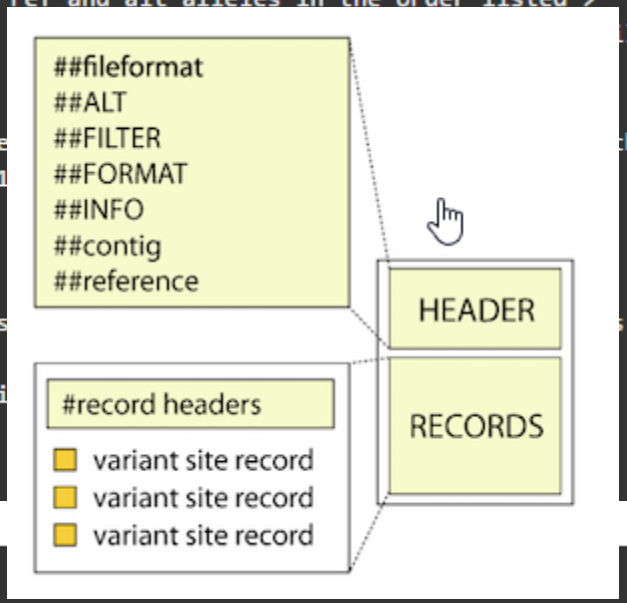
```
library(shiny)
library(datasets)
library(tidyr)
library(plotly)
library(RMySQL)
library(vcfR)

#connect MySQL PM.db
mydb = dbConnect(MySQL(), user='root',

shinyServer(function(input, output, se
#upload VCF file, ceate trigger event
  observeEvent(input$file1, {
    inFile <- input$file1
#null if no file was selected
    if (is.null(inFile))
      return(NULL)
#upload file from directory and skip t
    tempfile <- read.table(inFile$data
#Prepare latest job id a key
    max_job_id <- (as.numeric(dbGetQue
#Porvide insert SQL on FEI_job
    temp <- paste("INSERT INTO FEI_Job
#Append FEI_vcf with job id, SNP refer
#Use vcfR package to import VCF file a
    inFile1 <- read.vcfR(inFile$datapa
#Use vcfR method to get genotype from
    DataFile <- extract.gt(inFile1, el
#Remove "/" in genotype value
    DataFile <- gsub("/","",DataFile)
#Aggregrate job_id, snp_id and genotyp
    InsertFEI_vcf <- list(max_job_id,t
#cast as data frame
    InsertFEI_vcf <- as.data.frame(Ins
#rename column name of the data frame
    colnames(InsertFEI_vcf) <- c("job_
#clean up FEO_Riskogram_Data and FEO_R
    dbSendQuery(mydb,"truncate table F
    dbSendQuery(mydb,"truncate table F
#Write to MySQL PM database as table F
    dbWriteTable(mydb, "FEI_vcf", Inse
#insert new record with update job_id
    dbSendQuery(mydb,temp)

  })
```

ui.R

```
library(shiny)
library(plotly)


shinyUI(fluidPage(

  titlePanel("Disease risk predictor"),
  headerPanel("Input detail"),

   sidebarPanel(
    fileInput('file1', 'Upload a VCF File',
              accept=c('text/csv',
              'text/comma-separated-values,text/plain',
              '.csv/.vcf')),

   textInput("patientid", "Patient id", value = "90435378"),

   selectInput("age","Age",
              choices = c("Adult" = "A",
                          "Child" = "C",
                          "Newborn" = "N")),
    selectInput("sex","Sex",
              choices = c("Male" = "M",
                          "Female" = "F")),
    selectInput("race","Location",
              choices = c("Hong Kong" = "HK",
                          "World" = "World")),
    br(),
    actionButton("submit", "Go Prediction!"),
    br(),br(),
    selectInput("disease", "Choose a disease",
              choices = c ("Prostate Cancer" = "Ca Prostate", "Colon Cancer" = "Ca colon"))

    ),

  mainPanel(
    tabsetPanel(
        tabPanel("VCF",tableOutput("viewVCF")),
        tabPanel("Risk List",br(),column(2,tableOutput("view"))),
        tabPanel("Risk-o-gram",br(),plotlyOutput("disSPlot", "500px","200px"),br(),br(),column(6,tableOutput("view2"))),
        tabPanel("Disease Risk",br(),plotlyOutput("distPlot","250px","400px"),br(),br(),column(6,tableOutput("view1")))
            )
    )

))
```

Input - Variant call format (VCF)

```
##fileformat=VCFv4.1
##FILTER=<ID=LowQual,Description="Low quality">
##FORMAT=<ID=AD,Number=.,Type=Integer,Description="Allelic depths for the ref and alt alleles in the order listed">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Approximate read depth
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=PL,Number=G,Type=Integer,Description="Normalized, Phred-scale
##GATKCommandLine.HaplotypeCaller=<ID=HaplotypeCaller,Version=3.4-3-gd1ac1
.
.
.
##INFO=<ID=AC,Number=A,Type=Integer,Description="Allele count in genotypes
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency, for each
##INFO=<ID=AN,Number=1,Type=Integer,Description="Total number of alleles i
##contig=<ID=chr1,length=249250621,assembly=b37>
##reference=file:human_genome_b37.fasta
```




```
[HEADER LINES]
#CHROM  POS ID      REF ALT QUAL    FILTER  INFO          FORMAT          NA12878
1   873762  .       T   G   5231.78 PASS    [ANNOTATIONS] GT:AD:DP:GQ:PL  0/1:173,141:282:99:255,0,255
1   877664  rs3828047   A   G   3931.66 PASS    [ANNOTATIONS] GT:AD:DP:GQ:PL  1/1:0,105:94:99:255,255,0
1   899282  rs28548431  C   T   71.77   PASS    [ANNOTATIONS] GT:AD:DP:GQ:PL  0/1:1,3:4:26:103,0,26
1   974165  rs9442391   T   C   29.84   LowQual [ANNOTATIONS] GT:AD:DP:GQ:PL  0/1:14,4:14:61:61,0,255
```

Standardized text file format for representing SNP, indel, and structural variation calls

Input – VCF genotype information

```
1   873762   .          T   G   [CLIPPED] GT:AD:DP:GQ:PL    0/1:173,141:282:99:255,0,255
1   877664   rs3828047  A   G   [CLIPPED] GT:AD:DP:GQ:PL    1/1:0,105:94:99:255,255,0
1   899282   rs28548431 C   T   [CLIPPED] GT:AD:DP:GQ:PL    0/1:1,3:4:26:103,0,26
```

- 0/0 - the sample is homozygous reference
- 0/1 - the sample is heterozygous, carrying 1 copy of each of the REF and ALT alleles
- 1/1 - the sample is homozygous alternate

`extract.gt` *Extract elements from vcfR objects*

Description

Extract elements from the 'gt' slot, convert extracted genotypes to their allelic state, extract indels from the data structure or extract elements from the INFO column of the 'fix' slot.

Usage

```
extract.gt(x, element = "GT", mask = FALSE, as.numeric = FALSE,
  return.alleles = FALSE, allele.sep = "/", extract = TRUE)
```

Package 'vcfR' - Description Tools for working with variant call format ('VCF') files. Reads in 'VCF', 'GFF' and 'FASTA' data for visualization. Includes tools for filtering and writing to 'VCF' files.

Title: Tools for Working with Variant Call Format ('VCF') Files
February 22, 2016
https://cran.r-project.org/web/packages/vcfR/vcfR.pdf

Interface - RMySQL

- MySQL Connection
- Database name
- Username
- Password
- Host Details

```
# create a MySQL connection object
con <- dbConnect(MySQL(),
                 user = 'root',
                 password = 'password',
                 host = 'localhost',
                 dbname = 'world')
```

```
> dbGetQuery(con, "SELECT * FROM trial LIMIT 5;")
```

```
> res <- dbSendQuery(con, "SELECT * FROM trial;")
```

```
# create table in the database
> dbWriteTable(con, "trial", trial)
[1] TRUE
```

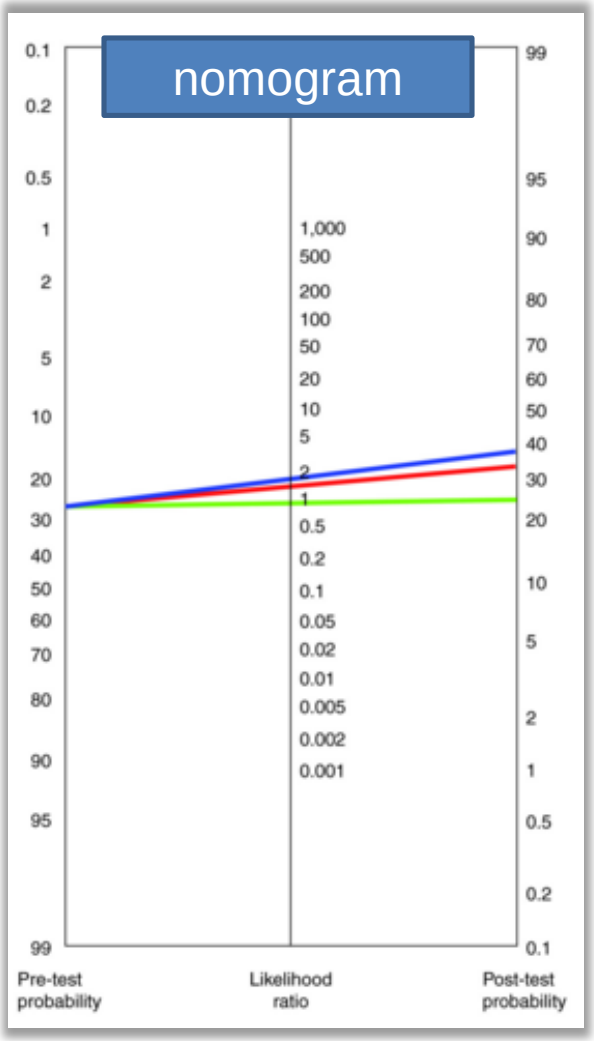
Presentation - Principle

“Clinicians begin with pretest estimates of disease probability and then adjust the probability as new diagnostic information arrives….”

