SOFTWARE TOOL ARTICLE

REVISED *TCGA Workflow*: Analyze cancer genomics and epigenomics data using Bioconductor packages [version 2; referees: 1 approved, 2 approved with reservations]

Tiago C. Silva[1,2*], Antonio Colaprico[3,4*], Catharina Olsen[3,4*], Fulvio D'Angelo[5,6*], Gianluca Bontempi[3-5], Michele Ceccarelli[7], Houtan Noushmehr[1,8]

[1]Department of Genetics, Ribeirao Preto Medical School, University of Sao Paulo, Ribeirao Preto, Brazil
[2]Department of Biomedical Sciences, Cedars-Sinai, Los Angeles, CA, USA
[3]Interuniversity Institute of Bioinformatics in Brussels, Brussels, Belgium
[4]Machine Learning Group, ULB, Brussels, Belgium
[5]Department of Science and Technology, University of Sannio, Benevento, Italy
[6]Biogem, Istituto di Ricerche Genetiche Gaetano Salvatore, Avellino, Italy
[7]Qatar Computing Research Institute (QCRI), HBKU, Doha, Qatar
[8]Department of Neurosurgery, Henry Ford Hospital, Detroit, MI, USA

* Equal contributors

v2 **First published:** 29 Jun 2016, **5**:1542 (doi: 10.12688/f1000research.8923.1)
Latest published: 28 Dec 2016, **5**:1542 (doi: 10.12688/f1000research.8923.2)

Abstract

Biotechnological advances in sequencing have led to an explosion of publicly available data via large international consortia such as The Cancer Genome Atlas (TCGA), The Encyclopedia of DNA Elements (ENCODE), and The NIH Roadmap Epigenomics Mapping Consortium (Roadmap). These projects have provided unprecedented opportunities to interrogate the epigenome of cultured cancer cell lines as well as normal and tumor tissues with high genomic resolution. The Bioconductor project offers more than 1,000 open-source software and statistical packages to analyze high-throughput genomic data. However, most packages are designed for specific data types (e.g. expression, epigenetics, genomics) and there is no one comprehensive tool that provides a complete integrative analysis of the resources and data provided by all three public projects. A need to create an integration of these different analyses was recently proposed. In this workflow, we provide a series of biologically focused integrative analyses of different molecular data. We describe how to download, process and prepare TCGA data and by harnessing several key Bioconductor packages, we describe how to extract biologically meaningful genomic and epigenomic data. Using Roadmap and ENCODE data, we provide a work plan to identify biologically relevant functional epigenomic elements associated with cancer. To illustrate our workflow, we analyzed two types of brain tumors: low-grade glioma (LGG) versus high-grade glioma (glioblastoma multiform or GBM). This workflow introduces the following Bioconductor packages: AnnotationHub, ChIPSeeker, ComplexHeatmap, pathview, ELMER, GAIA, MINET, RTCGAToolbox, TCGAbiolinks.

Open Peer Review

Referee Status: ? ✓ ?

	Invited Referees		
	1	**2**	**3**
REVISED **version 2** published 28 Dec 2016			? report
version 1 published 29 Jun 2016	? report	✓ report	

1 **Kyle Ellrott**, Oregon Health & Science University USA

2 **Elena Papaleo**, Danish Cancer Society Research Center Denmark

3 **Charlotte Soneson**, University of Zurich (UZH) Switzerland

Discuss this article

Comments (0)

This article is included in the Bioconductor channel.

Corresponding authors: Tiago C. Silva (tiagochst@usp.br), Houtan Noushmehr (houtan@usp.br)

How to cite this article: Silva TC, Colaprico A, Olsen C *et al.* ***TCGA Workflow*: Analyze cancer genomics and epigenomics data using Bioconductor packages [version 2; referees: 1 approved, 2 approved with reservations]** *F1000Research* 2016, **5**:1542 (doi: 10.12688/f1000research.8923.2)

Grant information: The project was supported by the São Paulo Research Foundation (FAPESP) (2015/02844-7 and 2016/01389-7 to T.C.S. & H.N. and 2015/07925-5 to H.N.), the BridgeIRIS project, funded by INNOVIRIS, Region de Bruxelles Capitale, Brussels, Belgium, and by GENomic profiling of Gastrointestinal Inflammatory-Sensitive CANcers (GENGISCAN), Belgian FNRS PDR (T100914F to G.B.). Funding for open access charge: São Paulo Research Foundation (FAPESP) (2015/07925-5).
The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: No competing interests were disclosed.

First published: 29 Jun 2016, **5**:1542 (doi: 10.12688/f1000research.8923.1)

REVISED **Amendments from Version 1**

In this revised version of our workflow, we made the following major changes:
- The introduction now includes the GDC NCI data portal (https://gdc.nci.nih.gov/).
- The codes to acquire TCGA data were rewritten in order to use the new version of the TCGAbiolinks package. TCGAbiolinks was entirely redesigned to query, download and prepare data from the GDC NCI data portal (https://gdc.nci.nih.gov/) instead of the inactive DCC TCGA data portal (https://tcga-data.nci.nih.gov).

Minor changes includes improvements based on the referee's comments, such as:
- The inclusion of a paragraph to introduce GISTIC data
- The inclusion of a paragraph explaining the differences between the open (TCGA level 3 and 4 data) and controlled data (TCGA level 1 and 2 data) and pointing to sources that might help the user request access to controlled data.
- Hyperlinks and references were corrected.
- Improvements in the text by removing a few redundancies.
- We used data aligned to reference genome hg19 in all steps of the workflow.

See referee reports

Introduction

Cancer is a complex genetic disease spanning multiple molecular events such as point mutations, structural variations, translocations and activation of epigenetic and transcriptional signatures and networks. The effects of these events take place at different spatial and temporal scales with interlayer communications and feedback mechanisms creating a highly complex dynamic system. To gain insight into the biology of tumors most of the research in cancer genomics is aimed at the integration of the observations at multiple molecular scales and the analysis of their interplay. Even if many tumors share similar recurrent genomic events, their relationships with the observed phenotype are often not understood. For example, although we know that the majority of the most aggressive form of brain tumors such as glioma harbor the mutation of a single gene (IDH), the mechanistic explanation of the activation of its characteristic epigenetic and transcriptional signatures are still far to be well characterized. Moreover, network-based strategies have recently emerged as an effective framework for the discovery functional disease drivers that act as main regulators of cancer phenotypes.

Indeed, recent technological developments allowed the deposition of large amounts of genomic and epigenomic data, such as gene expression, DNA methylation, and genomic localization of transcription factors, into freely available public international consortia like The Cancer Genome Atlas (TCGA), The Encyclopedia of DNA Elements (ENCODE), and The NIH Roadmap Epigenomics Mapping Consortium (Roadmap)[1]. An overview of the three consortia is described below:

- **The Cancer Genome Atlas (TCGA):** The TCGA consortium, which is a National Institute of Health (NIH) initiative, makes publicly available molecular and clinical information for more than 30 types of human cancers including exome (variant analysis), single nucleotide polymorphism (SNP), DNA methylation, transcriptome (mRNA), microRNA (miRNA) and proteome. Sample types available at TCGA are: primary solid tumors, recurrent solid tumors, blood derived normal and tumor, metastatic, and solid tissue normal[2].
- **The Encyclopedia of DNA Elements (ENCODE):** Found in 2003 by the National Human Genome Research Institute (NHGRI), the project aims to build a comprehensive list of functional elements that have an active role in the genome, including regulatory elements that govern gene expression. Biosamples includes immortalized cell lines, tissues, primary cells and stem cells[3].
- **The NIH Roadmap Epigenomics Mapping Consortium:** This was launched with the goal of producing a public resource of human epigenomic data in order to analyze biology and disease-oriented research. Roadmap maps DNA methylation, histone modifications, chromatin accessibility, and small RNA transcripts in stem cells and primary *ex vivo* tissues[4,5].

Briefly, these three consortia provide large scale epigenomic data onto a variety of microarrays and next-generation sequencing (NGS) platforms. Each consortium encompasses specific types of biological information on specific type of tissue or cell and when analyzed together, it provides an invaluable opportunity for research laboratories to better understand the developmental progression of normal cells to cancer state at the molecular level and importantly, correlate these phenotypes with tissue of origins.

Although there exists a wealth of possibilities[6] in accessing cancer associated data, Bioconductor represent the most comprehensive set of open source, updated and integrated professional tools for the statistical analysis of large scale genomic data. Thus, we propose our workflow within Bioconductor to describe how to download, process, analyze and integrate cancer data to understand specific cancer-related specific questions. However, there is no tool that solves the issue of integration in a comprehensive sequence and mutation information, epigenomic state and gene expression within the context of gene regulatory networks to identify oncogenic drivers and characterize altered pathways during cancer progression. Therefore, our workflow presents several Bioconductor packages to work with genomic and epigenomics data.

Methods

Access to the data

TCGA data is accessible via the the NCI Genomic Data Commons (GDC) data portal, GDC Legacy Archive and the Broad Institute's GDAC Firehose. The GDC Data Portal provides access to the subset of TCGA data that has been harmonized against GRCh38 (hg38) using GDC Bioinformatics Pipelines which provides methods to the standardization of biospecimen and clinical data, the re-alignment of DNA and RNA sequence data against a common reference genome build GRCh38, and the generation of derived data. Whereas the GDC Legacy Archive provides access to an unmodified copy of data that was previously stored in CGHub[7] and in the TCGA Data Portal hosted by the TCGA Data Coordinating Center (DCC), in which uses as references GRCh37 (hg19) and GRCh36 (hg18).

The previously stored data in CGHub, TCGA Data Portal and Broad Institute's GDAC Firehose, were provided as different levels or tiers that were defined in terms of a specific combination of both processing level (raw, normalized, integrated) and access level (controlled or open access). Level 1 indicated raw and controlled data, level 2 indicated processed and controlled data, level 3 indicated Segmented or Interpreted Data and open access and level 4 indicated region of interest and open access data. While the TCGA data portal provided level 1 to 3 data, Firehose only provides level 3 and 4. An explanation of the different levels can be found at TCGA Wiki. However, the GDC data portal no longer uses this based classification model in levels. Instead a new data model was created, its documentation can be found in GDC documentation. In this new model, data can be open or controlled access. While the GDC open access data does not require authentication or authorization to access it and generally includes high level genomic data that is not individually identifiable, as well as most clinical and all biospecimen data elements, the GDC controlled access data requires dbGaP authorization and eRA Commons authentication and generally includes individually identifiable data such as low level genomic sequencing data, germline variants, SNP6 genotype data, and certain clinical data elements. The process to obtain access to controlled data is found in GDC web site.

Finally, the data provided by GDC data portal and GDC Legacy Archive can be accessed using Bioconductor package TCGAbiolinks, while the data provided by Firehose can be accessed by Bioconductor package RTCGAToolbox.

The next steps describes how one could use TCGAbiolinks & RTCGAToolbox to download clinical, genomics, transcriptomics, epigenomics data, as well as subtype information and GISTIC results (i.e., identified genes targeted by somatic copy-number alterations (SCNAs) that drive cancer growth). All the data used in this workflow has as reference the Genome Reference Consortium human genome (build 37 - hg19).

Downloading data from TCGA data portal. The Bioconductor package TCGAbiolinks[8] has three main functions *GDCquery*, *GDCdownload* and *GDCprepare* that should sequentially be used to respectively search, download and load the data as an R object. *GDCquery* uses GDC API to search the data for a given project and data category and filters the results by samples, sample type, file type and others features if requested by the user. This function returns a object with a summary table with the results found (samples, files and other useful information) and the arguments used in the query. The most important *GDCquery* arguments are *project* which receives a GDC project (TCGA-USC, TCGA-LGG, TARGET-AML, etc), *data.category* which receives a data category (Transcriptome Profiling, Copy Number Variation, DNA methylation, Gene expression, etc), *data.type* which receives a data type (Gene expression quantification, Isoform Expression Quantification, miRNA Expression Quantification, Copy Number Segment, Masked Copy Number Segment, etc), *workflow.type*, which receives a GDC workflow type (HTSeq - Counts, HTSeq - FPKM-UQ, HTSeq - FPKM), *legacy*, which selects to use the legacy database or the harmonized database,*file.type*, which receives a file type for the searches in the legacy database (hg18.seg, hg19.seg, nocnv_,hg18.seg, nocnv_hg19.seg, rsem.genes.results, rsem.genes.normalized_results, etc) and *platform*, which receives a the platform for the searches in the legacy database (HumanMethylation27, Genome_Wide_SNP_6, IlluminaHiSeq_RNASeqV2, etc). A complete list of possible entries for arguments can be found in the TCGAbiolinks vignette. Listing 1 shows an example of this function.

After the search step, the user will be able to download the data using the *GDCdownload* function which can use either the GDC API to download the samples, or the gdc client tools. The downloaded data will be saved in a directory with the project name and a sub-folder with the data.category, for example "TCGA-GBM/DNA_methylation".

