Informe 3: Trabajando con el paquete TCGAbiolinks

Carmen Lebrero Cia

Introducción

Motivación: El Cancer Genome Atlas (TCGA) nos proporciona una enorme colección de data sets que no solo abarcan distintos tipos de cánceres sino que también un gran número de platformas experimentales. Aunque se puede acceder y descargar los datos desde la base de datos, la posibilidad de analizar estos datos descargados directamente con un paquete de R no esta disponible aún.

TCGAbiolinks consiste en tres partes o niveles. Primero, proporcionamos diferentes opciones para consultar y descargar datos relevantes del TGCA de todas las plataformas actuales y su preprocesado para usarlos con paquetes (herramientas) usadas normalmente bioinformática como Bioconductor o CRAN. Segundo, el paquete permite integrar diferentes tipos de datos y puede ser utilizado para diferentes tipos de análisis tratando con todos las plataformas como expresión diferencial, inferencia de redes o análisis de supervivencia, etc, y entonces te permite visualizar los resultados obtenidos. Tercero, hemos añadido un nivel social donde un investigador puede encontrar un interés similar en una comunidad bioinformática y permite a ambos encontrar una validación de resultados en la literatura de pubmed y encontrar un apartado de preguntas y respuestas como soporte de biocondutor, biostars o stackoverflow.

En este documento se describe cócomo buscar, descargar y analizar datos de TCGA usando el paquete TCGAbiolinks.

TCGAbiolinks is able to access The National Cancer Institute (NCI) Genomic Data Commons (GDC) thorough its GDC Application Programming Interface (API) to search, download and prepare relevant data for analysis in R..

Bibliografía / citaciones

If you use TCGAbiolinks, please cite:

- Colaprico, Antonio, et al. "TCGAbiolinks: an R/Bioconductor package for integrative analysis of TCGA data." Nucleic acids research 44.8 (2015): e71-e71.
- Silva, Tiago C., et al. "TCGA Workflow: Analyze cancer genomics and epigenomics data using Bioconductor packages." F1000Research 5 (2016). (https://f1000research.com/articles/5-1542/v2)
- Mounir, Mohamed, et al. "New functionalities in the TCGAbiolinks package for the study and integration of cancer data from GDC and GTEx." PLoS computational biology 15.3 (2019): e1006701. (https://doi.org/10.1371/journal.pcbi.1006701)

Otros links de interés

*Gao, Galen F., et al. "Before and After: Comparison of Legacy and Harmonized TCGA Genomic Data Commons' Data." Cell systems 9.1 (2019): 24-34. (https://doi.org/10.1016/j.cels.2019.06.006)

*TCGA Workflow Analyze cancer genomics and epigenomics data using Bioconductor packages: http://bioconductor.org/packages/TCGAWorkflow/

Instalación

You can install the stable version from Bioconductor. If you are having issues with the stable version, try using the development version.

```
if (!requireNamespace("BiocManager", quietly=TRUE))
    install.packages("BiocManager")
BiocManager::install("TCGAbiolinks")
```

```
## Bioconductor version 3.12 (BiocManager 1.30.10), R 4.0.3 (2020-10-10)

## Installing package(s) 'TCGAbiolinks'

## package 'TCGAbiolinks' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Carmen\AppData\Local\Temp\Rtmpo5lp9o\downloaded_packages

## Old packages: 'biomaRt', 'MatrixGenerics'
```