JAMA. 1999 Apr 7;281(13):1214-9

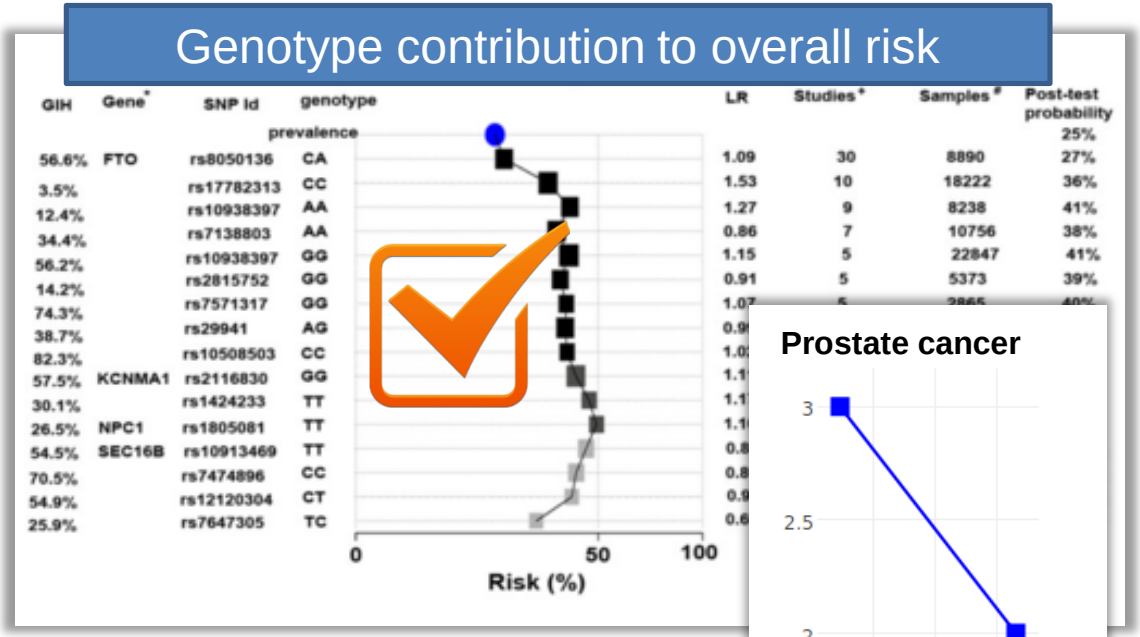
Users' guides to the medical literature: XV. How to use an article about disease probability for differential diagnosis. Evidence-Based Medicine Working Group.

Presentation – data visualization



Morgan et al. Genome medicine 2010, 2:30
Likelihood ratios for genome medicine

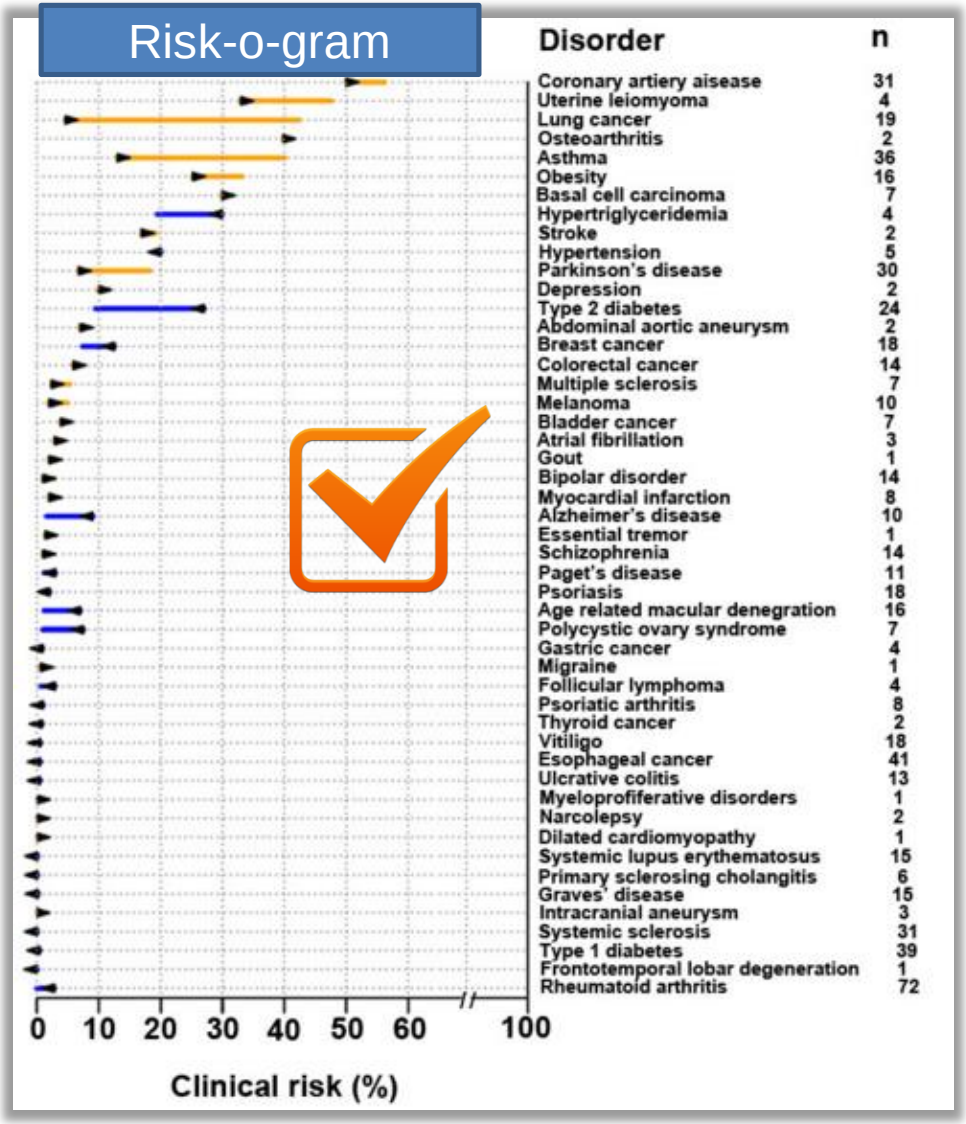
Gupta et al. BMC Genomics 2012, 13:440
Sequencing and analysis of a South Asian-Indian personal genome