Finally, *GDCprepare* transforms the downloaded data into a summarizedExperiment object[9] or a data frame. If *SummarizedExperiment* is set to TRUE, TCGAbiolinks will add to the object sub-type information, which was defined by The Cancer Genome Atlas (TCGA) Research Network reports (the full list of papers can be seen in

TCGAquery_subtype section in TCGAbiolinks vignette), and clinical information. Listing 1 shows how to use these functions to download DNA methylation and gene expression data from the GDC legacy database and 2 shows how to download copy number variation from harmonized data portal. Other examples, that access the harmonized data can be found in the TCGAbiolinks vignette.

```r
library(TCGAbiolinks)

# Obs: The data in the legacy database has been aligned to hg19
query.met.gbm <- GDCquery(project = "TCGA-GBM",
                     legacy = TRUE,
                     data.category = "DNA methylation",
                     platform = "Illumina Human Methylation 450",
                     barcode = c("TCGA-76-4926-01B-01D-1481-05", "TCGA-28-5211-01C-11D-1844-05"))
GDCdownload(query.met.gbm)

met.gbm.450 <- GDCprepare(query = query.met.gbm,
                      save = TRUE,
                      save.filename = "gbmDNAmet450k.rda" ,
                      summarizedExperiment = TRUE)
query.met.lgg <- GDCquery(project = "TCGA-LGG",
                          legacy = TRUE,
                          data.category = "DNA methylation" ,
                          platform = "Illumina Human Methylation 450" ,
                          barcode = c("TCGA-HT-7879-01A-11D-2399-05","TCGA-HT-8113-01A-11D-2399-05"))
GDCdownload(query.met.lgg)
met.lgg.450 <- GDCprepare(query = query.met.lgg,
                          save = TRUE,
                          save.filename = "lggDNAmet450k.rda" ,
                          summarizedExperiment = TRUE)
met.gbm.lgg <- SummarizedExperiment::cbind(met.lgg.450, met.gbm.450)


query.exp.lgg <- GDCquery(project = "TCGA-LGG",
                     legacy = TRUE,
                     data.category = "Gene expression",
                     data.type = "Gene expression quantification",
                     platform = "Illumina HiSeq",
                     file.type = "results",
                     sample.type = "Primary solid Tumor")
GDCdownload(query.exp.lgg)
exp.lgg <- GDCprepare(query = query.exp.lgg, save = TRUE, save.filename = "lggExp.rda")

query.exp.gbm <- GDCquery(project = "TCGA-GBM",
                     legacy = TRUE,
                     data.category = "Gene expression",
                     data.type = "Gene expression quantification",
                     platform = "Illumina HiSeq",
                     file.type = "results",
                        sample.type = "Primary solid Tumor")
GDCdownload(query.exp.gbm)
exp.gbm <- GDCprepare(query = query.exp.gbm, save = TRUE, save.filename = "gbmExp.rda")
exp.gbm.lgg <- SummarizedExperiment::cbind(exp.lgg, exp.gbm)
```

Listing 1. Downloading TCGA DNA methylation and gene expression data from GDC legacy database with TCGAbiolinks

```
library(TCGAbiolinks)
#----------------------------------------------------------------------------
#                Data.category: Copy number variation aligned to hg38
#----------------------------------------------------------------------------
query <- GDCquery(project = "TCGA-ACC",
                  data.category = "Copy Number Variation",
                  data.type = "Copy Number Segment",
                  barcode = c( "TCGA-OR-A5KU-01A-11D-A29H-01","TCGA-OR-A5JK-01A-11D-A29H-01"))
GDCDownload(query)
data <- GDCPrepare(query)

query <- GDCquery("TCGA-ACC",
                  "Copy Number Variation",
                  data.type = "Masked Copy Number Segment",
                  sample.type = c("Primary solid Tumor")) # see the barcodes with
query$ results[[1]]$cases
GDCDownload(query)
data <- GDCPrepare(query)
```

Listing 2. Downloading TCGA copy number variation data from GDC harmonized database with TCGAbiolinks

If a summarizedExperiment object was chosen, the data can be accessed with three different accessors: *assay* for the data information, *rowRanges* to gets the range of values in each row and *colData* to get the sample information (patient, batch, sample type, etc)[9,10]. An example is shown in Listing 3.

```
library(summarizedExperiment)
# get expression matrix
data <- assay(exp.gbm.lgg)

# get genes information
genes.info <- rowRanges(exp.gbm.lgg)

# get sample information
sample.info <- colData(exp.gbm.lgg)
```

Listing 3. summarizedExperiment accessors

The clinical data can be obtained using TCGAbiolinks through two methods. The first one will download only the indexed GDC clinical data which includes diagnoses (vital status, days to death, age at diagnosis, days to last follow up, days to recurrence), treatments (days to treatment, treatment id, therapeutic agents, treatment intent type), demographic (gender, race, ethnicity) and exposures (cigarettes per day, weight, height, alcohol history) information. This indexed clinical data can be obtained using the function *GDCquery_clinical* which can be used as described in Listing 4. This function has two arguments *project* ("TCGA-GBM","TARGETAML", etc) and *type* ("Clinical" or "Biospecimen"). The second method will download the xml files with all clinical data for the patient and retrieve the desired information from it. This will give access to all clinical data available which includes patient (tumor tissue site, histological type, gender, vital status, days to birth, days to last follow up, etc), drug (days to drug therapy start, days to drug therapy end, therapy types, drug name), radiation (days to radiation therapy start, days to radiation therapy end, radiation type, radiation dosage), new tumor event (days to new tumor event after initial treatment, new neoplasm event type, additional pharmaceutical therapy), follow up (primary therapy outcome success, follow up treatment success, vital status, days to last follow up, date of form completion), stage event (pathologic stage, tnm categories), admin (batch number, project code, disease code, Biospecimen Core Resource).

```
# get indexed clinical patient data for GBM samples
gbm_clin <- GDCquery_clinic(project = "TCGA-GBM", type = "Clinical")

# get indexed clinical patient data for LGG samples
lgg_clin <- GDCquery_clinic(project = "TCGA-LGG", type = "Clinical")

# Bind the results, as the columns might not be the same,
# we will will plyr rbind.fill, to have all columns from both files
clinical <- plyr::rbind.fill(gbm_clin,lgg_clin)

# if barcode is not set, it will consider all samples.
# We only set it to make the example faster
query.clin <- GDCquery(project = "TCGA-GBM",
                       data.category = "Clinical",
                       barcode = c("TCGA-08-0516","TCGA-02-0317"))
GDCDownload(query.clin)
clinical.patient <- GDCPrepare_clinic(query,"patient")
clinical.drug <- GDCPrepare_clinic(query,"drug")
clinical.radiation <- GDCPrepare_clinic(query,"radiation")
clinical.admin <- GDCPrepare_clinic(query,"admin")
clinical.followup <- GDCPrepare_clinic(query,"follow_up")
clinical.nte <- GDCPrepare_clinic(query, "new_tumor_event")
clinical.stage <- GDCPrepare_clinic(query,"stage_event")
```

Listing 4. Downloading clinical data with TCGAbiolinks

Mutation information is stored in two types of Mutation Annotation Format (MAF): Protected and Somatic (or Public) MAF files, which are derived from the GDC annotated VCF files. Annotated VCF files often have variants reported on multiple transcripts whereas the protected MAF (*protected.maf) only reports the most critically affected one and the Somatic MAFs (*somatic.maf) are further processed to remove low quality and potential germline variants. To download Somatic MAFs data using TCGAbiolinks, *GDCquery_maf* function is provided (see Listing 5).

```
mutation <- GDCquery_Maf(tumor = "ACC", pipelines = "mutect2")
```

Listing 5. Downloading mutation data with TCGAbiolinks

Finally, the Cancer Genome Atlas (TCGA) Research Network has reported integrated genome-wide studies of various diseases (ACC[11], BRCA[12], COAD[13], GBM[14], HNSC[15], KICH[16], KIRC[17], KIRP[18], LGG[14], LUAD[19], LUSC[20], PRAD[21], READ[13], SKCM[22], STAD[23], THCA[24] and UCEC[23]) which classified them in different subtypes. This classification can be retrieved using the *TCGAquery_subtype* function or by accessing the samples information in the SummarizedExperiment object that created by the *GDCprepare* function.

```
gbm.subtypes <- TCGAquery_subtype(tumor = "gbm")
brca.subtypes <- TCGAquery_subtype(tumor = "brca")
```

Listing 6. Accessing subtype information retrieved from TCGA papers

Downloading data from Broad TCGA GDAC. The Bioconductor package RTCGAToolbox[25] provides access to Firehose Level 3 and 4 data through the function *getFirehoseData*. The following arguments allows users to select the version and tumor type of interest:

- dataset - Tumor to download. A complete list of possibilities can be viewed with *getFirehoseDatasets* function.
- runDate - Stddata run dates. Dates can be viewed with *getFirehoseRunningDates* function.
- gistic2_Date - Analyze run dates. Dates can viewed with *getFirehoseAnalyzeDates* function.

These arguments can be used to select the data type to download: RNAseq_Gene, Clinic, miRNASeq_Gene, ccRNAseq2_Gene_Norm, CNA_SNP, CNV_SNP, CNA_Seq, CNA_CGH, Methylation, Mutation, mRNA_Array, miRNA_Array, and RPPA.

By default, RTCGAToolbox allows users to download up to 500 MB worth of data. To increase the size of the download, users are encouraged to use *fileSizeLimit* argument. An example is found in Listing 7. The *getData* function allow users to access the downloaded data (see lines 22–24 of Listing 7) as a S4Vector object.

```
1 library(RTCGAToolbox)
2
3 # Get the last run dates
4 lastRunDate <- getFirehoseRunningDates()[1]
```

```r
lastAnalyseDate <- getFirehoseAnalyzeDates(1)

# get DNA methylation data, RNAseq2 and clinical data for LGG
lgg.data <- getFirehoseData(dataset = "LGG",
                            gistic2_Date = getFirehoseAnalyzeDates(1), runDate = lastRunDate,
                            Methylation = TRUE, RNAseq2_Gene_Norm = TRUE, Clinic = TRUE,
                            Mutation = T,
                            fileSizeLimit = 10000)

# get DNA methylation data, RNAseq2 and clinical data for GBM
gbm.data <- getFirehoseData(dataset = "GBM",
                            runDate = lastDate, gistic2_Date = getFirehoseAnalyzeDates(1),
                            Methylation = TRUE, Clinic = TRUE, RNAseq2_Gene_Norm = TRUE,
                            fileSizeLimit = 10000)

# To access the data you should use the getData function
# or simply access with @ (for example gbm.data@Clinical)
gbm.mut <- getData(gbm.data,"Mutations")
gbm.clin <- getData(gbm.data,"Clinical")
gbm.gistic <- getData(gbm.data,"GISTIC")
```

Listing 7. Downloading TCGA data files with RTCGAtoolbox

Finally, using RTCGAToolbox the user can retrieve CNV level 4 data, including the amplified or deleted genes identified by GISTIC which rates each segment based on the frequency of occurrence combined with the amplitude of aberration, using a permutation test to assess the statistical significance. Among GISTIC results there are two tables that can be accessed by RTCGAToolbox:

- A gene-level table of copy number values for all samples. The copy number values in the table are in units of (copy number -2), so that no amplification or deletion is 0, genes with amplifications have positive values, and genes with deletions are negative values. The data are converted from marker level to gene level using the extreme method: a gene is assigned the greatest amplification or the least deletion value among the markers it covers.

- A gene-level table of discrete amplification and deletion indicators for all samples. A table value of 0 means no amplification or deletion above the threshold (diploid normal copy). Amplifications are positive numbers: 1 means amplification above the amplification threshold (low-level gain, 1 extra copy); 2 means amplifications larger to the arm level amplifications observed for the sample (high-level amplification, 2 or more extra copies). Deletions are represented by negative table values: -1 represents deletion beyond the threshold (possibly a heterozygous deletion); -2 means deletions greater than the minimum arm-level deletion observed for the sample (possibly a homozygous deletion).

More details about the GISTIC algorithm and its use are described in 26– 28. (see Listing 8).

```r
# Download GISTIC results
gistic <- getFirehoseData("GBM",gistic2_Date ="20141017" )

# get GISTIC results
gistic.allbygene <- gistic@GISTIC@AllByGene
gistic.thresholedbygene <- gistic@GISTIC@ThresholedByGene
```

Listing 8. Using RTCGAToolbox to get the GISTIC results

Genomic analysis

Copy number variations (CNVs) have a critical role in cancer development and progression. A chromosomal segment can be deleted or amplified as a result of genomic rearrangements, such as deletions, duplications, insertions and translocations. CNVs are genomic regions greater than 1 kb with an alteration of copy number between two conditions (e.g., Tumor *versus* Normal).

TCGA collects copy number data and allows the CNV profiling of cancer. Tumor and paired-normal DNA samples were analyzed for CNV detection using microarray and sequencing-based technologies. Level 3 processed data are the aberrant regions along the genome resulting from CNV segmentation, and they are available for all copy number technologies.

In this section, we will show how to analyze CNV level 3 data from TCGA to identify recurrent alterations in cancer genome. We analyzed GBM and LGG segmented CNV from SNP array (Affymetrix Genome-Wide Human SNP Array 6.0).