Paquetes necesarios

```
library(TCGAbiolinks)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(SummarizedExperiment)
```

```
## Loading required package: MatrixGenerics

## Loading required package: matrixStats

##
## Attaching package: 'matrixStats'
```

```
## The following object is masked from 'package:dplyr':
##
##     count

##
## Attaching package: 'MatrixGenerics'

## The following objects are masked from 'package:matrixStats':
##
##     colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
##     colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
##     colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
##     colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
##     colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
##     colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
##     colWeightedMeans, colWeightedMedians, colWeightedSds,
##     colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
##     rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
##     rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
##     rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
##     rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
##     rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
##     rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
##     rowWeightedSds, rowWeightedVars

## Loading required package: GenomicRanges

## Loading required package: stats4

## Loading required package: BiocGenerics

## Loading required package: parallel

##
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB

## The following objects are masked from 'package:dplyr':
##
##     combine, intersect, setdiff, union

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
```

```
## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind, colnames,
##     dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
##     grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
##     union, unique, unsplit, which.max, which.min

## Loading required package: S4Vectors

##
## Attaching package: 'S4Vectors'

## The following objects are masked from 'package:dplyr':
##
##     first, rename

## The following object is masked from 'package:base':
##
##     expand.grid

## Loading required package: IRanges

##
## Attaching package: 'IRanges'

## The following objects are masked from 'package:dplyr':
##
##     collapse, desc, slice

## The following object is masked from 'package:grDevices':
##
##     windows

## Loading required package: GenomeInfoDb

## Loading required package: Biobase

## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.

##
## Attaching package: 'Biobase'

## The following object is masked from 'package:MatrixGenerics':
##
##     rowMedians
```

```
## The following objects are masked from 'package:matrixStats':
##
##     anyMissing, rowMedians
```

```
library(DT)
```

Información de la sesión

```
version
```

```
##                _
## platform       x86_64-w64-mingw32
## arch           x86_64
## os             mingw32
## system         x86_64, mingw32
## status
## major          4
## minor          0.3
## year           2020
## month          10
## day            10
## svn rev        79318
## language       R
## version.string R version 4.0.3 (2020-10-10)
## nickname       Bunny-Wunnies Freak Out
```

```
packageVersion("TCGAbiolinks")
```

```
## [1] '2.17.4'
```

Searching arguments

You can easily search GDC data using the GDCquery function.

Using a summary of filters as used in the TCGA portal, the function works with the following arguments:

Table 1: Tabla de argumentos de las consultas

Argumento	Descripción
project	A list of valid project (see list with TCGAbiolinks:::getGDCprojects()$project_id)]
data.category	A valid project (see list with TCGAbiolinks:::getProjectSummary(project)) For the complete list please check the vignette. List for harmonized database:
workflow.type	
legacy	Search in the legacy repository
access	Filter by access type. Possible values: controlled, open

Argumento	Descripción
platform	Example: CGH- 1x1M_G4447A, IlluminaGA_RNASeqV2, AgilentG4502A_07, IlluminaGA_mRNA_DGE Human1MDuo, HumanMethylation450, HG-CGH-415K_G4124A, IlluminaGA_miRNASeq, HumanHap550, IlluminaHiSeq_miRNASeq, ABI H-miRNA_8x15K, HG-CGH-244A SOLiD_DNASeq, IlluminaDNAMethylation_OMA003_CPI IlluminaGA_DNASeq_automated, IlluminaDNAMethylation_OMA002_CPI, HG-U133_Plus_2, HuEx- 1_0-st-v2 Mixed_DNASeq, H-miRNA_8x15Kv2 IlluminaGA_DNASeq_curated, MDA_RPPA_Core IlluminaHiSeq_TotalRNASeqV2, HT_HG-U133A IlluminaHiSeq_DNASeq_automated diagnostic_images microsat_i, IlluminaHiSeq_RNASeq SOLiD_DNASeq_curated, IlluminaHiSeq_DNASeqC Mixed_DNASeq_curated, IlluminaGA_RNASeq IlluminaGA_DNASeq_Cont_automated, IlluminaGA_DNASeq IlluminaHiSeq_WGBS pathology_reports, IlluminaHiSeq_DNASeq_Cont_automated, Genome_Wide_SNP_6 biotissue_images, Mixed_DNASeq_automated, HumanMethylation27, Mixed_DNASeq_Cont_curated, IlluminaHiSeq_RNASeqV2, Mixed_DNASeq_Cont
file.type	To be used in the legacy database for some platforms, to define which file types to be used.
barcode	A list of barcodes to filter the files to download
experimental.strategy	Filter to experimental strategy. Harmonized: WXS, RNA-Seq, miRNA-Seq, Genotyping Array. Legacy: WXS, RNA-Seq, miRNA-Seq, Genotyping Array, DNA-Seq, Methylation array, Protein expression array, WXS,CGH array, VALIDATION, Gene expression array,WGS, MSI-Mono-Dinucleotide Assay, miRNA expression array, Mixed strategies, AMPLICON, Exon array, Total RNA-Seq, Capillary sequencing, Bisulfite-Seq
sample.type	A sample type to filter the files to download