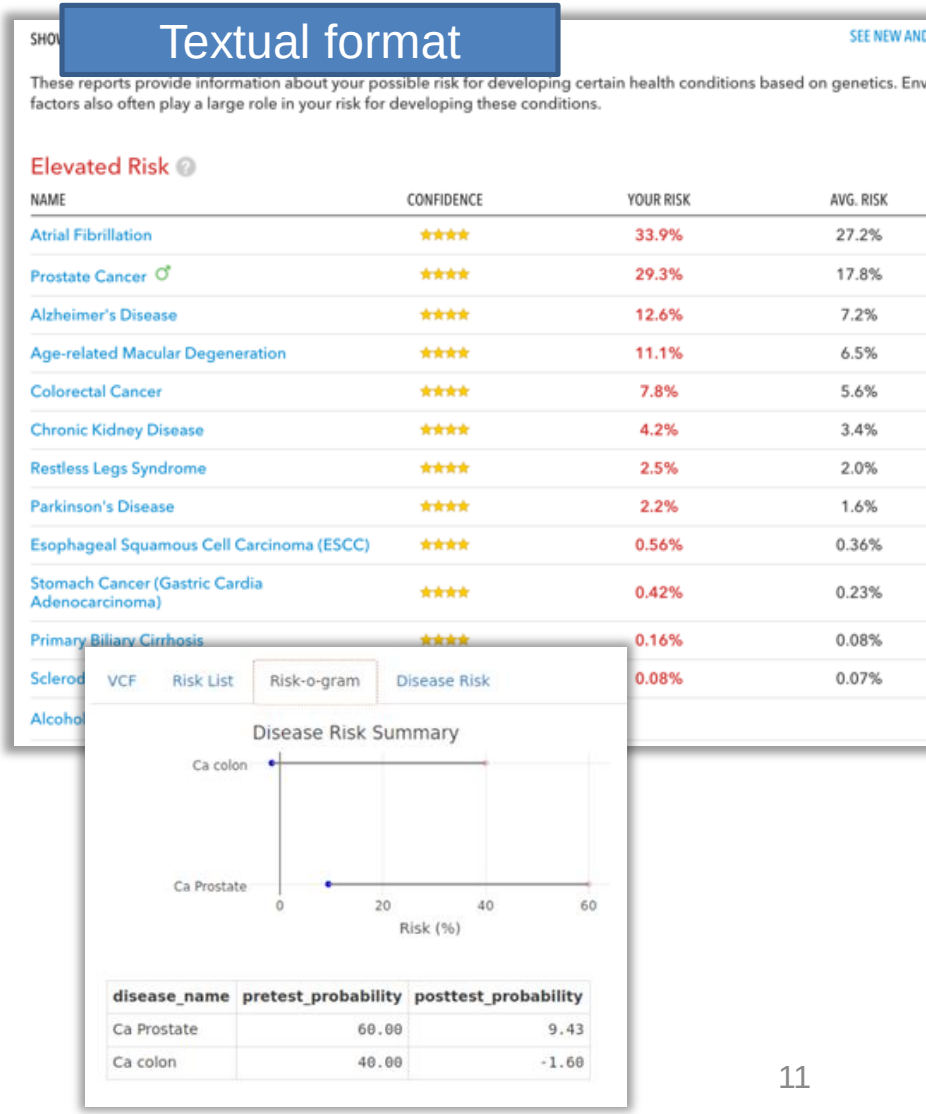
Presentation – data visualization

Gupta et al. BMC Genomics 2012, 13:440
Sequencing and analysis of a South Asian-Indian personal genome