Pre-Processing Data. The only CNV platform available for both LGG and GBM in TCGA is “Affymetrix Genome-Wide Human SNP Array 6.0”. Using TCGAbiolinks, we queried for CNV SNP6 level 3 data for primary solid tumor samples in the legacy database. Data for selected samples were downloaded and prepared in two separate rse objects (RangedSummarizedExperiment).

```r
###############################
## CNV data pre-processing   ##
###############################
library(TCGAbiolinks)

query.lgg.nocnv <- GDCquery(project = "TCGA-LGG",
                            data.category = "Copy number variation",
                            legacy = TRUE,
                            file.type = "nocnv_hg19.seg",
                            sample.type = c("Primary solid Tumor"))
GDCdownload(query.lgg.nocnv)
lgg.nocnv <- GDCprepare(query.lgg.nocnv, save = TRUE, save.filename = "LGGnocnvhg19.rda")

query.gbm.nocnv <- GDCquery(project = "TCGA-GBM",
                            data.category = "Copy number variation",
                            legacy = TRUE,
                            file.type = "nocnv_hg19.seg",
                            sample.type = c("Primary solid Tumor"))
GDCdownload(query.gbm.nocnv)
gbm.nocnv <- GDCprepare(query.gbm.nocnv, save = TRUE, save.file name = "GBMnocnvhg19.rda")
```

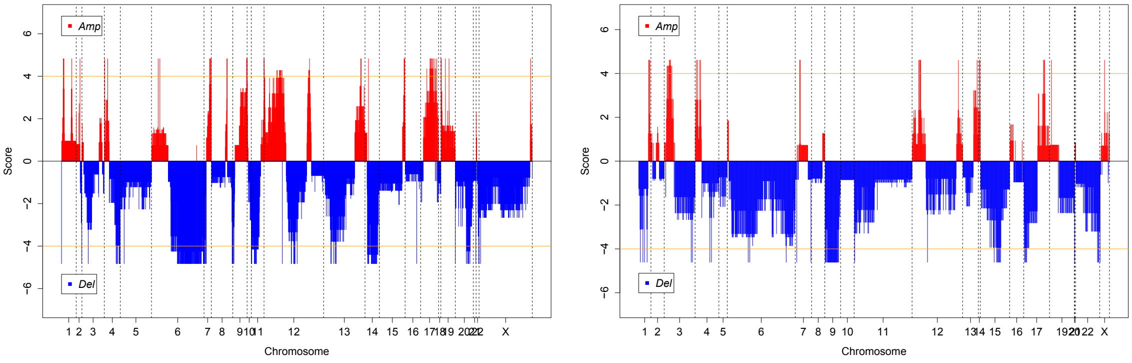
Listing 9. Searching, downloading and preparing CNV data with TCGAbiolinks

Identification of recurrent CNV in cancer. Cancer related CNV have to be present in many of the analyzed genomes. The most significant recurrent CNV were identified using GAIA[29], an iterative procedure where a statistical hypothesis framework is extended to take into account within-sample homogeneity. GAIA is based on a conservative permutation test allowing the estimation of the probability distribution of the contemporary mutations expected for non-driver markers. Segmented data retrieved from TCGA were used to generate a matrix including all needed information about the observed aberrant regions. Furthermore, GAIA requires genomic probes metadata (specific for each CNV technology), that can be downloaded from broadinstitute website.

```r
#############################
## CNV data pre-processing ##
#############################
library(TCGAbiolinks)
library(downloader)
library(readr)
library(gaia)

gaiaCNVplot <- function (calls, cancer=NULL, threshold=0.01)
{
    Calls <- calls[order(calls[,"Region Start [bp]"]),]
    Calls <- Calls[order(Calls[,"Chromosome"]),]
    rownames(Calls) <- NULL
    Chromo <- Calls[,"Chromosome"]
    Gains <- apply(Calls,1,function(x) ifelse(x["Aberration Kind"]==1, x["score"], 0))
    Losses <- apply(Calls, 1,function(x) ifelse(x["Aberration Kind"]==0, x["score"], 0))
    plot(Gains, ylim  = c(-max(Calls [,"score"]+2), max(Calls[,"score"]+2)), type = "h",
         col = "red", xlab = "Chromosome", ylab = "Score",
         #main = paste("Recurrent Copy Number Variations",cancer, sep =" - "),
         xaxt = "n")
    points(-(Losses), type = "h", col = "blue")
    abline(h = 0, cex = 4)
    abline(h = -log10(threshold), col = "orange", cex = 4, main="test")
    abline(h = log10(threshold), col = "orange", cex = 4, main="test")
    uni.chr <- unique(Chromo)
    temp <- rep(0, length(uni.chr))
    for (i in 1:length(uni.chr))  {
         temp[i] <- max(which(uni.chr[i] == Chromo))
    }
    for (i in 1:length(temp)) {
         abline(v = temp[i], col = "black",  lty  = "dashed", )
    }
    nChroms <- length(uni.chr)
    begin <- c()
    for (d in 1:nChroms)  {
         chrom <- sum(Chromo == uni.chr[d])
         begin <- append(begin,  chrom)
    }
    temp2 <- rep(0, nChroms)
    for (i in 1:nChroms) {
         if (i == 1) {
              temp2[1] <- (begin[1] * 0.5)
         }
         else if (i > 1) {
              temp2[i] <- temp[i - 1] + (begin[i] * 0.5)
         }
    }
    uni.chr[uni.chr==23] <- "X"
    uni.chr[uni.chr==24] <- "Y"
    for (i in 1:length(temp)) {
         axis(1, at = temp2[i], labels = uni.chr[i], cex.axis = 1)
    }
    legend(x=1,y=max(Calls[,"score"]+2),y.intersp=0.8, c("Amp"), pch =15, col=c("red"), text.font=3)
    legend(x=1,y=-max(Calls[,"score"]+0.5),y.intersp=0.8, c("Del"), pch =15, col=c("blue"), text.font=3)
}


for(cancer in c("LGG","GBM")){
    message(paste0("Starting", cancer))
    # Prepare CNV matrix
    cnvMatrix <-  get(load(paste0(cancer, "nocnvhg19.rda")))

    # Add label (0 for loss, 1 for gain)
    cnvMatrix <- cbind(cnvMatrix,Label=NA)
    cnvMatrix[cnvMatrix[, "Segment_Mean"] < -0.3, "Label"] <- 0
    cnvMatrix[cnvMatrix[, "Segment_Mean"] > 0.3, "Label"] <- 1
    cnvMatrix <- cnvMatrix [!is.na(cnvMatrix$Label),]

    # Remove "Segment_Mean" and change col.names
    cnvMatrix <- cnvMatrix[,-6]
    colnames(cnvMatrix) <- c("Sample.Name", "Chromosome","Start","End","Num.of.Markers", "Aberration")

    # Substitute Chromosomes "X" and "Y" with "23" and "24"
    xidx <- which(cnvMatrix$Chromosome=="X")
    yidx <- which(cnvMatrix$Chromosome=="Y")
    cnvMatrix[xidx,"Chromosome"] <- 23
    cnvMatrix[yidx,"Chromosome"] <- 24
    cnvMatrix$Chromosome <- sapply(cnvMatrix$Chromosome,as.integer)

    # Recurrent CNV identification with GAIA

    # Retrieve probes meta file from broadinstitute website
    # Recurrent CNV identification with GAIA
    gdac.root <- "ftp://ftp.broadinstitute.org/pub/GISTIC2.0/hg19_support/"
    file <- paste0(gdac.root, "genome.info.6.0_hg19.na31_minus_frequent_nan_probes_sorted_2.1.txt")
    # Retrieve probes meta file from broadinstitute website
    if(!file.exists(basename(file))) download(file,  basename(file))
    markersMatrix <- readr::read_tsv(basename(file), col_names = FALSE, col_types = "ccn", progress = TRUE)
    colnames(markersMatrix) <- c("Probe.Name", "Chromosome", "Start")
    unique(markersMatrix$Chromosome)
    xidx <- which(markersMatrix$Chromosome=="X")
    yidx <- which(markersMatrix$Chromosome=="Y")
    markersMatrix[xidx,"Chromosome"] <- 23
    markersMatrix[yidx,"Chromosome"] <- 24
    markersMatrix$Chromosome <- sapply(markersMatrix$Chromosome,as.integer)
    markerID <- apply(markersMatrix,1,function(x) paste0(x[2],":",x[3]))
    print(table(duplicated(markerID)))
    ## FALSE    TRUE
    ## 1831041     186
    # There are 186 duplicated markers
    print(table(duplicated(markersMatrix$Probe.Name)))
    ## FALSE
    ## 1831227
    #  ... with different names!
    # Removed duplicates
    markersMatrix <- markersMatrix[-which(duplicated(markerID)),]
    # Filter markersMatrix for common CNV
    markerID <- apply(markersMatrix,1,function(x) paste0(x[2],":",x[3]))

    file <- paste0(gdac.root, "CNV.hg19.bypos.111213.txt")
    if(!file.exists(basename(file))) download(file, basename(file))
    commonCNV <- readr::read_tsv(basename(file), progress = TRUE)
    commonID <- apply(commonCNV,1,function(x) paste0(x[2],":",x[3]))
    print(table(commonID %in% markerID))
    print(table(markerID %in% commonID))
    markersMatrix_fil <- markersMatrix[!markerID %in% commonID,]

    markers_obj <- load_markers(as.data.frame(markersMatrix_fil))
    nbsamples <- length(get(paste0("query.",tolower(cancer),".nocnv"))$results[[1]]$cases)
    cnv_obj <- load_cnv(cnvMatrix, markers_obj, nbsamples)
    results <- runGAIA(cnv_obj,
                       markers_obj,
                       output_file_name=paste0("GAIA_",cancer,"_flt.txt"),
           aberrations = -1,
                       chromosomes = -1,
            num_iterations = 10,
            threshold = 0.25)

    # Set q-value threshold
    threshold <- 0.0001

    # Plot the results
    RecCNV <- t(apply(results,1,as.numeric))
    colnames(RecCNV) <- colnames(results)
    RecCNV <- cbind(RecCNV, score=0)
    minval <- format(min(RecCNV[RecCNV[,"q-value"]!=0,"q-value"]),scientific=FALSE)
    minval <- substring(minval,1, nchar(minval)-1)
    RecCNV[RecCNV[,"q-value"]==0,"q-value"] <- as.numeric(minval)
    RecCNV[,"score"] <- sapply(RecCNV[,"q-value"],function(x) -log10(as.numeric(x)))
    RecCNV[RecCNV[,"q-value"]==as.numeric(minval),]

    gaiaCNVplot(RecCNV,cancer,threshold)
    save(results, RecCNV, threshold, file = paste0(cancer,"_CNV_results.rda"))
    message(paste0("Results saved as:", cancer,"_CNV_results.rda"))
}
```

Listing 10. Recurrent CNV identification in cancer with GAIA

Recurrent amplifications and deletions were identified for both LGG (Figure 1a) and GBM (Figure 1b), and represented in chromosomal overview plots by a statistical score ($–log_{10}$ *corrected p-value* for amplifications and log_{10} *corrected p-value* for deletions). Genomic regions identified as significantly altered in copy number (*corrected p-value* $< 10^{-4}$) were then annotated to report amplified and deleted genes potentially related with cancer.



(a) In LGG

(b) In GBM

Figure 1. Recurrent CNV (|score threshold| = 4).

Gene annotation of recurrent CNV. The aberrant recurrent genomic regions in cancer, as identified by GAIA, have to be annotated to verify which genes are significantly amplified or deleted. Using biomaRt we retrieved the genomic ranges of all human genes and we compared them with significant aberrant regions to select full length genes. An example of the result is shown in Table 1.

```
##############################
## Recurrent CNV annotation ##
##############################
library(biomaRt)
library(GenomicRanges)

mart <- useMart(biomart="ensembl", dataset="hsapiens_gene_ensembl")
genes <- getBM(attributes = c("hgnc_symbol", "chromosome_name","start_position","end_position"),
        mart=mart)
genes <- genes[genes[,1]!="" & genes[,2]%in%c(1:22,"X","Y"),]
xidx <- which(genes[,2]=="X")
yidx <- which(genes[,2]=="Y")
genes[xidx, 2] <- 23
genes[yidx, 2] <- 24
genes[,2] <- sapply(genes[,2],as.integer)
genes <- genes[order(genes[,3]),]
genes <- genes[order(genes[,2]),]
colnames(genes) <- c("GeneSymbol","Chr","Start","End")
genes_GR <- makeGRangesFromDataFrame(genes,keep.extra.columns = TRUE)

for(cancer in c("LGG","GBM")){
   load(paste0(cancer,"_CNV_results.rda"))
   sCNV <- RecCNV[RecCNV[,"q-value"]<=threshold,c(1:4,6)]
   sCNV <- sCNV[order(sCNV[,3]),]
   sCNV <- sCNV[order(sCNV[,1]),]
   colnames(sCNV) <- c("Chr","Aberration","Start","End","q-value")
   sCNV_GR <- makeGRangesFromDataFrame(sCNV,keep.extra.columns = TRUE)

   hits <- findOverlaps(genes_GR, sCNV_GR, type="within")
   sCNV_ann <- cbind(sCNV[subjectHits(hits),],genes[queryHits(hits),])
   AberrantRegion <- paste0(sCNV_ann[,1],":",sCNV_ann[,3],"-",sCNV_ann[,4])
   GeneRegion <- paste0(sCNV_ann[,7],":",sCNV_ann[,8],"-",sCNV_ann[,9])
   AmpDel_genes <- cbind(sCNV_ann[,c(6,2,5)],AberrantRegion,GeneRegion)
   AmpDel_genes[AmpDel_genes[,2]==0,2] <- "Del"
   AmpDel_genes[AmpDel_genes[,2]==1,2] <- "Amp"
   rownames(AmpDel_genes) <- NULL

   save(RecCNV, AmpDel_genes, file = paste0(cancer,"_CNV_results.rda"))
}
```

Listing 11. Gene annotation of recurrent CNV

Visualizing multiple genomic alteration events. In order to visualize multiple genomic alteration events we recommend using OncoPrint plot which is provided by Bioconductor package complexHeatmap[30]. The Listing 12 shows how to download mutation data using *GDCquery_maf* (line 4), then we filtered the genes to obtain genes with mutations found among glioma specific pathways (lines 6 – 12). The following steps prepared the data into a matrix to fit *oncoPrint* function. We defined SNPs as blue, insertions as green and deletions as red. The upper barplot indicates the number of genetic mutation per patient, while the right barplot shows the number of genetic mutations per gene. Also, it is possible to add annotations to rows or columns. For the columns, an insertion made at the top will remove the barplot. The final result for adding the annotation to the bottom is highlighted in Figure 2.

Table 1. Chromosome 20 recurrent deleted genes in LGG.