Lista de los proyectos que podemos encontrar

```
TCGAbiolinks:::getGDCprojects()$project_id
```

```
##  [1] "CMI-MBC"               "CMI-ASC"               "GENIE-MSK"
```

```
##  [4] "TCGA-UCEC"             "TCGA-ACC"              "TCGA-LGG"
##  [7] "TCGA-SARC"             "TCGA-PAAD"             "TCGA-ESCA"
## [10] "TCGA-PRAD"             "GENIE-VICC"            "TCGA-LAML"
## [13] "TCGA-KIRC"             "TCGA-PCPG"             "TCGA-HNSC"
## [16] "BEATAML1.0-COHORT"     "GENIE-JHU"             "TCGA-OV"
## [19] "TCGA-GBM"              "TCGA-UCS"              "WCDT-MCRPC"
## [22] "TCGA-MESO"             "TCGA-TGCT"             "TCGA-KICH"
## [25] "TCGA-READ"             "TCGA-UVM"              "TCGA-THCA"
## [28] "OHSU-CNL"              "GENIE-DFCI"            "GENIE-NKI"
## [31] "VAREPOP-APOLLO"        "GENIE-GRCC"            "FM-AD"
## [34] "GENIE-UHN"             "GENIE-MDA"             "TCGA-LIHC"
## [37] "TCGA-THYM"             "TCGA-CHOL"             "MMRF-COMMPASS"
## [40] "TARGET-ALL-P1"         "ORGANOID-PANCREATIC"   "TCGA-DLBC"
## [43] "TCGA-KIRP"             "TCGA-BLCA"             "CGCI-HTMCP-CC"
## [46] "CPTAC-2"               "TARGET-WT"             "TARGET-ALL-P3"
## [49] "TARGET-CCSK"           "CPTAC-3"               "TARGET-NBL"
## [52] "TARGET-AML"            "CGCI-BLGSP"            "HCMI-CMDC"
## [55] "TCGA-SKCM"             "TARGET-ALL-P2"         "NCICCR-DLBCL"
## [58] "CTSP-DLBCL1"           "TCGA-LUSC"             "TCGA-STAD"
## [61] "BEATAML1.0-CRENOLANIB" "TCGA-LUAD"             "TCGA-COAD"
## [64] "TCGA-CESC"             "TARGET-RT"             "TARGET-OS"
## [67] "TCGA-BRCA"
```

TCGAbiolinks: Downloading and preparing files for analysis

TCGAbiolinks has provided a few functions to download and prepare data from GDC for analysis. This section starts by explaining the different downloads methods and the SummarizedExperiment object, which is the default data structure used in TCGAbiolinks, followed by some examples.

Nosotros queremos obtener todos los datos posibles del proyecto **TCGA-KIRC**. Vamos a ver qué tipo de datos podemos encontrar aquí.

```
TCGAbiolinks:::getProjectSummary("TCGA-KIRC")
```

```
## $case_count
## [1] 537
##
## $data_categories
##   case_count               data_category file_count
## 1        534     Transcriptome Profiling       3065
## 2        339 Simple Nucleotide Variation       3016
## 3        537                  Biospecimen       3257
## 4        537                     Clinical        627
## 5        533             DNA Methylation        899
## 6        534       Copy Number Variation       3393
## 7        535            Sequencing Reads       1998
##
## $file_size
## [1] 2.265568e+13
##
## $file_count
## [1] 16255
```

Observamos 7 tipos de datos distintos. Transcriptoma, SNV, Clinicos, Metilación...