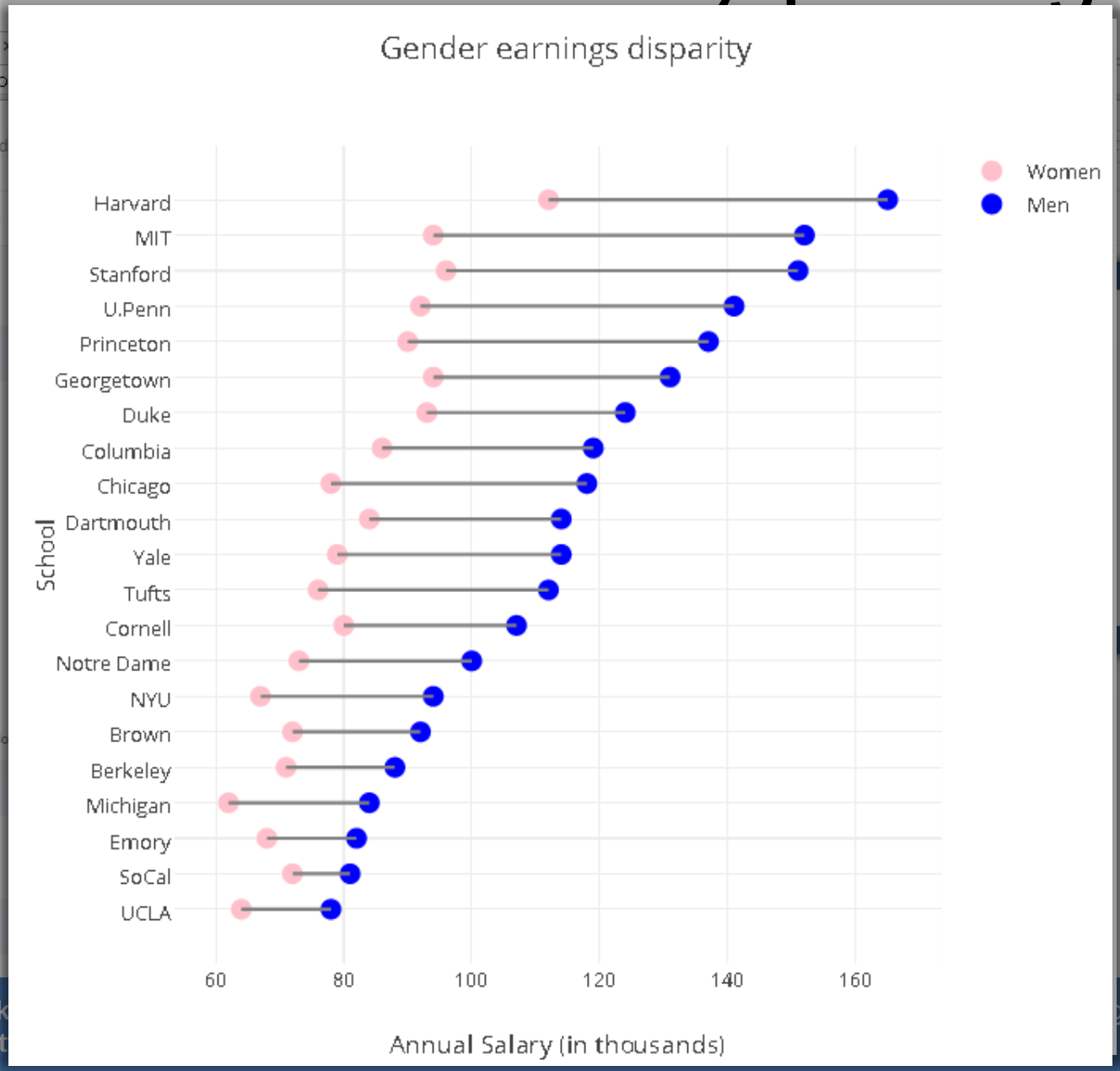
Risk-o-gram



Textual format

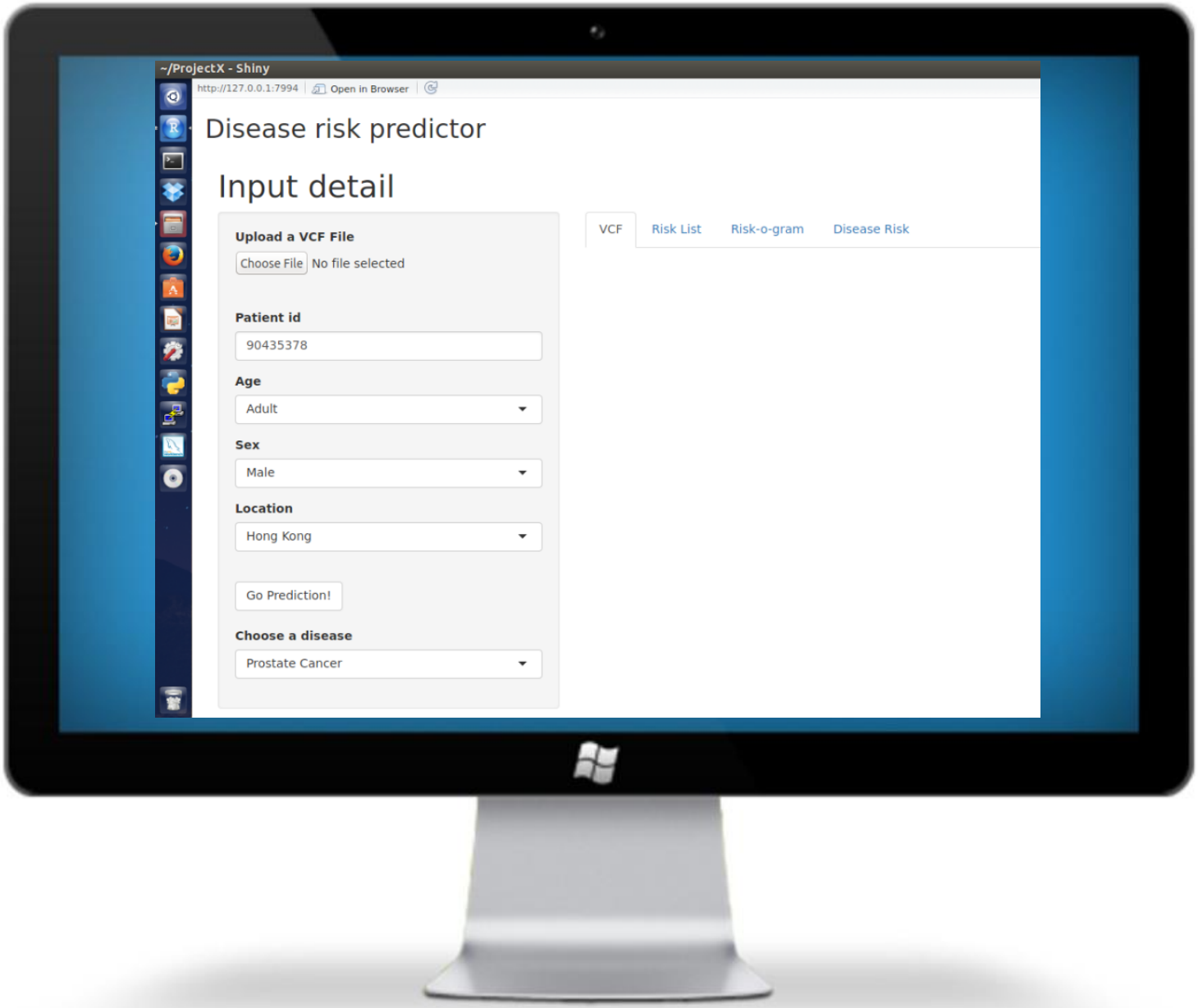


Presentation - Plotly package

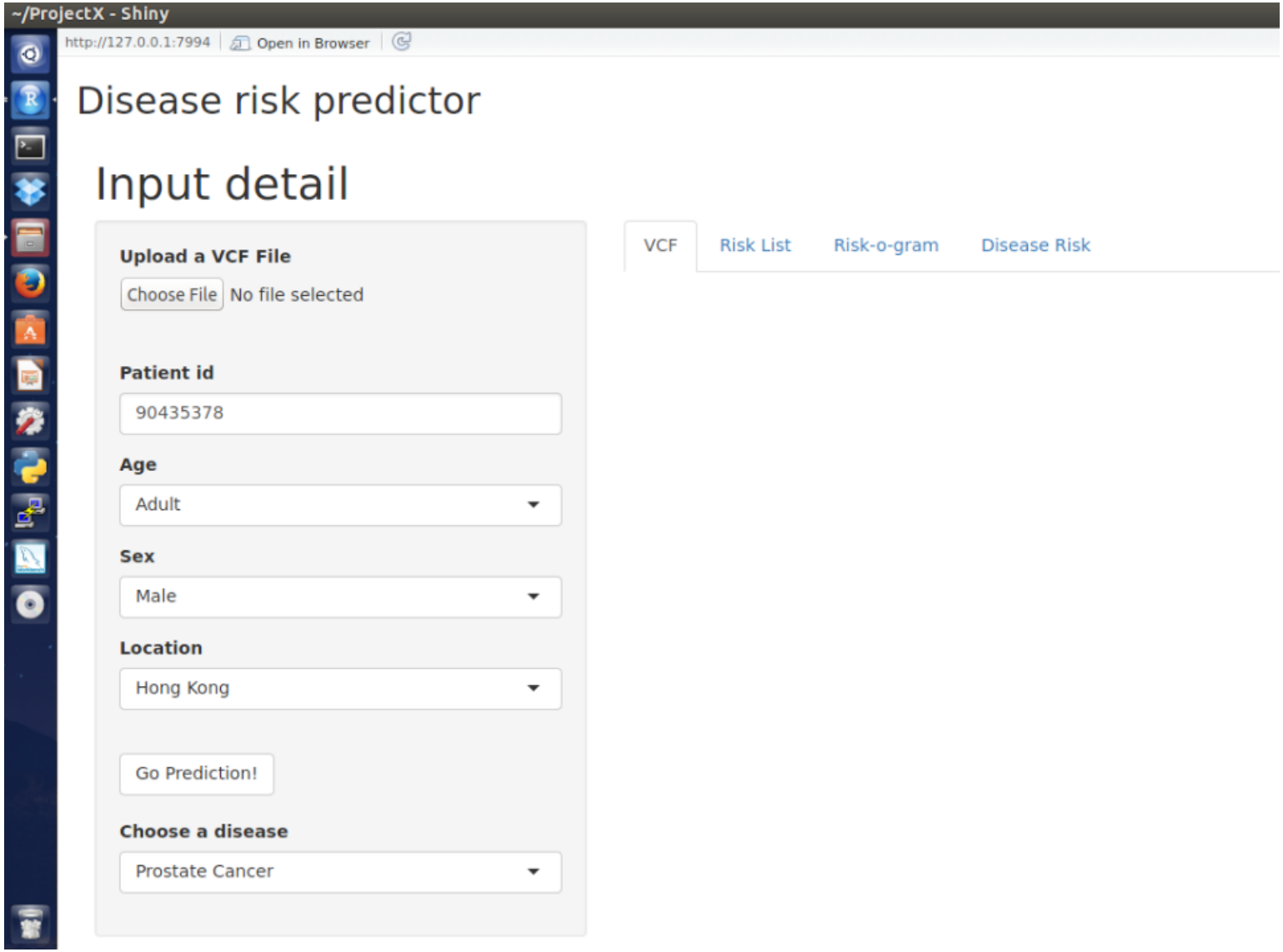


Plotly is an R pack
https://plot.ly/r/get
library plotly.js

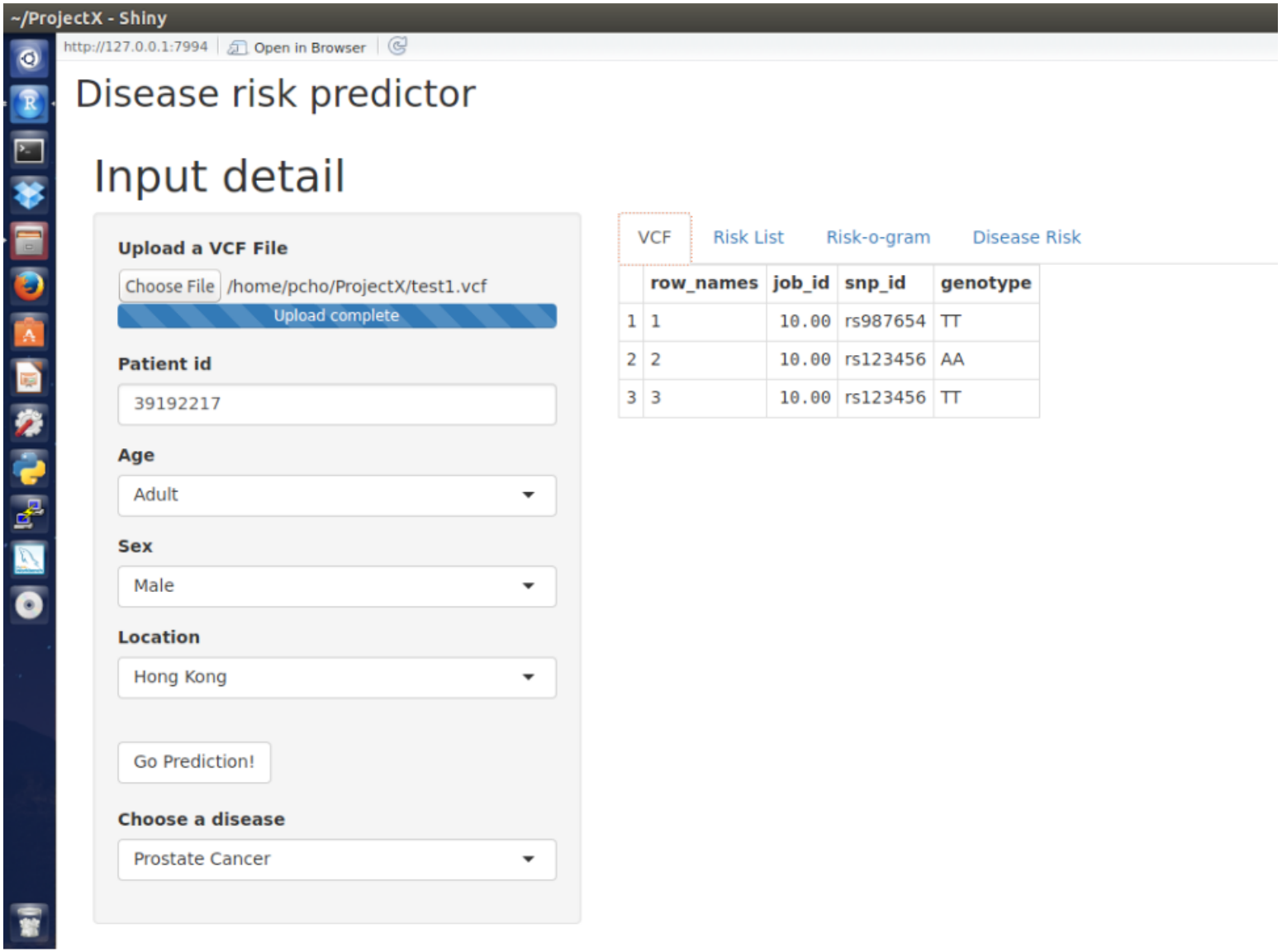
User Interface - screenshot



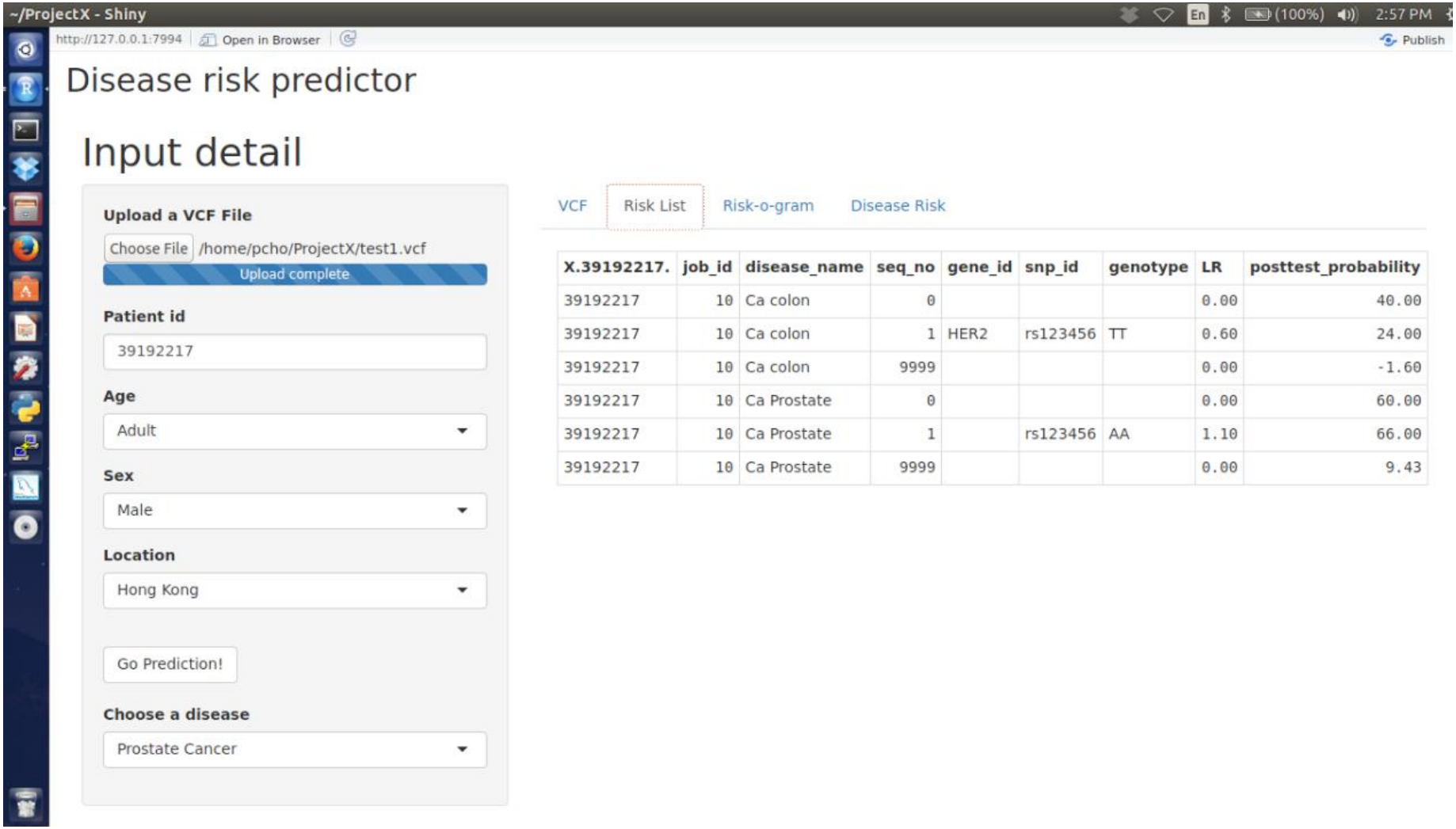
User Interface - screenshot



