ISB bigquery for TCGA: Some Bioconductor strategies

Vince Carey

June 22, 2016

Road map

- Basic overview on cancer genomics cloud approach to level 3 TCGA
- BigQuery intro by doing
- A bladder cancer CDK use case from ISB
- An interactive oncoPrint
- Exercises and commentary

Overview

- TCGA is a collection of omics assay results and clinical characteristics of donors of tumor tissue on a wide variety of cancers
- The public data has so far been a perennial source of logistical challenges for interested bioinformaticians
  - Access is sufficiently complex to warrant several independently developed Bioconductor packages
  - Coordination of data structures and vocabularies consumes significant effort
- NCI Cancer Genomic Cloud pilots: "Democratize" access, federate management and analysis methods

The ISB Cancer Genomic Cloud Pilot: A way in via Google BigQuery

You need an authentication token, 'my_billing' contains secret info

```
getBQ = function ()
{
    library(dplyr)
    library(bigrquery)
    my_billing = <secret>
    src_bigquery("isb-cgc", "tcga_201510_alpha", billing = my_billing)
}
```

Let's try it, dplyr idiom

```
bq = getBQ()
bq
```

```
## src:  bigquery [isb-cgc:tcga_201510_alpha]
## tbls: Annotations, Biospecimen_data, Clinical_data, Copy_Number_segments,
##   DNA_Methylation_betas, miRNA_expression, mRNA_BCGSC_HiSeq_RPKM,
##   mRNA_UNC_HiSeq_RSEM, Protein_RPPA_data, Somatic_Mutation_calls
```

Access now available for all tumor types

```
LUAD_Clin = bq %>% tbl("Clinical_data") %>%
      filter(Study=="LUAD") %>% as.data.frame()
```

```
dim(LUAD_Clin)
```

```
## [1] 522  65
```

- dplyr idiom is not necessary
- `bigrquery query_exec()` will submit BigQuery-compliant SQL

Some variables (lots of NA, blanks)

```
#datatable(LUAD_Clin[,c(1,49,54)], options=list(lengthMenu=c(3,5)))
head(LUAD_Clin[,c(1,49,54)])
```

```
##   ParticipantBarcode pathologic_T primary_therapy_outcome_success
## 1       TCGA-67-3772          T2                             <NA>
## 2       TCGA-67-3774          T2                             <NA>
## 3       TCGA-67-3776          T2                             <NA>
## 4       TCGA-L9-A8F4         T2a                             <NA>
## 5       TCGA-44-5643         T2b                             <NA>
## 6       TCGA-44-5644         T2a     Complete Remission/Response
```

Use case from ISB BigQuery walkthrough

For bladder cancer patients that have mutations in the CDKN2A (cyclin-dependent kinase inhibitor 2A) gene, what types of mutations are they, what is their gender, vital status, and days to death - and for 3 downstream genes (MDM2 (MDM2 proto-oncogene), TP53 (tumor protein p53), CDKN1A (cyclin-dependent kinase inhibitor 1A)), what are the gene expression levels for each patient?

Break it down

- Bladder cancer: Study is BLCA
- Mutation data: filter to CDKN2A and tabulate type
- Clinical data: merge
- Expression data: MDM2, TP53, CDKN1A on these patients

Mutations – NB order of operations can affect timing/timeout

```
mudf = bq %>% tbl("Somatic_Mutation_calls") %>%
  filter(Study=="BLCA") %>% filter(Hugo_Symbol == "CDKN2A") %>%
  select(ParticipantBarcode, Study, Hugo_Symbol,
              Variant_Type, Variant_Classification) %>%
              as.data.frame()
head(mudf,3)
```

```
##   ParticipantBarcode Study Hugo_Symbol Variant_Type Variant_Classification
## 1       TCGA-XF-AAN3  BLCA      CDKN2A          SNP      Missense_Mutation
## 2       TCGA-ZF-A9R4  BLCA      CDKN2A          SNP      Missense_Mutation
## 3       TCGA-ZF-AA4N  BLCA      CDKN2A          SNP            Splice_Site
```

Expression

```
exdf = bq %>% tbl("mRNA_UNC_HiSeq_RSEM") %>%
  filter(Study=="BLCA") %>% filter(HGNC_gene_symbol
          %in% c("MDM2", "TP53", "CDKN2A", "CDKN1A")) %>%
  select(ParticipantBarcode, Study, HGNC_gene_symbol, normalized_count) %>%
              as.data.frame()
```

```
##
Running query:   RUNNING  2.6s
```