IMPORTANTE!!! En windows los paths tienen un límite de 260 caracteres, por lo que si nuestro path (de working directory) es muy largo a la hora de descargar los archivos y que se cree el correspondiente archivo en nuestro directorio de trabajo encontraremos un error y no se descargará ningún archivo. Lo más sencillo será cambiar el directorio de trabajo para descargar estos datos y luego moverlo todo al working directory.

Descargando datos de expresión proteica (Protein expression)

```
setwd("D:/datos")
query1 <- GDCquery(project = "TCGA-KIRC",
                   data.category = "Protein expression",
                   legacy = TRUE)
```

```
## ------------------------------------

## o GDCquery: Searching in GDC database

## ------------------------------------

## Genome of reference: hg19

## ------------------------------------------

## oo Accessing GDC. This might take a while...

## ------------------------------------------

## ooo Project: TCGA-KIRC

## --------------------

## oo Filtering results

## --------------------

## ----------------

## oo Checking data

## ----------------

## ooo Check if there are duplicated cases

## ooo Check if there results for the query

## -------------------

## o Preparing output

## -------------------
```

```
GDCdownload(query1)
```

```
## Downloading data for project TCGA-KIRC

## Of the 478 files for download 478 already exist.

## All samples have been already downloaded
```

```
ExprProtKIRC <- GDCprepare(query1)
```

```
##   |                                                                      |
```

```
head(ExprProtKIRC)[1:6,1:3]
```

```
##   Composite Element REF TCGA-A3-3316-01A-03-1737-20 TCGA-BP-4970-01A-21-1739-20
## 1       14-3-3_beta-R-V                -0.04705005                 0.001799387
## 2    14-3-3_epsilon-M-C                -0.13894662                 0.014300577
## 3       14-3-3_zeta-R-V                -0.10960432                -0.155629920
## 4            4E-BP1-R-V                 0.06210369                -0.410073616
## 5       4E-BP1_pS65-R-V                -0.05923436                -0.249723679
## 6   4E-BP1_pT37_T46-R-V                 1.26711361                -0.687955634
```

Descargando datos de variación en el número de copias (Copy number variation)

```
setwd("D:/datos")
query2 <- GDCquery(project = "TCGA-KIRC",
                   data.category = "Copy Number Variation", data.type = "Copy Number Segment",
                   legacy = FALSE)
```

```
## -------------------------------------

## o GDCquery: Searching in GDC database

## -------------------------------------

## Genome of reference: hg38

## -------------------------------------------

## oo Accessing GDC. This might take a while...

## -------------------------------------------

## ooo Project: TCGA-KIRC

## --------------------

## oo Filtering results
```

```
## --------------------

## ooo By data.type

## ---------------

## oo Checking data

## ----------------

## ooo Check if there are duplicated cases

## ooo Check if there results for the query

## -------------------

## o Preparing output

## -------------------
```

```
GDCdownload(query2)
```

```
## Downloading data for project TCGA-KIRC

## Of the 1122 files for download 1122 already exist.

## All samples have been already downloaded
```

```
CNVKIRC <- GDCprepare(query2)
```

```
## Reading copy number variation files
```

Descargando datos de Bioespecimen (Bioespecimen)

```
setwd("D:/datos")
query.biospecimen <- GDCquery(project = "TCGA-KIRC",
                  data.category = "Biospecimen",
                  data.type = "Biospecimen Supplement",
                  data.format = "BCR Biotab")
```

```
## ------------------------------------

## o GDCquery: Searching in GDC database

## ------------------------------------

## Genome of reference: hg38
```

```
## ------------------------------------------
## oo Accessing GDC. This might take a while...
## ------------------------------------------
## ooo Project: TCGA-KIRC
## --------------------
## oo Filtering results
## --------------------
## ooo By data.format
## ooo By data.type
## ----------------
## oo Checking data
## ----------------
## ooo Check if there are duplicated cases
## Warning: There are more than one file for the same case. Please verify query results. You can use th
## ooo Check if there results for the query
## -------------------
## o Preparing output
## -------------------
GDCdownload(query.biospecimen)

## Downloading data for project TCGA-KIRC
## Of the 10 files for download 10 already exist.
## All samples have been already downloaded
biospecimen.BCRtab.all <- GDCprepare(query.biospecimen)

##   |                                                                      |
```