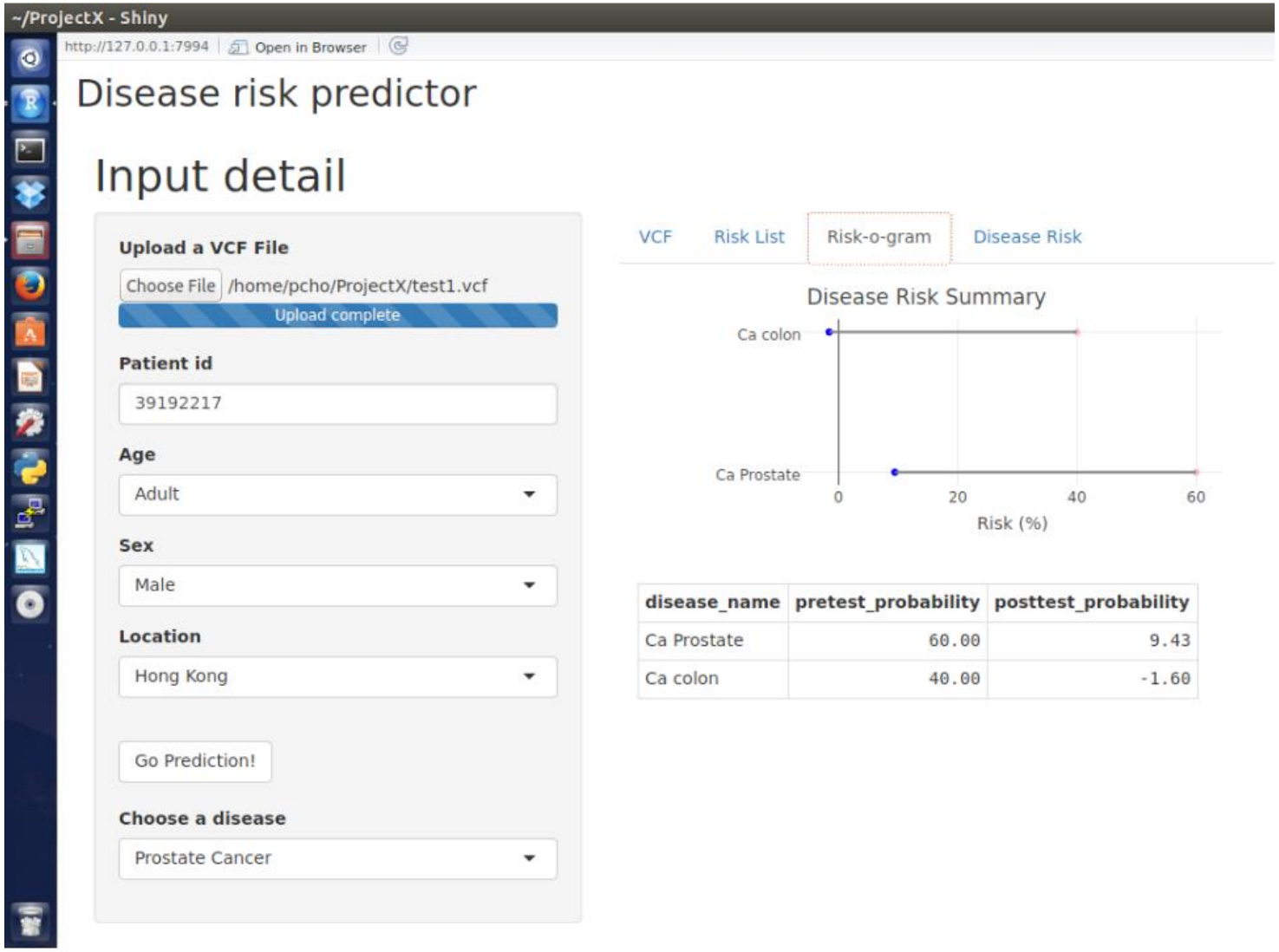
User Interface - screenshot



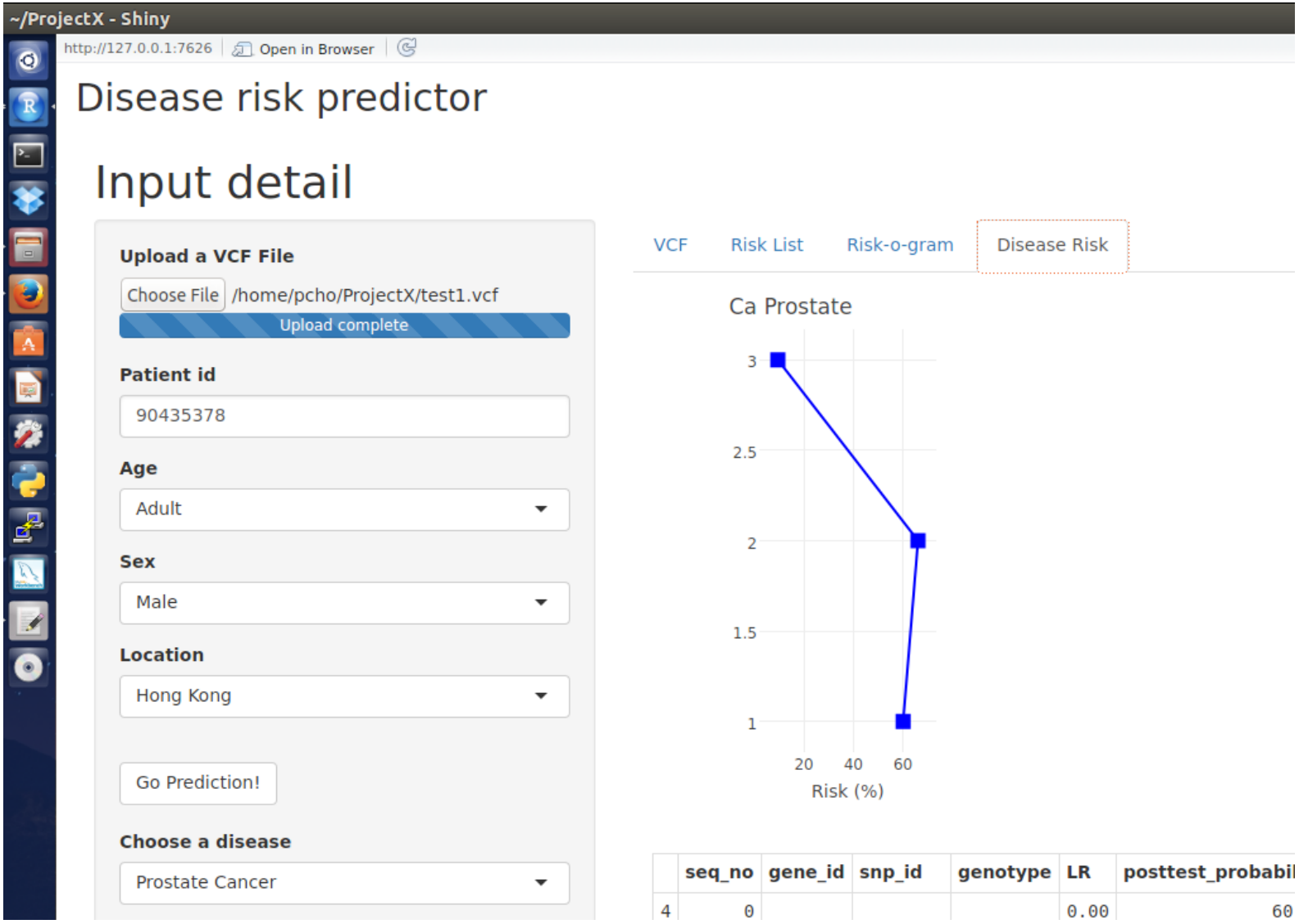
User Interface - screenshot



User Interface - screenshot



User Interface - screenshot

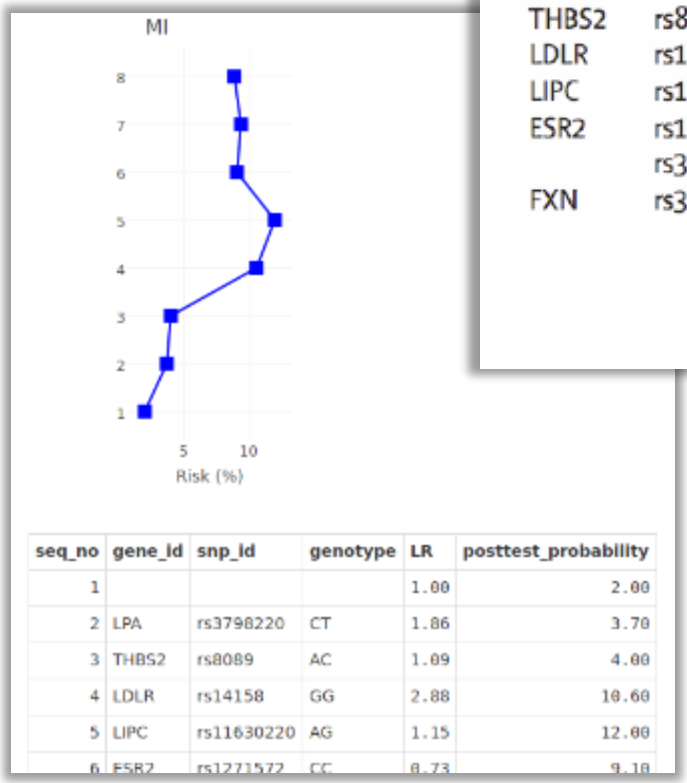


Evaluation - Approach

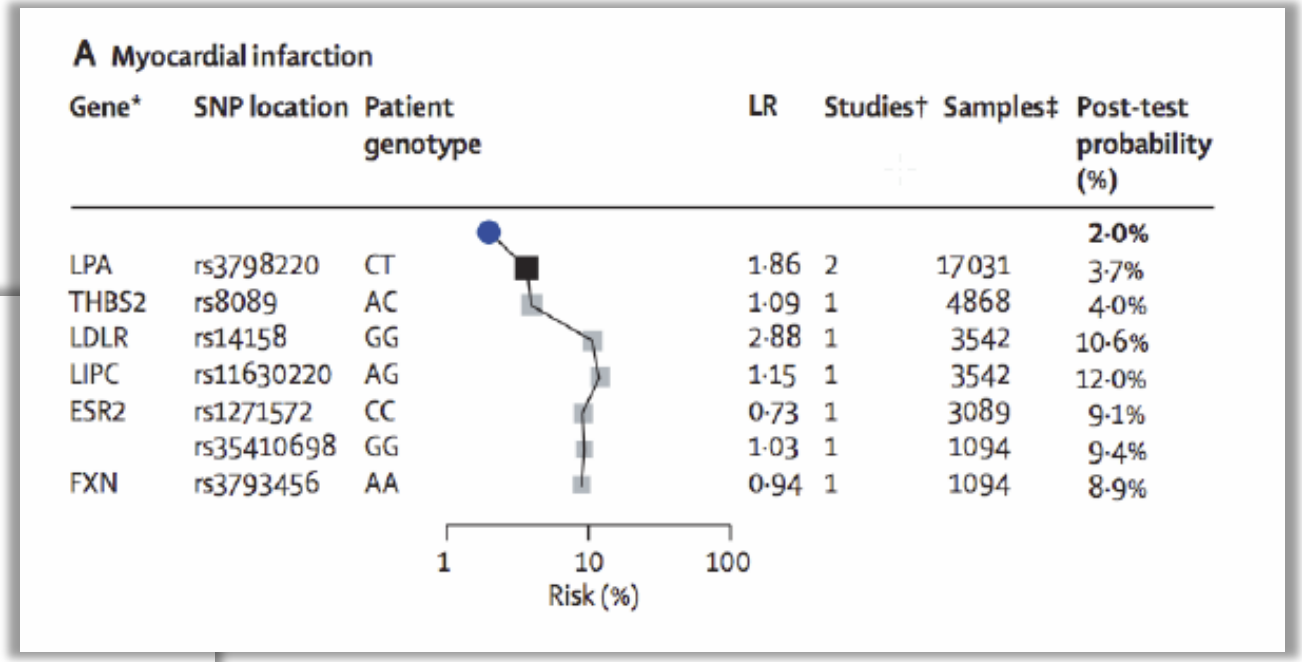
- Programming logic validation
  - Synthetic data
- System validation
  - 23andMe sample