	GeneSymbol	Aberration	q-value	AberrantRegion	GeneRegion
1	EIF4E2P1	Del	5.74967741935484e-05	20:20540891-21005246	20:20659710-20659964
2	LLPHP1	Del	5.74967741935484e-05	20:20540891-21005246	20:20721187-20721879
3	RN7SL607P	Del	5.74967741935484e-05	20:20540891-21005246	20:20738433-20738731
4	MRPS11P1	Del	5.74967741935484e-05	20:20540891-21005246	20:20854121-20854642
5	RPL24P2	Del	5.74967741935484e-05	20:21091497-21220212	20:21114723-21115197

```r
 1 library(ComplexHeatmap)
 2 library(TCGAbiolinks)
 3
 4 LGGmut <- GDCquery_Maf(tumor = "LGG", pipelines = "mutect2")
 5 GBMmut <- GDCquery_Maf(tumor = "GBM", pipelines = "mutect2")
 6
 7 mut <- plyr::rbind.fill(LGGmut,GBMmut)
 8
 9 # Filtering mutations in gliomas
10 EA_pathways <- TCGAbiolinks:::listEA_pathways
11 Glioma_pathways <- EA_pathways[grep("glioma", tolower(EA_pathways$Pathway)),]
12 Glioma_signaling <- Glioma_pathways[Glioma_pathways$Pathway == "Glioma Signaling",]
13 Glioma_signaling_genes <- unlist(strsplit(as.character(Glioma_signaling$Molecules),","))
14
15 mut <- mut[mut$Hugo_Symbol %in% Glioma_signaling_genes,]
16
17 samples <- unique(mut$Tumor_Sample_Barcode)
18 genes <- unique(mut$Hugo_Symbol)
19 mat <- matrix(0,length(genes),length(samples))
20 colnames(mat) <- samples
21 rownames(mat) <- genes
22
23 pb <- txtProgressBar(min = 0, max = nrow(mat), style = 3)
24
25 for (i in 1:nrow(mat)) {
26     curGene <- rownames(mat)[i]
27     setTxtProgressBar(pb, i)
28     for (j in 1:ncol(mat)) {
29         curSample <- colnames(mat)[j]
30
31         if (length(intersect(mut$Tumor_Sample_Barcode, curSample))==1){
32             mat1 <- mut[mut$Tumor_Sample_Barcode == curSample,]
33             if (length(intersect(mat1$Hugo_Symbol, curGene))==1){
34                 mat3 <- mat1[mat1$Hugo_Symbol == curGene,]
35                 mat[curGene,curSample]<- as.character(mat3$Variant_Type)[1]
36             }
37         }
38     }
39 }
```

```
close(pb)

mat[mat==0] <- ""
colnames(mat) <- substr(colnames(mat),1,12)

mat[is.na(mat)] = ""
mat[1:3, 1:3]

alter_fun = list(
    background = function(x, y, w, h) {
        grid.rect(x, y, w-unit(0.5, "mm"), h-unit(0.5, "mm"), gp = gpar(fill = "#CCCCCC", col = NA))
    },
    SNP = function(x, y, w, h) {
        grid.rect(x, y, w-unit(0.5, "mm"), h-unit(0.5, "mm"), gp = gpar(fill = "blue", col = NA))
    },
    DEL = function(x, y, w, h) {
        grid.rect(x, y, w-unit(0.5, "mm"), h-unit(0.5, "mm"), gp = gpar(fill = "red", col = NA))
    },
    INS = function(x, y, w, h) {
        grid.rect(x, y, w-unit(0.5, "mm"), h*0.33, gp = gpar(fill = "#008000", col = NA))
    }
)

col = c("INS" = "#008000", "DEL" = "red", "SNP" = "blue")

clin.gbm <- GDCquery_clinic("TCGA-GBM", "Clinical")
clin.lgg <- GDCquery_clinic("TCGA-LGG", "Clinical")
clinical <- plyr::rbind.fill(clin.lgg,clin.gbm)
annotation <- clinical[match(colnames(mat),clinical$bcr_patient_barcode),
                      c("disease","vital_status","ethnicity")]
annotation <- HeatmapAnnotation(annotation_height = rep(unit(0.3, "cm"),ncol(annotation)),
                              df = annotation,
                              col = list(disease = c("LGG"="green",
                             "GBM"="orange"),
                                         vital_status = c("alive"="blue",
                            "dead"="red",
                            "not reported"="grey"),
           ethnicity = c("hispanic or latino"="purple",
                         "not hispanic or latino"="black",
             "not reported" = "grey")),
                              annotation_legend_param = list(title_gp = gpar(fontsize = 16,
fontface = "bold"),
                                                          labels_gp = gpar(fontsize = 16),
# size labels
                                                          grid_height = unit(8, "mm")))

pdf("LGG_GBM_oncoprint.pdf",width = 20,height = 20)
p <- oncoPrint(mat, get_type = function(x) strsplit(x, ";")[[1]],
          remove_empty_columns = FALSE,
          column_order = NULL, # Do not sort the columns
          alter_fun = alter_fun, col = col,
          row_names_gp = gpar(fontsize = 16), # set size for row names
          pct_gp = gpar(fontsize = 16), # set size for percentage labels
          axis_gp = gpar(fontsize = 16),# size of axis
          column_title = "OncoPrint for TCGA LGG, genes in Glioma signaling",
          column_title_gp = gpar(fontsize = 22),
          pct_digits = 2,
          row_barplot_width = unit(4, "cm"), #size barplot
          bottom_annotation = annotation,
          heatmap_legend_param = list(title = "Mutations", at = c("DEL", "INS", "SNP"),
                                      labels = c("DEL", "INS", "SNP"),
                                      title_gp = gpar(fontsize = 16, fontface = "bold"),
                                      labels_gp = gpar(fontsize = 16), # size labels
                                      grid_height = unit(8, "mm")
          )
)
draw(p, annotation_legend_side = "bottom")
dev.off()
```

Listing 12. Oncoprint

Overview of genomic alterations by circos plot

Genomic alterations in cancer, including CNV and mutations, can be represented in an effective overview plot named circos. We used circlize CRAN package to represent significant CNV (resulting from GAIA analysis) and recurrent

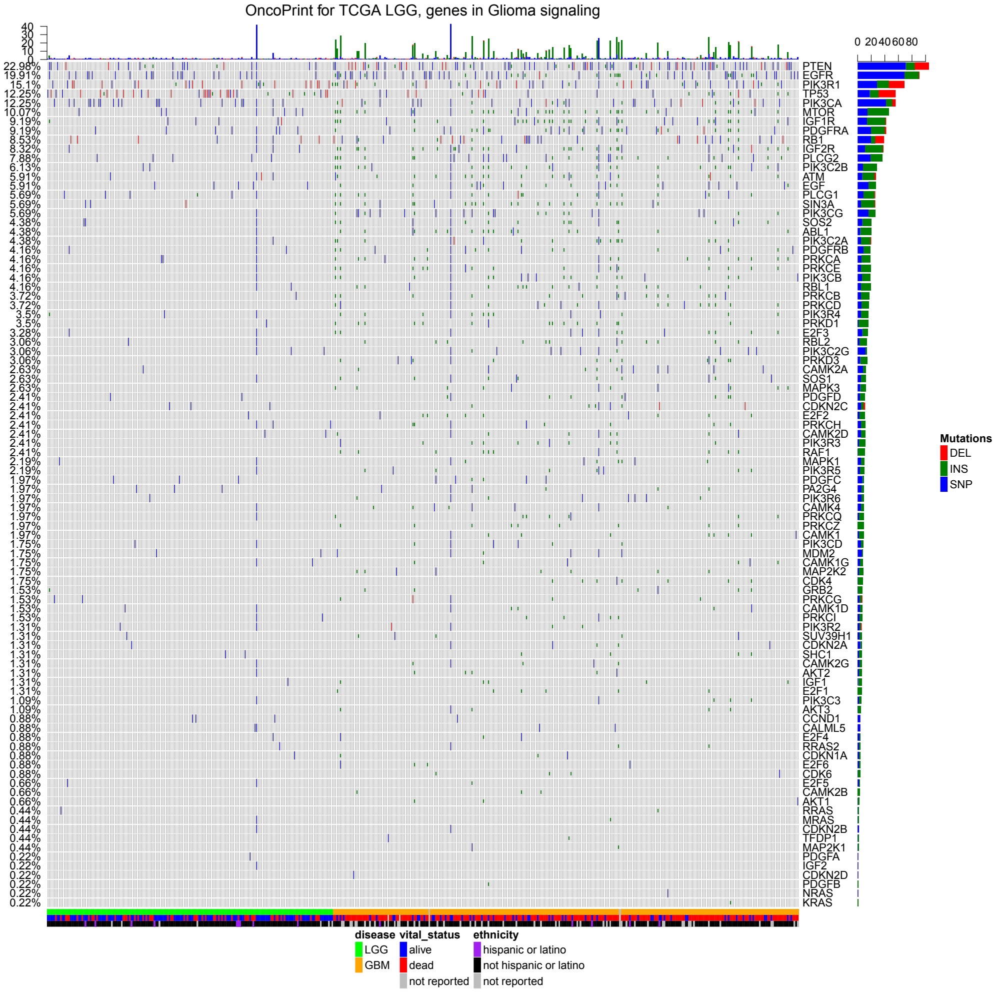


Figure 2. Oncoprint for LGG samples. Blue defines SNP, green defines insertions and red defines deletions. The upper barplot shows the number of these genetic mutations for each patient, while the right barplot shows the number of genetic mutations for each gene. The bottom bar shows the group of each sample.

mutations (selecting curated genetic variations retrieved from TCGA that are identified in at least two tumor samples) in LGG. Circos plot can illustrate molecular alterations genome-wide or only in one or more selected chromosomes. The Figure 3 shows the resulting circos plot for all chromosomes, while the Figure 4 shows the plot for only the chromosome 17.

```
##############################################
## Genomic aberration overview - Circos plot ##
##############################################

# Retrieve curated mutations for selected cancer (e.g. "LGG")
library(TCGAbiolinks)
mut <- GDCquery_Maf(tumor = "LGG")
# Select only potentially damaging mutations
mut <- mut[mut$Variant_Classification %in% c("Missense_Mutation", "Nonsense_Mutation", "Nonstop_Mutation",
"Frame_Shift_Del", "Frame_Shift_Ins"),]
# Select recurrent mutations (identified in at least two samples)
mut.id <- paste0(mut$Chromosome, ":", mut$Start_position, "-", mut$End_position, "|",
mut$Reference_Allele)
mut <- cbind(mut.id, mut)
numSamples <- table(mut.id)
s.mut <- names(which(numSamples>=2))
# Prepare selected mutations data for circos plot
s.mut <- mut[mut$mut.id %in% s.mut,]
s.mut <- s.mut[,c("Chromosome", "Start_position", "End_position", "Variant_Classification",
"Hugo_Symbol")]
s.mut <- unique(s.mut)
s.mut[,1] <- as.character(s.mut[,1])
s.mut[,4] <- as.character(s.mut[,4])
s.mut[,5] <- as.character(s.mut[,5])
typeNames <- unique(s.mut[,4])
type <- c(4:1)
names(type) <- typeNames[1:4]
Type <- type[s.mut[ ,4]]
s.mut <- cbind(s.mut,Type)
s.mut <- s.mut[,c(1:3,6,4,5)]

# Load recurrent CNV data for selected cancer (e.g. "LGG")
load("LGG_CNV_results.rda")
# Prepare selected sample CNV data for circos plot
s.cnv <- as.data.frame(RecCNV[RecCNV[ ,"q−value"]<=10^−4,c(1:4,6)])
s.cnv <- s.cnv[,c(1,3,4,2)]
xidx <- which(s.cnv$Chromosome==23)
yidx <- which(s.cnv$Chromosome==24)
s.cnv[xidx, "Chromosome"] <- "X"
s.cnv[yidx, "Chromosome"] <- "Y"
Chromosome <- sapply(s.cnv[,1],function(x) paste0("chr",x))
s.cnv <- cbind(Chromosome, s.cnv[,-1])
s.cnv[,1] <- as.character(s.cnv[,1])
s.cnv[,4] <- as.character(s.cnv[,4])
s.cnv <- cbind(s.cnv,CNV=1)
colnames(s.cnv) <- c("Chromosome", "Start_position", "End_position", "Aberration_Kind", "CNV")

# Draw genomic circos plot
library(circlize)
pdf("CircosPlot.pdf", width=15, height=15)
par(mar=c(1, 1, 1, 1), cex=1)
circos.initializeWithIdeogram()
# Add CNV results
colors <- c("forestgreen", "firebrick")
names(colors) <- c(0,1)
circos.genomicTrackPlotRegion(s.cnv, ylim = c(0,1.2),
                               panel.fun = function(region, value, ...) {
                                   circos.genomicRect(region, value, ytop.column = 2, ybottom = 0,
                                                   col = colors[value[[1]]],
                                                   border="white")
```

```
                                  cell.xlim = get.cell.meta.data("cell.xlim")
                                  circos.lines(cell.xlim, c(0,0), lty = 2, col = "#00000040")
                              })
# Add mutation results
colors <- c("blue","green","red","gold")
names(colors)  <- typeNames[1:4]
circos.genomicTrackPlotRegion(s.mut, ylim = c(1.2,4.2),
                              panel.fun = function(region, value, ...) {
                                  circos.genomicPoints(region, value, cex = 0.8, pch = 16, col = colors[value[[2]]], ...)
                              })

circos.clear()

legend(-0.2, 0.2, bty="n", y.intersp=1, c("Amp","Del"), pch=15, col=c("firebrick","forestgreen"), title="CNVs", text.font=3, cex=1.2, title.adj=0)
legend(-0.2, 0, bty="n", y.intersp=1, names(colors), pch=16, col=colors, title="Mutations", text.font=3, cex=1.2, title.adj=0)
dev.off()

# Draw single chromosome circos plot (e.g. "Chr 17")
pdf("CircosPlotChr17.pdf",width=18,height=13)
par(mar=c(1,1,1,1),cex=1.5)
circos.par("start.degree" = 90, canvas.xlim = c(0,1), canvas.ylim = c(0,1),
           gap.degree = 270, cell.padding = c(0,0,0,0), track.margin = c(0.005,0.005))
circos.initializeWithIdeogram(chromosome.index = "chr17")
circos.par(cell.padding = c(0,0,0,0))
# Add CNV results
colors <- c("forestgreen","firebrick")
names(colors) <- c(0,1)
circos.genomicTrackPlotRegion(s.cnv, ylim = c(0,1.2),
                              panel.fun = function(region, value, ...) {
                                  circos.genomicRect(region, value, ytop.column = 2, ybottom = 0,
                                                  col = colors[value[[1]]],
                                                  border="white")
                                  cell.xlim = get.cell.meta.data("cell.xlim")
                                  circos.lines(cell.xlim, c(0,0), lty = 2, col = "#00000040")
                              })

# Add mutation results representing single genes
genes.mut <- paste0(s.mut$Hugo_Symbol,"-",s.mut$Type)
s.mutt <- cbind(s.mut,genes.mut)
n.mut <- table(genes.mut)
idx <- !duplicated(s.mutt$genes.mut)
s.mutt <- s.mutt[idx,]
s.mutt <- cbind(s.mutt,num=n.mut[s.mutt$genes.mut])
genes.num <- paste0(s.mutt$Hugo_Symbol," (",s.mutt$num.Freq,")")
s.mutt <- cbind(s.mutt[,-c(6:8)],genes.num)
s.mutt[,6] <- as.character(s.mutt[,6])
s.mutt[,4] <- s.mutt[,4]/2
s.mutt$num.Freq <- NULL
colors <- c("blue","green","red","gold")
names(colors)  <- typeNames[1:4]
circos.genomicTrackPlotRegion(s.mutt, ylim = c(0.3,2.2), track.height = 0.05,
                              panel.fun = function(region, value, ...) {
                                  circos.genomicPoints(region, value, cex = 0.8, pch = 16, col = colors[value[[2]]], ...)
                              })

circos.genomicTrackPlotRegion(s.mutt, ylim = c(0, 1), track.height = 0.1, bg.border = NA)
i_track = get.cell.meta.data("track.index")

circos.genomicTrackPlotRegion(s.mutt, ylim = c(0, 1),
                              panel.fun = function(region, value, ...) {
                                  circos.genomicText(region, value,
                                                     y = 1,
                                                     labels.column = 3,
                                                     col = colors[value[[2]]],
                                                     facing ="clockwise",adj = c(1,0.5),
                                                     posTransform = posTransform.text,
cex = 1.5, niceFacing = T)
                              }, track.height = 0.1, bg.border = NA)

circos.genomicPosTransformLines(s.mutt,
                                posTransform = function(region, value)
                                    posTransform.text(region,
                                                      y = 1,
                                                      labels = value[[3]],
                                                      cex = 0.8, track.index = i_track+1),
                                direction = "inside", track.index = i_track)

circos.clear()

legend(0.25, 0.2, bty="n", y.intersp=1, c("Amp","Del"), pch=15, col=c("firebrick",
"forestgreen"), title="CNVs", text.font=3, cex=1.3,  title.adj=0)
legend(0, 0.2, bty="n", y.intersp=1, names(colors), pch=16, col=colors, title="Mutations",
text.font=3, cex=1.3, title.adj=0)
dev.off()
```