Descargando datos de SNV (Simple nucleotide variation)

```
setwd("D:/datos")
query4 <- GDCquery(project = "TCGA-KIRC",
                   data.category = "Simple nucleotide variation", data.type = "Simple nucleotide variati
                   legacy = TRUE)
GDCdownload(query4)
SNVKIRC <- GDCprepare(query4)
```

```
setwd("D:/datos")
query4 <- GDCquery(project = "TCGA-KIRC",
                   data.category = "Simple Nucleotide Variation", data.type = "Masked Somatic Mutation",
                   legacy = FALSE)
GDCdownload(query4)
SNVKIRC <- GDCprepare(query4)
```

Descargando datos de Expresión Génica (Gene Expression)

```
setwd("D:/datos")
query5 <- GDCquery(project = "TCGA-KIRC",
                   data.category = "Gene expression", data.type = "Exon quantification",
                   legacy = TRUE)
GDCdownload(query5)
ExpGenKIRC <- GDCprepare(query5)
```

```
setwd("D:/datos")
query5B <- GDCquery(project = "TCGA-KIRC",
                   data.category = "Gene expression", data.type = "miRNA gene quantification",
                   legacy = TRUE)
GDCdownload(query5B)
ExpGenKIRC2 <- GDCprepare(query5B)
```

```
setwd("D:/datos")
query5C <- GDCquery(project = "TCGA-KIRC",
                   data.category = "Gene expression",
                   data.type = "Gene expression quantification",
                   experimental.strategy = "RNA-Seq",
                   file.type = "results",
                   legacy = TRUE)
GDCdownload(query5C)
ExpGenKIRC3 <- GDCprepare(query5C)
```

Descargando datos de microarrays (Raw microarray data)

```
setwd("D:/datos")
query6 <- GDCquery(project = "TCGA-KIRC",
                   data.category = "Raw microarray data", data.type = "Normalized intensities",
                   legacy = TRUE)
GDCdownload(query6)
MicroarrayKIRC <- GDCprepare(query6)
```

```
setwd("D:/datos")
query6B <- GDCquery(project = "TCGA-KIRC",
                    data.category = "Raw microarray data", data.type = "Raw intensities",
                    legacy = TRUE)
GDCdownload(query6B)
MicroarrayKIRC2 <- GDCprepare(query6B)
```

Descargando datos de metilación de ADN (DNA methylation)

```
setwd("D:/datos")
query7 <- GDCquery(project = "TCGA-KIRC",
                    data.category = "DNA methylation",
                    legacy = TRUE)
GDCdownload(query7)
MetKIRC <- GDCprepare(query7)
```

Descargando datos clínicos (Clinical)

```
setwd("D:/datos")
query8 <- GDCquery(project = "TCGA-KIRC",
                    data.category = "Clinical", data.type = "Clinical data",
                    legacy = TRUE)
GDCdownload(query8)
ClinicKIRC <- GDCprepare_clinic(query8, clinical.info = "patient")
```

```
setwd("D:/datos")
query8B <- GDCquery(project = "TCGA-KIRC",
                    data.category = "Clinical", data.type = "Clinical Supplement",
                    legacy = TRUE)
```

```
## ---------------------------------------

## o GDCquery: Searching in GDC database

## ---------------------------------------

## Genome of reference: hg19

## ---------------------------------------------

## oo Accessing GDC. This might take a while...

## ---------------------------------------------

## ooo Project: TCGA-KIRC

## --------------------

## oo Filtering results
```

```
## --------------------

## ooo By data.type

## ----------------

## oo Checking data

## ----------------

## ooo Check if there are duplicated cases

## ooo Check if there results for the query

## -------------------

## o Preparing output

## -------------------
```

```
GDCdownload(query8B)
```

```
## Downloading data for project TCGA-KIRC

## Of the 537 files for download 537 already exist.

## All samples have been already downloaded
```

```
ClinicKIRC2 <- GDCprepare_clinic(query8B, clinical.info = "patient")
```

```
##   |                                                                      |
```

```
## Warning: `funs()` is deprecated as of dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

Descargando datos de secuenciación en bruto (Raw sequencing data)