Evaluation – Programming logic validation

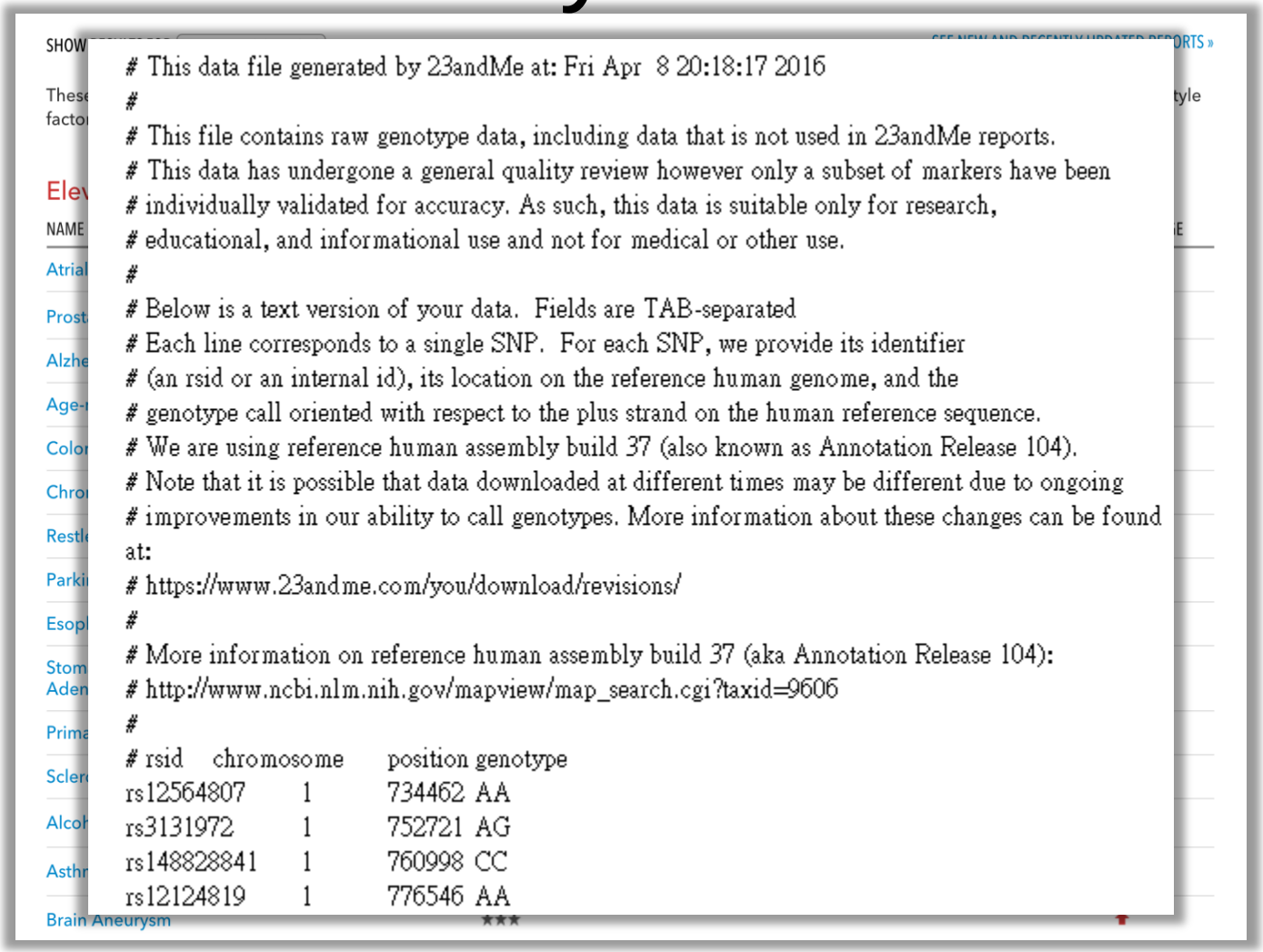
http://stanford.edu/class/gene210/files/lectures/2011/4_Salari_AnalyzingGWASData.pdf



seq_no	gene_id	snp_id	genotype	LR	posttest_probability
1				1.00	2.00
2	LPA	rs3798220	CT	1.86	3.70
3	THBS2	rs8089	AC	1.09	4.00
4	LDLR	rs14158	GG	2.88	10.60
5	LIPC	rs11630220	AG	1.15	12.00
6	ESR2	rs1271572	CC	0.73	9.10