Listing 13. **Genomic aberration overview by circos plot**

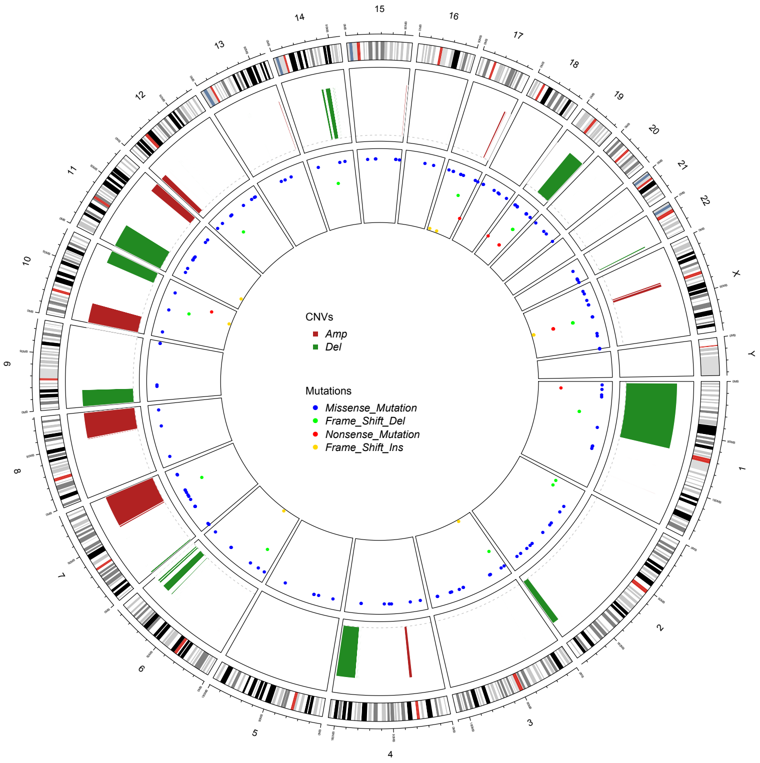


Figure 3. **Circos plot of recurrent CNVs and mutations in LGG.**

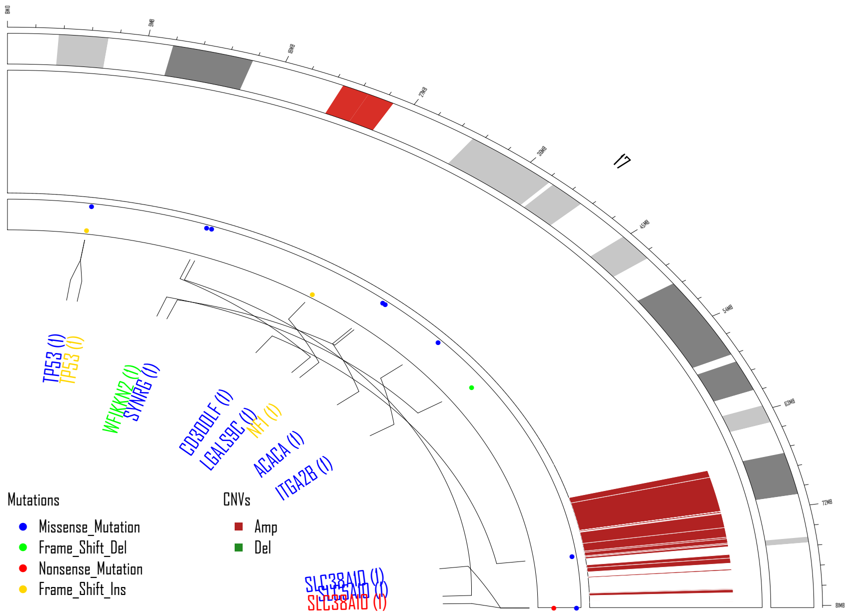


Figure 4. Circos plot of chromosome 17 recurrent CNVs and mutations in LGG.

Transcriptomic analysis

Pre-Processing Data. The LGG and GBM data used for following transcriptomic analysis were downloaded using TCGAbiolinks. We downloaded only primary solid tumor (TP) samples, which resulted in 516 LGG samples and 156 GBM samples, then prepared it in two separate rse objects (*RangedSummarizedExperiment*) saving them as an R object with a file name including both the name of the cancer and the name of the platform used for gene expression data (see Listing 14).

```
library(TCGAbiolinks)

query <- GDCquery(project = "TCGA-GBM",
                  data.category = "Gene expression",
                  data.type = "Gene expression quantification",
                  platform = "Illumina HiSeq",
                  file.type = "results",
                  sample.type = c("Primary solid Tumor"),
                  legacy = TRUE)
GDCdownload(query)
gbm.exp <- GDCprepare(query,
                       save = TRUE,
                       summarizedExperiment = TRUE,
                       save.filename = "GBMIllumina_HiSeq.rda")

query <- GDCquery(project = "TCGA-LGG",
                  data.category = "Gene expression",
                  data.type = "Gene expression quantification",
                  platform = "Illumina HiSeq",
                  file.type = "results",
                  sample.type = c("Primary solid Tumor"),
                  legacy = TRUE)
GDCdownload(query)
lgg.exp <- GDCprepare(query, save = TRUE,
                      summarizedExperiment = TRUE,
                      save.filename = "LGGIllumina_HiSeq.rda")
```

Listing 14. Searching, downloading and preparing RNA-seq data with TCGAbiolinks

To pre-process the data, first, we searched for possible outliers using the *TCGAanalyze_Preprocessing* function, which performs an Array Array Intensity correlation AAIC (lines 14–17 and 26–29 of Listing 15). The Array-array intensity correlation plot (AAIC) is a re-adaptation of the function correlationPlot from the R/Bioconductor affyQCReport package[31] that shows an heat map of the array-array Spearman/Pearson rank correlation coefficients. The arrays are ordered using the phenotypic data (if available) in order to place arrays with similar samples adjacent to each other. Self-self correlations are on the diagonal and by definition have a correlation coefficient of 1.0. Data from similar tissues or treatments will tend to have higher coefficients. This plot is useful for detecting outliers, failed hybridizations, or mistracked samples.

In our example, we defined a square symmetric matrix of Pearson correlation among all samples in each cancer type (LGG or GBM). This matrix found 0 samples with low correlation (cor.cut = 0.6) that can be identified as possible outliers.

Second, using the *TCGAanalyze_Normalization* function, which encompasses the functions of the EDASeq package, we normalized mRNA transcripts.

The *TCGAanalyze_Normalization* performs normalization using the following functions from EDASeq: newSeqExpressionSet, withinLaneNormalization, betweenLaneNormalization, counts. The within-lane normalization procedures to adjust for GC-content effect (or other gene-level effects) on read counts[32]. The between-lane normalization procedures to adjust for distributional differences between lanes (e.g., sequencing depth): global-scaling and full-quantile normalization[33].

```
 1 library(TCGAbiolinks)
 2
 3 rse <- get(load("LGGIllumina_HiSeq.rda"))
 4 dataClin_LGG <- GDCquery_clinic("TCGA-LGG", "Clinical")
 5
 6 dataPrep_LGG <- TCGAanalyze_Preprocessing(object = rse,
 7                                      cor.cut = 0.6,
 8                                      datatype = "raw_count",
 9                                      filename = "LGG_IlluminaHiSeq_RNASeqV2.png")
10
11 rse <- get(load("GBMIllumina_HiSeq.rda"))
12 dataClin_GBM <- GDCquery_clinic("TCGA-GBM", "Clinical")
13
14 dataPrep_GBM <- TCGAanalyze_Preprocessing(object = rse,
15                                          cor.cut = 0.6,
16                                          datatype = "raw_count",
17                                          filename = "GBM_IlluminaHiSeq_RNASeqV2.png")
18
19 dataNorm <- TCGAanalyze_Normalization(tabDF = cbind(dataPrep_LGG, dataPrep_GBM),
20                                      geneInfo = geneInfo,
21                                      method = "gcContent")
22
23 dataFilt <- TCGAanalyze_Filtering(tabDF = dataNorm,
24                                  method = "quantile",
25                                  qnt.cut = 0.25)
26
27 save(dataFilt, file = paste0("LGG_GBM_Norm_IlluminaHiSeq.rda"))
28
29 dataFiltLGG <- subset(dataFilt, select = substr(colnames(dataFilt),1,12) %in% dataClin_LGG$bcr_
        patient_barcode)
30 dataFiltGBM <- subset(dataFilt, select = substr(colnames(dataFilt),1,12) %in% dataClin_GBM$bcr_
        patient_barcode)
31
32 dataDEGs <- TCGAanalyze_DEA(mat1 = dataFiltLGG,
33                            mat2 = dataFiltGBM,
34                            Cond1type = "LGG",
35                            Cond2type = "GBM",
                            fdr.cut = 0.01,
                            logFC.cut = 1,
                            method = "glmLRT")
```

Listing 15. Normalizing mRNA transcripts and differential expression analysis with TCGAbiolinks

Using *TCGAanalyze_DEA*, we identified 2,901 differentially expressed genes (DEG) (log fold change >=1 and FDR < 1%) between 515 LGG and 155 GBM samples.

EA: enrichment analysis. In order to understand the underlying biological process of DEGs we performed an enrichment analysis using *TCGAanalyze_EA_complete* function (see Listing 16).

```
ansEA <- TCGAanalyze_EAcomplete(TFname="DEA genes LGG Vs GBM", RegulonList = rownames(dataDEGs))

TCGAvisualize_EAbarplot(tf = rownames(ansEA$ResBP),
                        GOBPTab = ansEA$ResBP, GOCCTab = ansEA$ResCC,
                        GOMFTab = ansEA$ResMF, PathTab = ansEA$ResPat,
                        nRGTab = rownames(dataDEGs),
                        nBar = 20)
```

Listing 16. Enrichment analysis

TCGAanalyze_EAbarplot outputs a bar chart as shown in Figure 5 with the number of genes for the main categories of three ontologies (i.e., GO:biological process, GO:cellular component, and GO:molecular function).

The Figure 5 shows canonical pathways significantly over-represented (enriched) by the DEGs. The most statistically significant canonical pathways identified in the DEGs are ranked according to their p-value corrected FDR (-Log10) (colored bars) and the ratio of list genes found in each pathway over the total number of genes in that pathway (ratio, red line).