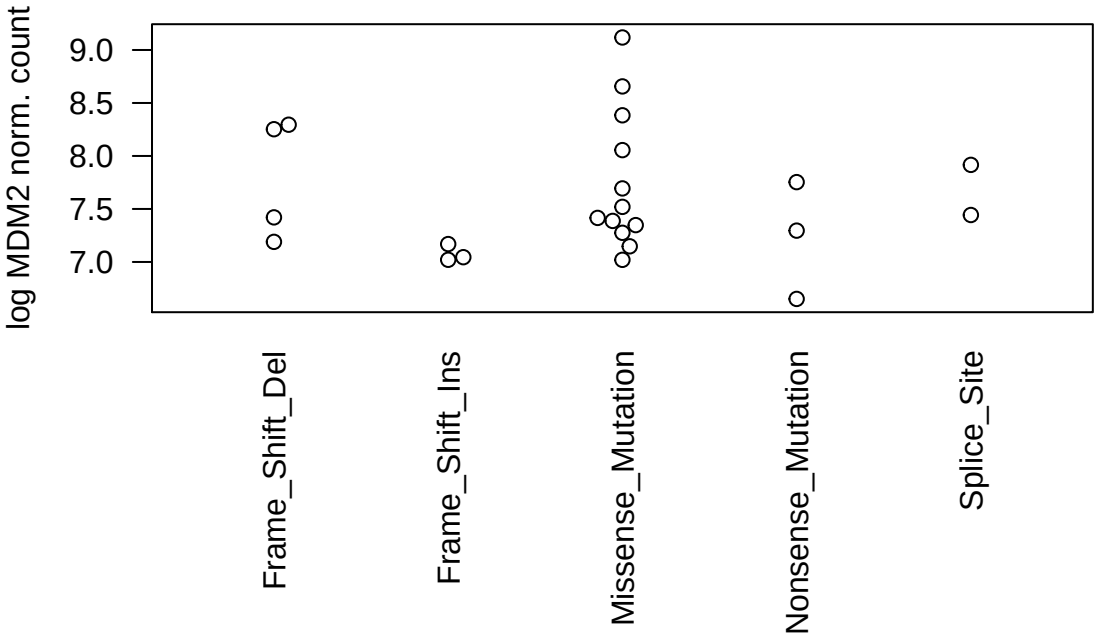
```
## 7.0 gigabytes processed
```

Condense multiple CDKN2A mutations of the same type in an individual

```
o = order(mudf$ParticipantBarcode, mudf$Variant_Classification)
mudf = mudf[o,]
cls = with(mudf, split(Variant_Classification, ParticipantBarcode))
todrop = lapply(cls, duplicated)
mudf = mudf[-which(unlist(todrop)),]
```

Merge mutation and expression data

```
muex = merge(mudf, exdf, by="ParticipantBarcode", all.x=TRUE)
par(mar=c(12,5,3,3),las=2)
with(muex[muex$HG=="MDM2",],
   beeswarm(split(log(normalized_count+1), Variant_Classification),
     ylab="log MDM2 norm. count"))
```



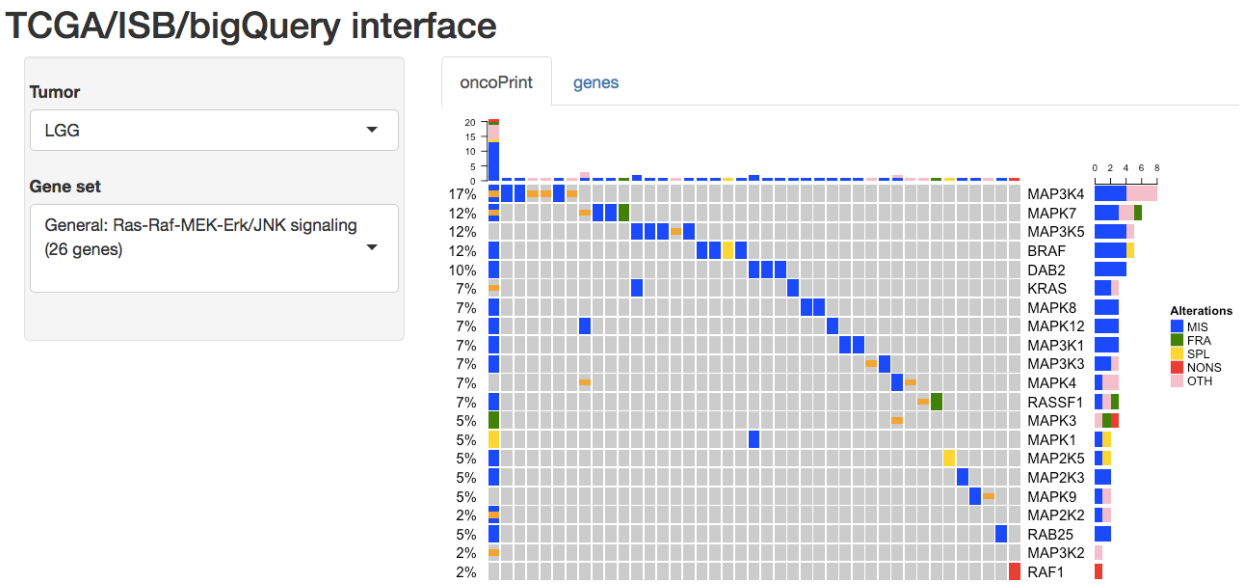
Exercises

- See the walkthrough at ISB
- Write the BigQuery SQL to carry out the merge and use `query_exec` to verify that the R operations agree with the native operations
- Merge the clinical data and test for an effect of CDKN2A mutation class on survival time distribution
- Define and execute a test of the null hypothesis that the mean of (MDM2, TP53, CDKN1A) is constant over CDKN2A mutation classes
- Generalize the computing framework for this test to allow free selection of upstream mutation carriers and downstream expression target patterns for any TCGA tumor family

Interactive oncoprint

To achieve the following display, use

```
library(cgcR)
bq = getBQ() # set your project properly
isbApp(bq) # then pick LGG as the tumor to study
```



Exercises

- Add additional gene sets to the isbApp
- Introduce a systematic approach to labeling mutation classes
- Improve the heatmap tile generation/coding
- Add hoverOver functionality so that relevant information on the sample is produced to help interpret mutation patterns – might take a lot of transformation of code to ggvis or ggplot2/plotly/rbokeh

Comments

- Clinical data curation still important
- Molecular data quality assessment/QC still important
- See `MultiAssayExperiment` package and TCGA archive in S3 bucket
- Additional BigQuery project in ISB CGC: `ccle_201602_alpha` but lacks chemosensitivity profiles