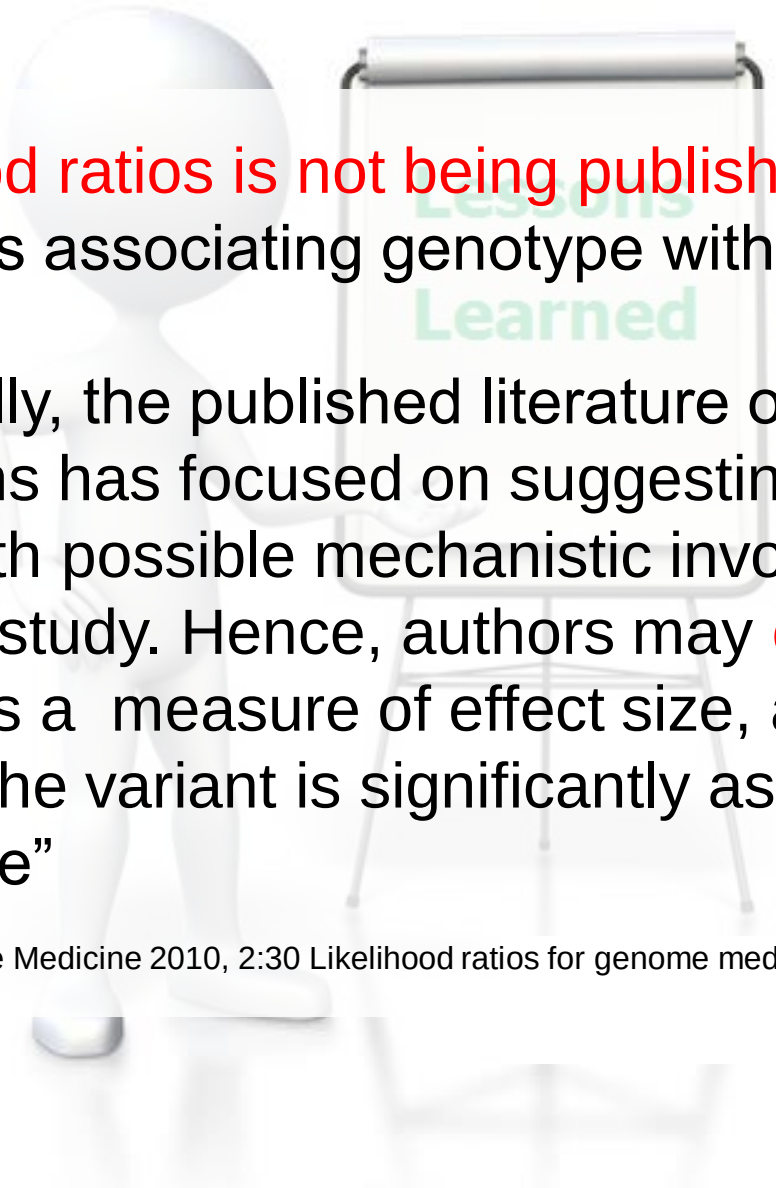
Evaluation – System validation



```
# This data file generated by 23andMe at: Fri Apr  8 20:18:17 2016
#
# This file contains raw genotype data, including data that is not used in 23andMe reports.
# This data has undergone a general quality review however only a subset of markers have been
# individually validated for accuracy. As such, this data is suitable only for research,
# educational, and informational use and not for medical or other use.
#
# Below is a text version of your data.  Fields are TAB-separated
# Each line corresponds to a single SNP.  For each SNP, we provide its identifier
# (an rsid or an internal id), its location on the reference human genome, and the
# genotype call oriented with respect to the plus strand on the human reference sequence.
# We are using reference human assembly build 37 (also known as Annotation Release 104).
# Note that it is possible that data downloaded at different times may be different due to ongoing
# improvements in our ability to call genotypes. More information about these changes can be found
at:
# https://www.23andme.com/you/download/revisions/
#
# More information on reference human assembly build 37 (aka Annotation Release 104):
# http://www.ncbi.nlm.nih.gov/mapview/map_search.cgi?taxid=9606
#
# rsid    chromosome    position genotype
rs12564807      1       734462 AA
rs3131972       1       752721 AG
rs148828841     1       760998 CC
rs12124819      1       776546 AA
```

Limitations and difficulty

- Difficult to collect likelihood ratio
  - Limited resource
- Incomplete SNP data set which likelihood ratio is available
  - Limited supply of Literature
- Manual curate
  - Maintenance difficulty, error prone
- Alternate metric
  - Odd ratio



"…likelihood ratios is not being published in the primary publications associating genotype with disease…"

"Traditionally, the published literature on genetic associations has focused on suggesting interesting variants with possible mechanistic involvement in the disease of study. Hence, authors may only report an odd ratio as a measure of effect size, and a P value to show that the variant is significantly associated with that disease"

Morgan et al. Genome Medicine 2010, 2:30 Likelihood ratios for genome medicine

Finding

"..the objective of "personalized genomics" is not necessarily to predict disease with any certainty, but rather to provide another line of evidence that physicians and other medical practitioners can consider in their interactions with patients…"

Patel et al. Genome Medicine 2013, 5:58. Whole genome sequencing in support of wellness and health maintenance


thank
you!

Appendix

R source code – server.R

server.R

```r
library(shiny)
library(datasets)
library(tidyr)
library(plotly)
library(RMySQL)
library(vcfR)

#connect MySQL PM.db
mydb = dbConnect(MySQL(), user='root', password='sandy', host='localhost', dbname="PM")

shinyServer(function(input, output, session) {
#upload VCF file, ceate trigger event for VCF upload on submit button
  observeEvent(input$file1, {
    inFile <- input$file1
#null if no file was selected
    if (is.null(inFile))
      return(NULL)
#upload file from directory and skip the VCF metadata with header retain
    tempfile <- read.table(inFile$datapath, skip=10, header = FALSE)
#Prepare latest job id a key
    max_job_id <- (as.numeric(dbGetQuery(mydb, "Select max(job_id) from FEI_Job")) + 1)
#Porvide insert SQL on FEI_job
    temp <- paste("INSERT INTO FEI_Job (job_id, patient_id, location_id) VALUES(",max_job_id,",'",input$patientid,"','",input$race,"');")
#Append FEI_vcf with job id, SNP reference number and genotype
#Use vcfR package to import VCF file as vcf class
    inFile1 <- read.vcfR(inFile$datapath)
#Use vcfR method to get genotype from VCF, 0/0= ref homozygous, 0/1= heterozygous, 1/1 = alt homozygous
    DataFile <- extract.gt(inFile1, element ="GT", mask = FALSE, as.numeric = FALSE, return.alleles = TRUE, allele.sep ="/", extract = TRUE)
#Remove "/" in genotype value
    DataFile <- gsub("/","",DataFile)
#Aggregrate job_id, snp_id and genotype
    InsertFEI_vcf <- list(max_job_id,tempfile[,3],DataFile[,1])
#cast as data frame
    InsertFEI_vcf <- as.data.frame(InsertFEI_vcf)
#rename column name of the data frame
    colnames(InsertFEI_vcf) <- c("job_id","snp_id","genotype")
#clean up FEO_Riskogram_Data and FEO_Risk_List_Data table
    dbSendQuery(mydb,"truncate table FEO_Riskogram_Data;")
    dbSendQuery(mydb,"truncate table FEO_Risk_List_Data;")
#Write to MySQL PM database as table FEI_vcf
    dbWriteTable(mydb, "FEI_vcf", InsertFEI_vcf, overwrite =TRUE)
#insert new record with update job_id in FEI_job table to trigger store procedure in MySQL
    dbSendQuery(mydb,temp)

  })
```