PEA: Pathways enrichment analysis. To verify if the genes found have a specific role in a pathway, the Bioconductor package pathview[34] can be used. Listing 17 shows an example how to use it. It can receive, for example, a named vector of gene with the expression level, the pathway.id which can be found in KEGG database, the species ('hsa' for Homo sapiens) and the limits for the gene expression (see Figure 6).

```
GenelistComplete <- rownames(assay(rse,1))

# DEGs TopTable
dataDEGsFiltLevel <- TCGAanalyze_LevelTab(dataDEGs,"LGG","GBM",
                                          dataFilt[,colnames(dataFiltLGG)],
                                          dataFilt[,colnames(dataFiltGBM)])

dataDEGsFiltLevel$GeneID <- 0

# Converting Gene symbol to geneID
library(clusterProfiler)
eg = as.data.frame(bitr(dataDEGsFiltLevel$mRNA,
                        fromType="SYMBOL",
                        toType="ENTREZID",
                        OrgDb="org.Hs.eg.db"))
eg <- eg[!duplicated(eg$SYMBOL),]

dataDEGsFiltLevel <- dataDEGsFiltLevel[dataDEGsFiltLevel$mRNA %in% eg$SYMBOL,]

dataDEGsFiltLevel <- dataDEGsFiltLevel[order(dataDEGsFiltLevel$mRNA,decreasing=FALSE),]
eg <- eg[order(eg$SYMBOL,decreasing=FALSE),]

# table(eg$SYMBOL == dataDEGsFiltLevel$mRNA) should be TRUE
all(eg$SYMBOL == dataDEGsFiltLevel$mRNA)
dataDEGsFiltLevel$GeneID <- eg$ENTREZID

dataDEGsFiltLevel_sub <- subset(dataDEGsFiltLevel, select = c("GeneID", "logFC"))
```

```
genelistDEGs <- as.numeric(dataDEGsFiltLevel_sub$logFC)
names(genelistDEGs) <- dataDEGsFiltLevel_sub$GeneID

require("pathview")
# pathway.id: hsa05214 is the glioma pathway
# limit: sets the limit for gene expression legend and color
hsa05214 <- pathview(gene.data  = genelistDEGs,
                     pathway.id = "hsa05214",
                     species    = "hsa",
                     limit      = list(gene=as.integer(max(abs(genelistDEGs)))))
```

Listing 17. Pathways enrichment analysis with pathview package

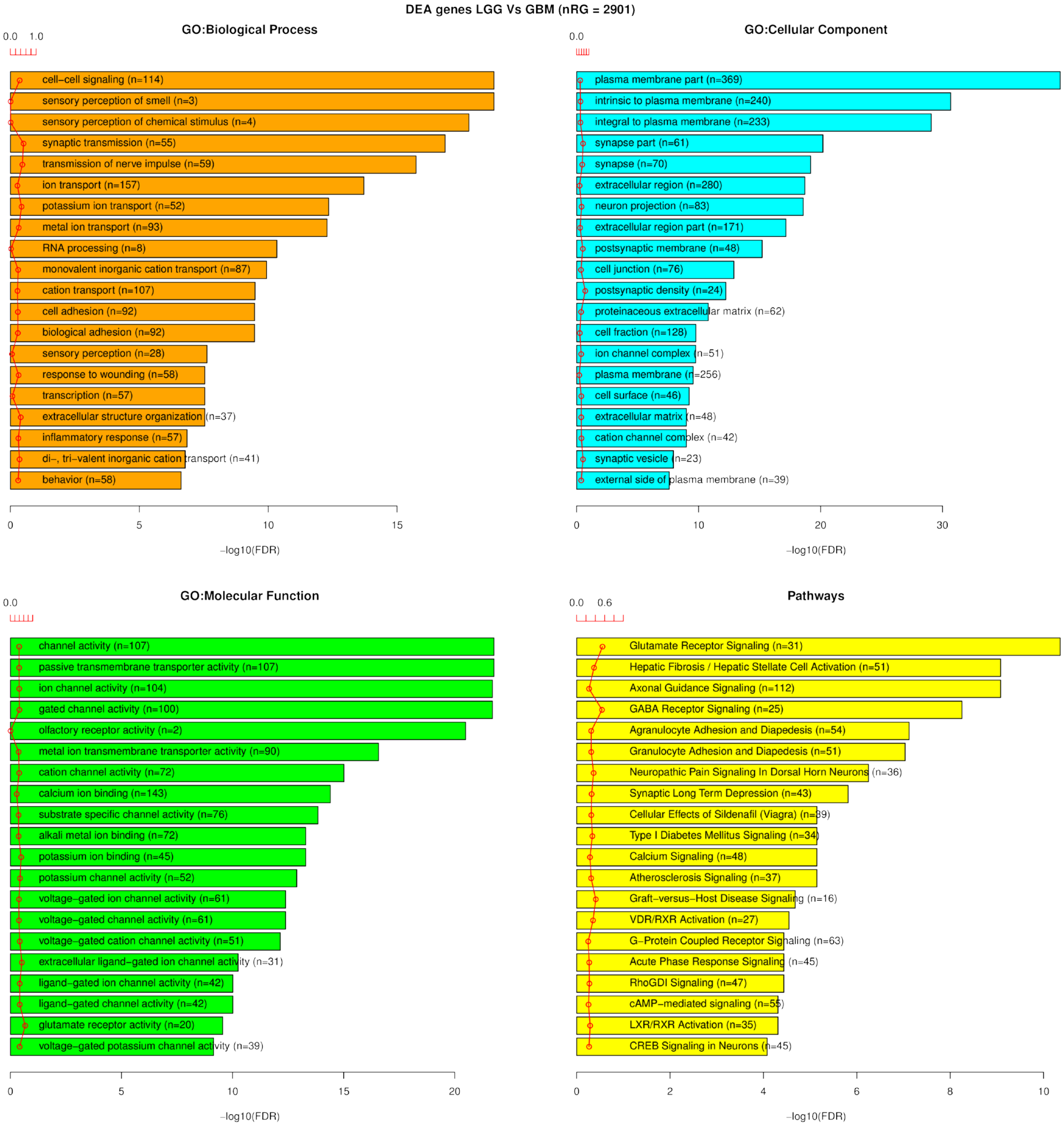


Figure 5. The plot shows canonical pathways significantly over-represented (enriched) by the DEGs (differentially expressed genes) with the number of genes for the main categories of three ontologies (GO:biological process, GO:cellular component, and GO:molecular function, respectively). The most statistically significant canonical pathways identified in DEGs are listed according to their p-value corrected FDR (-Log10) (colored bars) and the ratio of listed genes found in each pathway over the total number of genes in that pathway (ratio, red line).

The red genes are up-regulated and the green genes are down-regulated in the LGG samples compared to GBM.

Inference of gene regulatory networks. Starting with the set of differentially expressed genes, we infer gene regulatory networks using the following state-of-the art inference algorithms: ARACNE[35], CLR[36], MRNET[37] and C3NET[38]. These methods are based on mutual inference and use different heuristics to infer the edges in the network. These methods have been made available via Bioconductor/CRAN packages (MINET[39] and c3net[38], respectively).

Many gene regulatory interactions have been experimentally validated and published. These 'known' interactions can be accessed using different tools and databases such as BioGrid[40] or GeneMANIA[41]. However, this knowledge is far from complete and in most cases only contains a small subset of the real interactome. The quality assessment of the inferred networks can be carried out by comparing the inferred interactions to those that have been validated. This comparison results in a confusion matrix as presented in Table 2.

Table 2. Confusion matrix, comparing inferred network to network of validated interactions.

	validated	not validated/non-existing
inferred	TP	FP
not inferred	FN	TN

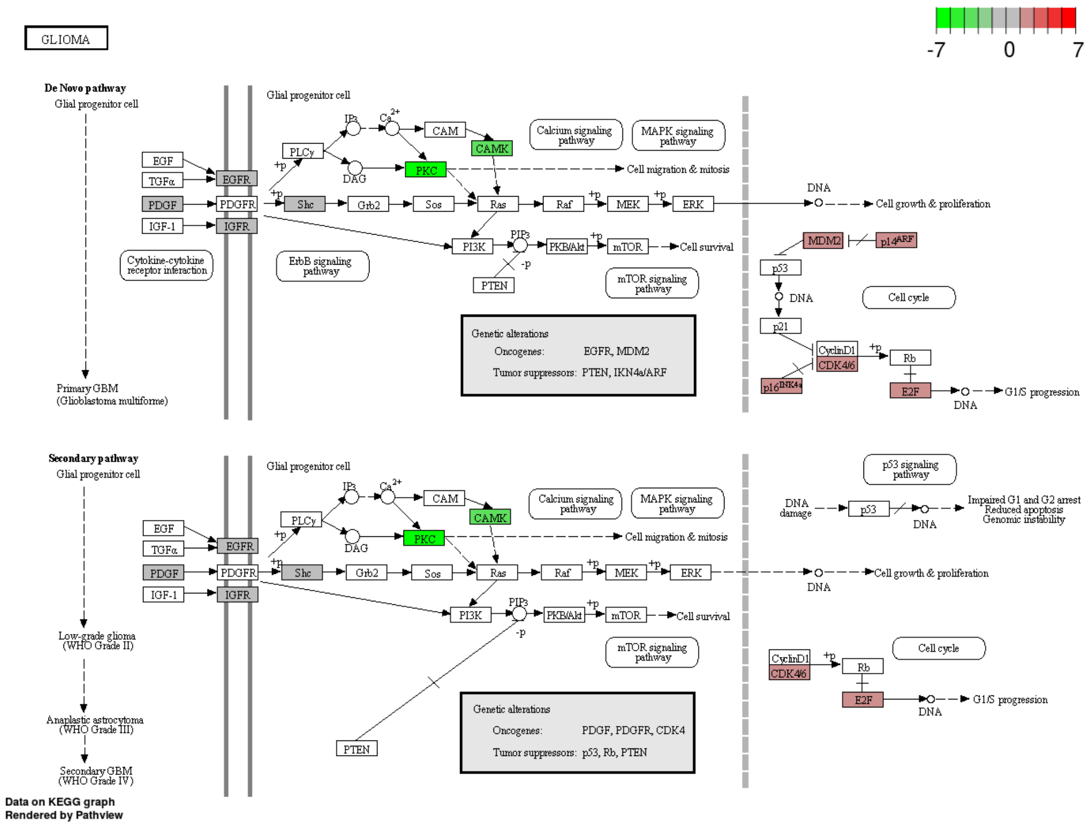


Figure 6. Pathways enrichment analysis : glioma pathway. Red defines genes that are up-regulated and green defines genes that are down-regulated.

Different quality measures can then be computed such as the false positive rate

$$fpr = \frac{FP}{FP + TN},$$

the true positive rate (also called recall)

$$tpr = \frac{TP}{TP + FN}$$

and the precision

$$p = \frac{TP}{TP + FP}.$$

The performance of an algorithm can then be summarized using ROC (false positive rate versus true positive rate) or PR (precision versus recall) curves.

A weakness of this type of comparison is that an edge that is not present in the set of known interactions can either mean that an experimental validation has been tried and did not show any regulatory mechanism or (more likely) has not yet been attempted.

In the following, we ran the nce on i) the 2,901 differentially expressed genes identified in Section “Transcriptomic analysis”.

Retrieving known interactions

As previously stated, different sources for protein-protein interactions are available (e.g. I2D, BioGrid database). In this example, we obtained a set of known interactions from the BioGrid database, but the users can chose their preferred database.

```r
get.adjacency.biogrid <- function(tmp.biogrid, names.genes = NULL){

  if(is.null(names.genes)){
    names.genes <- sort(union(unique(tmp.biogrid[,"Official.Symbol.Interactor.A"]),
    unique(tmp.biogrid[,"Official.Symbol.Interactor.B"])))
    ind <- seq(1,nrow(tmp.biogrid))
  }else{
    ind.A <- which(tmp.biogrid[,"Official.Symbol.Interactor.A"]%in%names.genes)
    ind.B <- which(tmp.biogrid[,"Official.Symbol.Interactor.B"]%in%names.genes)

    ind <- intersect(ind.A, ind.B)
  }

  mat.biogrid <- matrix(0, nrow=length(names.genes), ncol=length(names.genes), dimnames=list(names.genes, names.genes))

  for(i in ind){
    mat.biogrid[tmp.biogrid[i,"Official.Symbol.Interactor.A"], tmp.biogrid[i,"Official.Symbol.Interactor.B"]] <- mat.biogrid[tmp.biogrid[i,"Official.Symbol.Interactor.B"], tmp.biogrid[i,"Official.Symbol.Interactor.A"]] <- 1
  }
  diag(mat.biogrid) <- 0

  return(mat.biogrid)
}
```

There are 3,941 unique interactions between the 2,901 differentially expressed genes.

Using differentially expressed genes from TCGAbiolinks workflow

We start this analysis by inferring two gene regulatory networks (the corresponding number of edges are presented in Table 3) for the GBM data set and one gene set for the LGG data.