```
setwd("D:/datos")
query9 <- GDCquery(project = "TCGA-KIRC",
                   data.category = "Raw sequencing data",  data.type = "Aligned reads",
                   legacy = TRUE)
GDCdownload(query9)
SeqKIRC <- GDCprepare(query9)
```

```
setwd("D:/datos")
query9B <- GDCquery(project = "TCGA-KIRC",
                   data.category = "Sequencing Reads",
                   legacy = FALSE)
GDCdownload(query9B)
SeqKIRC2 <- GDCprepare(query9B)
```

Resumen

La lista de archivos del entorno R al final de esta sesión sería la siguiente:

```
ls()
```

```
## [1] "biospecimen.BCRtab.all" "ClinicKIRC2"            "CNVKIRC"
## [4] "ExprProtKIRC"           "query.biospecimen"      "query1"
## [7] "query2"                 "query8B"
```

Hemos conseguido crear correctamente objetos de R manipulables para los siguientes datos de TCGA-KIRC:

- Bioespecimen
- Datos clínicos
- Copy Number Variation (CNV)
- Expresión Proteica

Hemos conseguido descargar los siguientes datos:

- **Harmonized data**

```
list.dirs("D:/datos/GDCdata/TCGA-KIRC/harmonized", full.names = FALSE, recursive = FALSE)
```

```
## [1] "Biospecimen"                 "Copy_Number_Variation"
## [3] "Simple_Nucleotide_Variation"
```

- **Legacy**

```
list.dirs("D:/datos/GDCdata/TCGA-KIRC/legacy", full.names = FALSE, recursive = FALSE)
```

```
## [1] "Biospecimen"         "Clinical"            "DNA_methylation"
## [4] "Gene_expression"     "Protein_expression"  "Raw_microarray_data"
```

Dentro de Clinical data:

```
list.dirs("D:/datos/GDCdata/TCGA-KIRC/legacy/Clinical", full.names = FALSE, recursive = FALSE)
```

```
## [1] "Clinical_data"       "Clinical_Supplement"
```

Dentro de Gene Expression:

```
list.dirs("D:/datos/GDCdata/TCGA-KIRC/legacy/Gene_expression", full.names = FALSE, recursive = FALSE)
```

```
## [1] "Exon_quantification"            "Gene_expression_quantification"
## [3] "miRNA_gene_quantification"
```

Dentro de Protein Expression:

```
list.dirs("D:/datos/GDCdata/TCGA-KIRC/legacy/Protein_expression", full.names = FALSE, recursive = FALSE)
```

```
## [1] "Protein_expression_quantification"
```

Dentro de Raw Microarray Data:

```
list.dirs("D:/datos/GDCdata/TCGA-KIRC/legacy/Raw_microarray_data", full.names = FALSE, recursive = FALSE)
```

```
## [1] "Normalized_intensities"
```

No hemos conseguido datos de secuenciación.

Problemas que han surgido

Entre los problemas que hemos encontrado y por lo que no se han podido crear los objetos de R a pesar de haber descargado todos los datos correctamente encontramos:

1. En Single nucleotide variation (SNV): "Error in unzip(basename(bin)): argumento zip inválido" y "Error in GDCPrepare(query4): There are samples duplicated. We will not be able to prepare it".
2. En expresión génica: "Error in GDCPrepare(query5): There are samples duplicated. We will not be able to prepare it" y "Error in GDCPrepare(query5B): There are samples duplicated. We will not be able to prepare it".
3. En datos de microarrays (Raw microarray data): "Error in (funcion (classes, fdef, mtable): unable to find an inhereted method for function 'metadata <-' for signature"'function")".
4. En datos de metilación de ADN (DNA methylation): "Error: no se puede ubicar un vector de tamaño 5.8 Mb".
5. Datos de secuenciación en bruto: "Error in unzip(basename(bin)) : argumento zip inválido" y "Error in 0:ceiling(nrow(manifest)/step - 1) : el resultado seria un vector muy largo".