R source code – server.R

server.R

```
46
47
48  #Do this when Go prediction button is clicked
49  observeEvent(input$submit, {
50  #Create view for VCF UI display
51      output$viewVCF <- renderTable({dbGetQuery(mydb, "Select * from FEI_vcf")})
52  #Update table read for FEO_Risk_List_Data
53      a <- as.data.frame(fetch((dbSendQuery(mydb, "select * from FEO_Risk_List_Data"))))
54  #Update table read for FEO_Riskogram_Data
55      s <- as.data.frame(fetch((dbSendQuery(mydb, "select * from FEO_Riskogram_Data"))))
56      s <- s[order(s$posttest_probability - s$pretest_probability), ]
57
58  #Filter FEO_Risk_List_Data with chosen disease
59   data <- reactive({
60      read <- a
61      f <- subset(read, disease_name %in% input$disease, select = c(seq_no, gene_id, snp_id, genotype, LR, posttest_probability))
62      return(f)
63  })
64
65  #Update Risk List table
66   output$view <- renderTable({
67      if(is.null(data())){return()}
68      as.data.frame(c(input$patientid,a))
69  },include.rownames=FALSE)
70
71  #update Disease Risk table
72    output$view1 <- renderTable({z <- data()},include.rownames=FALSE)
73  #update Disease risk diagram
74    output$distPlot <- renderPlotly({
75      plot_ly(z <- data(), x = posttest_probability, y = seq_no,
76      mode = "markers+lines", marker = list(color = "blue", symbol ="square", size = 12)
77          )
78
79      layout(
80         title = input$disease,
81         xaxis = list(title = "Risk (%)"),
82         yaxis = list(title =""),
83         margin = list(l = 100),
84         paper_bgcolor = "white"
85      )
86
87      })
```

R source code – server.R

server.R

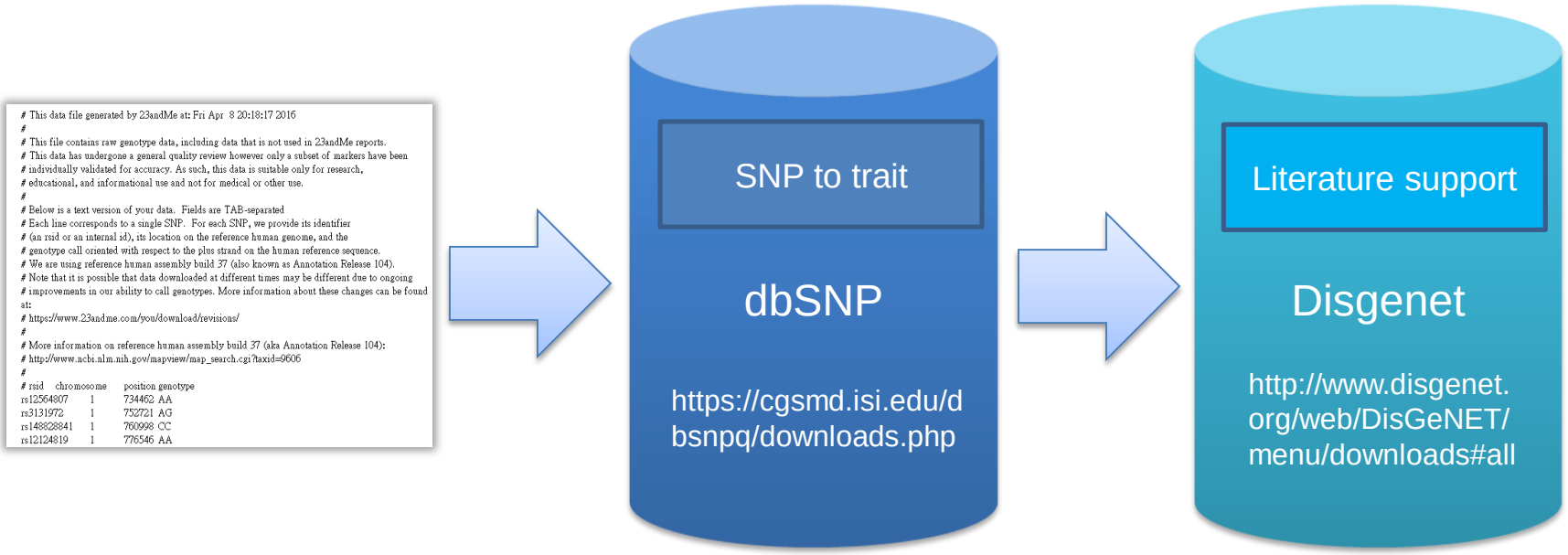
```
88
89  #update risk-g-gram table
90    output$view2 <- renderTable({s[,3:5]},include.rownames=FALSE)
91  #update risk-g-gram graph
92    output$disSPlot <- renderPlotly({
93      gather(s, Cat, value, pretest_probability, posttest_probability) %>%
94      plot_ly(x = value, y = disease_name, mode = "markers",color = Cat, colors = c("pink", "blue"))%>%
95      add_trace(x = value, y = disease_name, mode = "lines", group = disease_name, line = list(color = "gray"))
96      layout(
97          title = "Disease Risk Summary",
98          xaxis = list(title = "Risk (%)"),
99          yaxis = list(title=""),
100         margin = list(l = 150),
101         paper_bgcolor = "white",
102         showlegend = FALSE
103         )
104     })
105
106
107 })
108
109 })
```

R source code – ui.R

ui.R

```
library(shiny)
library(plotly)


shinyUI(fluidPage(

  titlePanel("Disease risk predictor"),
  headerPanel("Input detail"),

   sidebarPanel(
    fileInput('file1', 'Upload a VCF File',
              accept=c('text/csv',
             'text/comma-separated-values,text/plain',
             '.csv/.vcf')),

   textInput("patientid", "Patient id", value = "90435378"),

   selectInput("age","Age",
             choices = c("Adult" = "A",
                         "Child" = "C",
                         "Newborn" = "N")),
    selectInput("sex","Sex",
             choices = c("Male" = "M",
                         "Female" = "F")),
    selectInput("race","Location",
             choices = c("Hong Kong" = "HK",
                         "World" = "World")),
    br(),
    actionButton("submit", "Go Prediction!"),
    br(),br(),
    selectInput("disease", "Choose a disease",
             choices = c ("Prostate Cancer" = "Ca Prostate", "Colon Cancer" = "Ca colon"))

    ),

  mainPanel(
    tabsetPanel(
        tabPanel("VCF",tableOutput("viewVCF")),
        tabPanel("Risk List",br(),column(2,tableOutput("view"))),
        tabPanel("Risk-o-gram",br(),plotlyOutput("disSPlot", "500px","200px"),br(),br(),column(6,tableOutput("view2"))),
        tabPanel("Disease Risk",br(),plotlyOutput("distPlot","250px","400px"),br(),br(),column(6,tableOutput("view1")))
            )
    )

))
```

Evaluation – 23andMe sample




```
# This data file generated by 23andMe at: Fri Apr  8 20:18:17 2016
#
# This file contains raw genotype data, including data that is not used in 23andMe reports.
# This data has undergone a general quality review however only a subset of markers have been
# individually validated for accuracy. As such, this data is suitable only for research,
# educational, and informational use and not for medical or other use.
#
# Below is a text version of your data.  Fields are TAB-separated
# Each line corresponds to a single SNP.  For each SNP, we provide its identifier
# (an rsid or an internal id), its location on the reference human genome, and the
# genotype call oriented with respect to the plus strand on the human reference sequence.
# We are using reference human assembly build 37 (also known as Annotation Release 104).
# Note that it is possible that data downloaded at different times may be different due to ongoing
# improvements in our ability to call genotypes. More information about these changes can be found
at:
# https://www.23andme.com/you/download/revisions/
#
# More information on reference human assembly build 37 (aka Annotation Release 104):
# http://www.ncbi.nlm.nih.gov/mapview/map_search.cgi?taxid=9606
#
# rsid    chromosome    position genotype
rs12564807    1    734462 AA
rs3131972    1    752721 AG
rs148828841    1    760998 CC
rs12124819    1    776546 AA
```

Association type (example)	Count (unique SNP)
Biomarker (rs8008270)	137 (57)
Biomarker, GeneticVariation (rs1058205)	110 (31)
GeneticVariation (rs10498792)	26 (15)

Evaluation - SNPedia

GWAS snp	
PMID	[PMID 23535732]
Trait	Prostate cancer
Title	Identification of 23 new prostate cancer susceptibility loci using the iCOGS custom genotyping array.
Risk Allele	G
P-val	2E-14
Odds Ratio	1.12 [1.08-1.16]

Biomarker (rs8008270)

GWAS	
SNP	rs10498792
PubMedID	[PMID 17903305]
Condition	Prostate cancer
Gene	PKHD1
Risk Allele	
pValue	3.00E-006
OR	NA
95% CI	

GeneticVariation (rs10498792)

GWAS snp	
PMID	[PMID 24919509]
Trait	Prostate-specific antigen levels
Title	A genome-wide association study of serum levels of prostate-specific antigen in the Japanese population.
Risk Allele	
P-val	8E-21
Odds Ratio	.09 [0.067-0.103] unit increase

Biomarker, GeneticVariation (rs1058205)

Evaluation – 23andMe sample

Assumptions

- Only +ve Pre-test Probability and +ve Likelihood ratio are considered, and to be obtained from external data sources
- Prostate Cancer is the first disease use with the research prototype

Interpreting Odds

- "Odds" is often known as the ratio of money that may be won versus the amount of money bet
- In statistics, an **odds** of an event is the ratio of:
  - **The probability that the event WILL occur** *to* **the probability that the event will NOT occur**
    - For example, in 100 births, the probability of a delivery being a boy is 51% and being a girl is 49%
    - The odds of a delivery being a boy is **51/49 = 1.04**
- In simpler term, an odds of an event can be calculated as:
  - **Number of events divided by number of non-events**