Table 3. **Number of edges in the inferred gene regulatory networks; first two lines: networks inferred using 2,901 differentially expressed genes.**

gene set	inference algorithm	aracne	c3net	clr	mrnet
DE	GBM	5,903	2,678	1,718,328	1,682,334
	LGG	4,443	2,684	1,939,142	1,859,121

```
### plot details (colors & symbols)
mycols<-c('#e41a1c','#377eb8','#4daf4a','#984ea3','#ff7f00','#ffff33','#a65628')

### load network inference libraries
library(minet)
library(c3net)

### deferentially identified genes using TCGAbiolinks
names.genes.de <- rownames(dataDEGs)

### read biogrid info
library(downloader)
file <- "http://thebiogrid.org/downloads/archives/Release%20Archive/BIOGRID-3.4.133/BIOGRID-ALL
     -3.4.133.tab2.zip"
download(file ,basename(file))
unzip(basename(file),junkpaths =T)
tmp.biogrid <- read.csv(gsub("zip","txt",basename(file)), header=TRUE, sep="\t", stringsAsFactors=
    FALSE)

net.biogrid.de <- get.adjacency.biogrid(tmp.biogrid, names.genes.de)

for (cancertype in c("LGG", "GBM")) {

    if(cancertype == "GBM"){
        mydata <- dataFiltGBM[names.genes.de, ]
    }else if(cancertype == "LGG"){
        mydata <- dataFiltLGG[names.genes.de, ]
    }
    ### infer networks
    net.aracne <- minet(t(mydata), method = "aracne")
    net.mrnet <- minet(t(mydata))
    net.clr <- minet(t(mydata), method = "clr")
    net.c3net <- c3net(mydata)

    ### validate compared to biogrid network
    tmp.val <- list(validate(net.aracne, net.biogrid.de), validate(net.mrnet, net.biogrid.de),
                    validate(net.clr, net.biogrid.de), validate(net.c3net, net.biogrid.de))

    ### plot roc and compute auc for the different networks
    dev1 <- show.roc(tmp.val[[1]],cex=0.3,col=mycols[1],type="l")
    res.auc <- auc.roc(tmp.val[[1]])
    for(count in 2:length(tmp.val)){
        show.roc(tmp.val[[count]],device=dev1,cex=0.3,col=mycols[count],type="l")
        res.auc <- c(res.auc, auc.roc(tmp.val[[count]]))
    }

    legend("bottomright", legend=paste(c("aracne","mrnet","clr","c3net"), signif(res.auc,4), sep=": "),
          col=mycols[1:length(tmp.val)],lty=1, bty="n" )
    dev.copy2pdf(width=8,height=8,device = dev1, file = paste0("roc_biogrid_",cancertype,".pdf"))
    save(net.aracne, net.mrnet, net.clr, net.c3net, file=paste0("nets_",cancertype,".RData"))

}
```

In Figure 7, the obtained ROC curve and the corresponding area under curve (AUC) are presented. It can be observed that CLR and MRNET perform best when comparing the inferred network with known interactions from the BioGrid database.

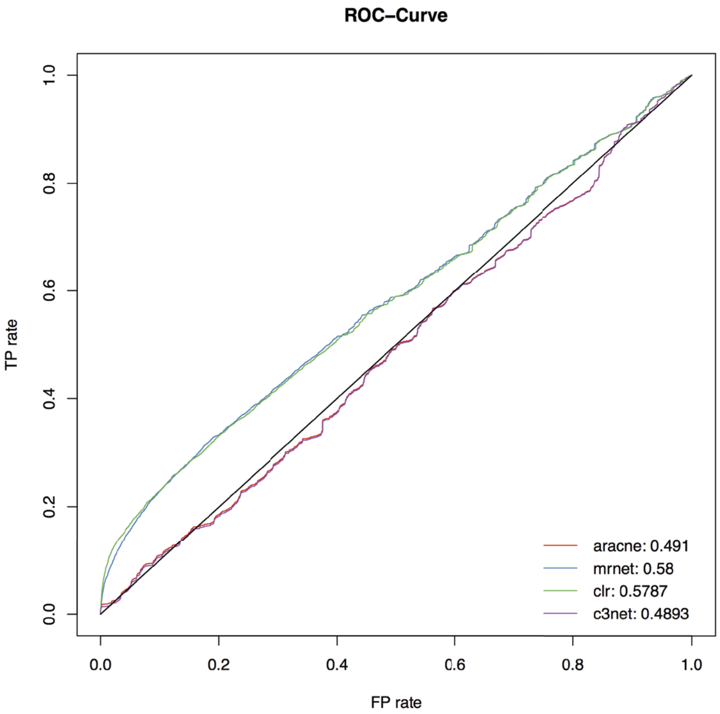


Figure 7. ROC with corresponding AUC for inferred GBM networks compared to BioGrid interactions using 2901 genes.

Epigenetic analysis

The DNA methylation is an important component in numerous cellular processes, such as embryonic development, genomic imprinting, X-chromosome inactivation, and preservation of chromosome stability[42].

In mammals DNA methylation is found sparsely but globally, distributed in definite CpG sequences throughout the entire genome; however, there is an exception. CpG islands (CGIs) which are short interspersed DNA sequences that are enriched for GC. These islands are normally found in sites of transcription initiation and their methylation can lead to gene silencing[43].

Thus, the investigation of the DNA methylation is crucial to understanding regulatory gene networks in cancer as the DNA methylation represses transcription[44]. Therefore, the DMR (Differentially Methylation Region) detection can help us investigate regulatory gene networks.

This section describes the analysis of DNA methylation using the Bioconductor package TCGAbiolinks[8]. For this analysis, and due to the time required to perform it, we selected only 10 LGG samples and 10 GBM samples that have both DNA methylation data from Infinium HumanMethylation450 and gene expression from Illumina HiSeq 2000 RNA Sequencing Version 2 analysis (lines 1–56 of the Listing 18 describes how to make the data acquisition). We started by checking the mean DNA methylation of different groups of samples, then performed a DMR in which we search for regions of possible biological significance, (e.g., regions that are methylated in one group and unmethylated in the other). After finding these regions, they can be visualized using heatmaps.

Visualizing the mean DNA methylation of each patient. It should be highlighted that some pre-processing of the DNA methylation data was done. The DNA methylation data from the 450k platform has three types of probes cg (CpG loci), ch (non-CpG loci) and rs (SNP assay). The last type of probe can be used for sample identification and tracking and should be excluded for differential methylation analysis according to the ilumina manual. Therefore, the rs probes were removed (see Listing 18 lines 68). Also, probes in chromosomes X, Y were removed to eliminate potential artifacts originating from the presence of a different proportion of males and females[45]. The last pre-processing steps were to remove probes with at least one NA value (see Listing 18 lines 65).

After this pre-processing step and using the function *TCGAvisualize_meanMethylation* function, we can look at the mean DNA methylation of each patient in each group. It receives as argument a *summarizedExperiment* object with the DNA methylation data, and the arguments *groupCol* and *subgroupCol* which should be two columns from the sample information matrix of the *summarizedExperiment* object (accessed by the *colData* function) (see Listing 18 lines 70–74).

```
 1 #----------------------------
 2 # Obtaining DNA methylation
 3 #----------------------------
 4 library(TCGAbiolinks)
 5 library(stringr)
 6 # Samples
 7 matched_met_exp <- function(project, n = NULL){
 8     # get primary solid tumor samples: DNA methylation
 9     message("Download DNA methylation information")
10     met450k <- GDCquery(project = project,
11                         data.category = "DNA methylation",
12                         platform = "Illumina Human Methylation 450",
13                         legacy = TRUE,
14                         sample.type = c("Primary solid Tumor"))
15     met450k.tp <- met450k$results[[1]]$cases
16 
17     # get primary solid tumor samples: RNAseq
18     message("Download gene expression information")
19     exp <- GDCquery(project = project,
20                     data.category = "Gene expression",
21                     data.type = "Gene expression quantification",
22                     platform = "Illumina HiSeq",
23                     file.type = "results",
24                     sample.type = c("Primary solid Tumor"),
25                     legacy = TRUE)
26     exp.tp <- exp$results[[1]]$cases
27     print(exp.tp[1:10])
28     # Get patients with samples in both platforms
29     patients <- unique(substr(exp.tp,1,15)[substr(exp.tp,1,12) %in% substr(met450k.tp,1,12)])
30     if(!is.null(n)) patients <- patients[1:n] # get only n samples
31     return(patients)
32 }
33 lgg.samples <- matched_met_exp("TCGA-LGG", n = 10)
34 gbm.samples <- matched_met_exp("TCGA-GBM", n = 10)
35 samples <- c(lgg.samples,gbm.samples)
36 
37 
38 #-----------------------------------
39 # 1 - Methylation
40 # ----------------------------------
41 # For methylation it is quicker in this case to download the tar.gz file
42 # and get the samples we want instead of downloading files by files
43 query.lgg <- GDCquery(project = "TCGA-LGG",
44                       data.category = "DNA methylation",
45                       platform = "Illumina Human Methylation 450",
46                       legacy = TRUE, barcode = lgg.samples)
47 GDCdownload(query.lgg)
48 met.lgg <-GDCprepare(query.lgg, save = FALSE)
49 
50 query.gbm <- GDCquery(project = "TCGA-GBM",
51                       data.category = "DNA methylation",
52                       platform = "Illumina Human Methylation 450",
53                       legacy = TRUE, barcode = gbm.samples)
54 GDCdownload(query.gbm)
55 met.gbm <- GDCprepare(query.gbm, save = FALSE)
56 met <- SummarizedExperiment::cbind(met.lgg, met.gbm)
57 
```

```
#----------------------------
# Mean methylation
#----------------------------
# Plot a barplot for the groups in the disease column in the
# summarizedExperiment object

# remove probes with NA (similar to na.omit)
met <- subset(met,subset = (rowSums(is.na(assay(met))) == 0))

# remove probes in chromossomes X, Y and NA
met <- subset(met,subset = !as.character(seqnames(met)) %in% c("chrNA","chrX","chrY"))

TCGAvisualize_meanMethylation(met,
                              groupCol = "disease_type",
                              group.legend  = "Groups",
                              filename = "mean_lgg_gbm.png",
                              print.pvalue = TRUE)
```

Listing 18. Visualizing the DNA mean methylation of groups

Figure 8 illustrates a mean DNA methylation plot for each sample in the GBM group (140 samples) and a mean DNA methylation for each sample in the LGG group. Genome-wide view of the data highlights a difference between the groups of tumors (p-value = $6.1x10^{-06}$).

Searching for differentially methylated CpG sites. The next step is to define differentially methylated CpG sites between the two groups. This can be done using the *TCGAanalyze_DMR* function (see Listing 19). The DNA methylation data (level 3) is presented in the form of beta-values that uses a scale ranging from 0.0 (probes completely unmethylated) up to 1.0 (probes completely methylated).

To find these differentially methylated CpG sites, first, the function calculates the difference between the mean DNA methylation (mean of the beta-values) of each group for each probe. Second, it test for differential expression between two groups using the Wilcoxon test adjusting by the Benjamini-Hochberg method. Arguments of TCGAanalyze_DMR was set to require a minimum absolute beta-values difference of 0.25 and an adjusted p-value of less than 10^{-2}.

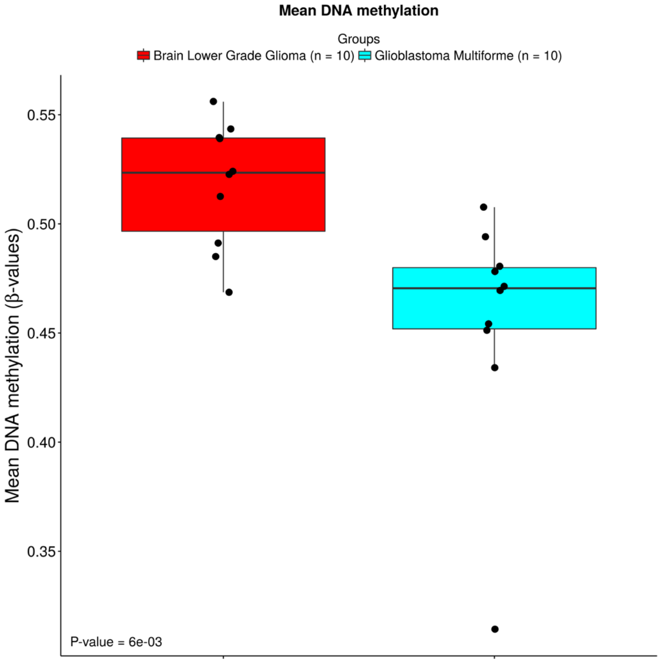


Figure 8. Boxplot of mean DNA methylation of each sample (black dots).

After these tests, a volcano plot (x-axis: difference of mean DNA methylation, y-axis: statistical significance) is created to help users identify the differentially methylated CpG sites and return the object with the results in the rowRanges. Figure 9 shows the volcano plot produced by Listing 19. This plot aids the user in selecting relevant thresholds, as we search for candidate biological DMRs.

```
# Becareful! Depending on the number of probes and samples this function might take some days.
# To make this example faster we used only the chromosome 9
# This should take some minutes
met.chr9 <- subset(met,subset = as.character(seqnames(met)) %in% c("chr9"))

met.chr9 <- TCGAanalyze_DMR(met.chr9,
                            groupCol = "disease_type", # a column in the colData matrix
                            group1 = "Glioblastoma Multiforme", # a type of the the disease type column
                            group2= "Brain Lower Grade Glioma", # a type of the the disease column
                            p.cut = 10^-2,
                            diffmean.cut = 0.25,
                            legend = "State",
                            plot.filename = "LGG_GBM_metvolcano.png",
                            cores = 1 # if set to 1 there will be a progress bar
)
```

Listing 19. Finding differentially methylated CpG sites

To visualize the level of DNA methylation of these probes across all samples, we use heatmaps that can be generate by the bioconductor package complexHeatmap[30]. To create a heatmap using the complexHeatmap package, the user should provide at least one matrix with the DNA methylation levels. Also, annotation layers can be added and placed at the bottom, top, left side and right side of the heatmap to provide additional metadata description. The Listing 20 shows the code to produce the heatmap of a DNA methylation datum (Figure 10).

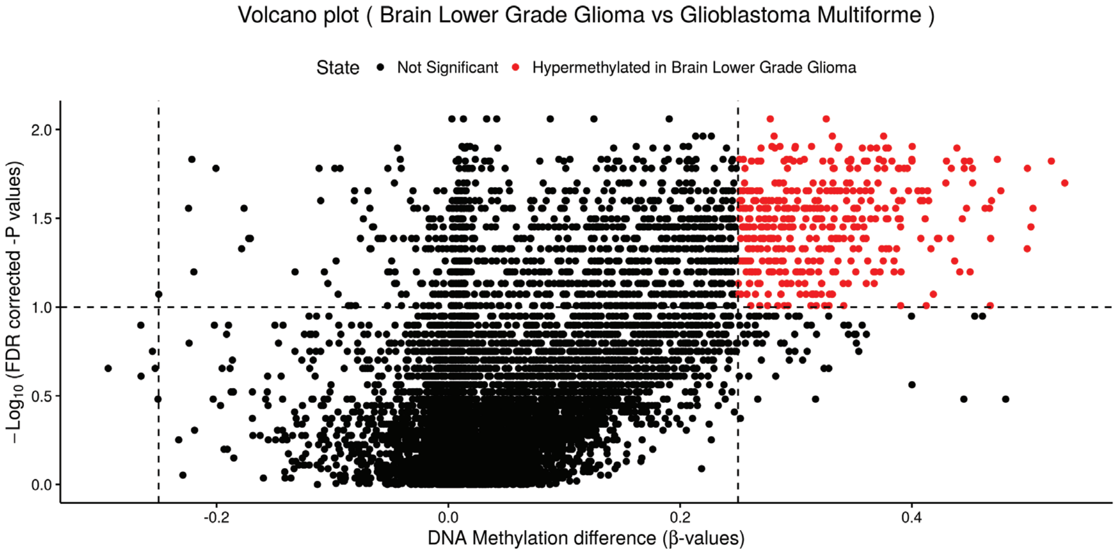


Figure 9. Volcano plot: searching for differentially methylated CpG sites (x-axis:difference of mean DNA methylation, y-axis: statistical significance).

```
 1 #--------------------------
 2 # DNA methylation heatmap
 3 #--------------------------
 4 library(ComplexHeatmap)
 5
 6 clin.gbm <- GDCquery_clinic("TCGA-GBM", "Clinical")
 7 clin.lgg <- GDCquery_clinic("TCGA-LGG", "Clinical")
 8 clinical <- plyr::rbind.fill(clin.lgg, clin.gbm)
 9
10 # get the probes that are Hypermethylated or Hypomethylated
11 # met is the same object of the section 'DNA methylation analysis'
12 sig.met <- met.chr9[values(met.chr9)[,"status.Glioblastoma.Multiforme.Brain.Lower.Grade.Glioma"] %in%
      c("Hypermethylated","Hypomethylated"),]
13
14 # To speed up the example, we will select no more than 100 probes
15 nb.probes <- ifelse(nrow(sig.met) > 100, 100, nrow(sig.met)) # If there is more than 100 get 100
16 sig.met.100 <- sig.met[1:nb.probes,]
17
18 # top annotation, which sampples are LGG and GBM
19 # We will add clinical data as annotation of the samples
20 # we will sort the clinical data to have the same order of the DNA methylation matrix
21 clinical.order <- clinical[match(substr(colnames(sig.met.100),1,12),clinical$bcr_patient_barcode),]
22 ta = HeatmapAnnotation(df = clinical.order[,c("disease","gender","vital_status","race")],
23                       col = list(disease = c("LGG" = "grey", "GBM" = "black"),
24                                  gender = c("male"="blue","female"="pink")))
25
26 # row annotation: add the status for LGG in relation to GBM
27 # For exmaple: status.gbm.lgg Hypomethyated means that the
28 # mean DNA methylation of probes for lgg are hypomethylated
29 # compared to GBM ones.
30 ra = rowAnnotation(df = values(sig.met.100)["status.Glioblastoma.Multiforme.Brain.Lower.Grade.Glioma"],
31                  col = list("status.Glioblastoma.Multiforme.Brain.Lower.Grade.Glioma" = c("
     Hypomethylated" = "orange",
32                                                 "Hypermethylated" = "darkgreen")),
33                   width = unit(1, "cm"))
34
35 heatmap <- Heatmap(assay(sig.met.100),
36                    name = "DNA methylation",
37                    col = matlab::jet.colors(200),
38                    show_row_names = F,
39                    cluster_rows = T,
40                    cluster_columns = F,
41                    show_column_names = F,
42                    bottom_annotation = ta,
43                    column_ title = "DNA methylation")
44 # Save to pdf
45 pdf("heatmap.pdf",width = 10, height = 8)
46 draw(heatmap, annotation_legend_side = "bottom")
47 dev.off()
```

Listing 20. Creating heatmaps for DNA methylation using ComplexHeatmap

Motif analysis. Motif discovery is the attempt to extract small sequence signals hidden within largely non-functional intergenic sequences. These small sequence nucleotide signals (6–15 bp) might have a biological significance as they can be used to control the expression of genes. These sequences are called Regulatory motifs. The bioconductor package rGADEM[46,47] provides an efficient *de novo* motif discovery algorithm for large-scale genomic sequence data.

The user may be interested in looking for unique signatures in the regions defined by 'differentially methylated' to identify candidate transcription factors that could bind to these elements affected by the accumulation or absence of DNA methylation. For this analysis we use a sequence of 100 bases before and after the probe location (See lines 6–8

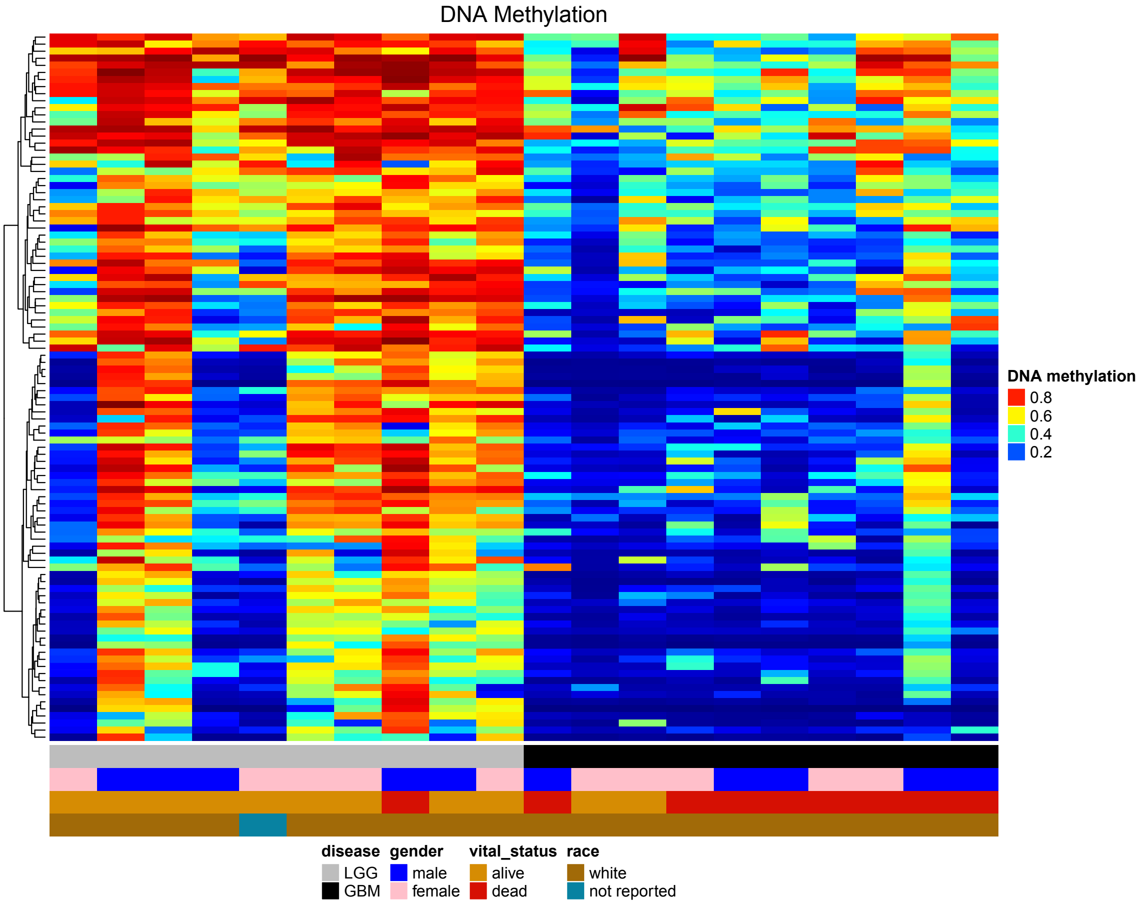


Figure 10. Heatmap of DNA methylation in probes. Rows are probes and columns are samples (patients). The DNA methylation values range from 0.0 (completely DNA unmethylated, blue) to 1.0 (completely DNA methylated, red). The groups of each sample were annotated in the top bar and the DNA methylation status for each probe was annotated in the right bar.

in the Listing 21). An object will be returned which contains all relevant information about your motif analysis (i.e., sequence consensus, pwm, chromosome, p-value, etc).

Using bioconductor package motifStack[48] it is possible to generate a graphic representation of multiple motifs with different similarity scores (see Figure 11).

```
library(rGADEM)
library(BSgenome.Hsapiens.UCSC.hg19)
library(motifStack)

probes <- rowRanges(met.chr9)[values(met.chr9)[,"status.Glioblastoma.Multiforme.
      Brain.Lower.Grade.Glioma"] %in% c("Hypermethylated" ,"Hypomethylated"),]
# Get hypo/hyper methylated probes and make a 200bp window
# surrounding each probe.
sequence <- RangedData(space=as.character(probes@seqnames),
                        IRanges(start=probes@ranges@start - 100,
                                end=probes@ranges@start + 100), strand="*")
#look for motifs
gadem <- GADEM(sequence, verbose=FALSE, genome=Hsapiens)

# How many motifs were found?
length(gadem@motifList)
```

```

# get the number of occurences
nOccurrences(gadem)

# view all sequences consensus
consensus(gadem)

# print motif
pwm <- getPWM(gadem)
pfm  <- new("pfm",mat=pwm[[1]],name="Novel Site 1")
plotMotifLogo(pfm)

# Number of instances of motif 1?
length(gadem@motifList[[1]]@alignList)
```

Listing 21. rGADEM: *de novo* motif discovery

After rGADEM returns it’s results, the user can use MotIV package[49–52] to start the motif matching analysis (line 4 of Listing 22). The result is shown in Figure 12.

```
 1 library(MotIV)
 2
 3 analysis.jaspar <- motifMatch(pwm)
 4 summary(analysis.jaspar)
 5 plot(analysis.jaspar, ncol=1, top=5, rev=FALSE,
 6      main="", bysim=TRUE, cex=0.3)
 7
 8 #  visualize the quality of the results around the alignments
 9 # E-value give an estimation of the match.
10 alignment <- viewAlignments(analysis.jaspar )
11 print(alignment[[1]])
```

Listing 22. MotIV: motifs matches analysis.

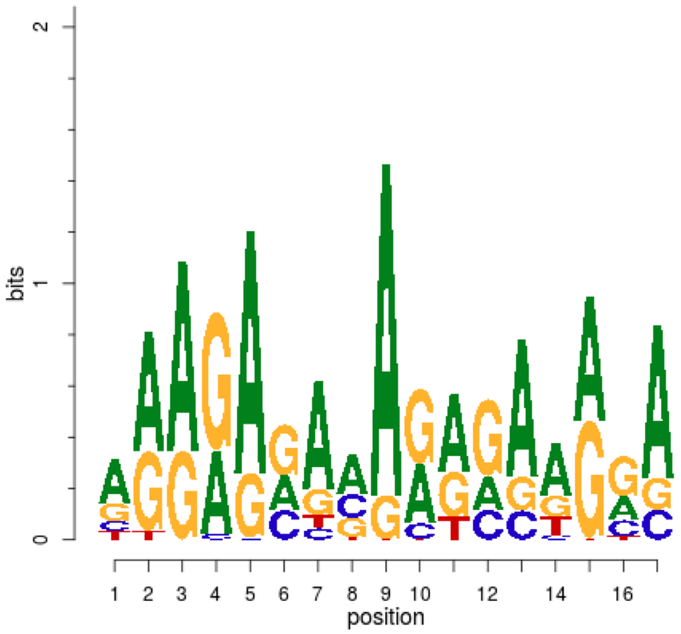


Figure 11. Motif logos found during *de-novo* motif analysis.

Integrative (Epigenomic & Transcriptomic) analysis

Recent studies have shown that providing a deep integrative analysis can aid researchers in identifying and extracting biological insight from high through put data[42,53,54]. In this section, we will introduce a Bioconductor package called ELMER to identify regulatory enhancers using gene expression + DNA methylation data + motif analysis. In addition, we show how to integrate the results from the previous sections with important epigenomic data derived from both the ENCODE and Roadmap.

Integration of DNA methylation & gene expression data. After finding differentially methylated CpG sites, one might ask whether nearby genes also undergo a change in expression either an increase or a decrease. DNA methylation at promoters of genes has been shown to be associated with silencing of the respective gene. The starburst plot is proposed to combine information from two volcano plots and is applicable for studies of DNA methylation and gene expression[55]. Even though, being desirable that both gene expression and DNA methylation data are from the same samples, the starburst plot can be applied as a meta-analysis tool, combining data from different samples[56]. We used the *TCGAvisualize_starburst* function to create a starburst plot. The log_{10} (FDR-corrected P value) for DNA methylation is plotted on the x axis, and for gene expression on the y axis, for each gene. The horizontal black dashed line shows the FDR-adjusted P value of 10^{-2} for the expression data and the vertical ones shows the FDR-adjusted P value

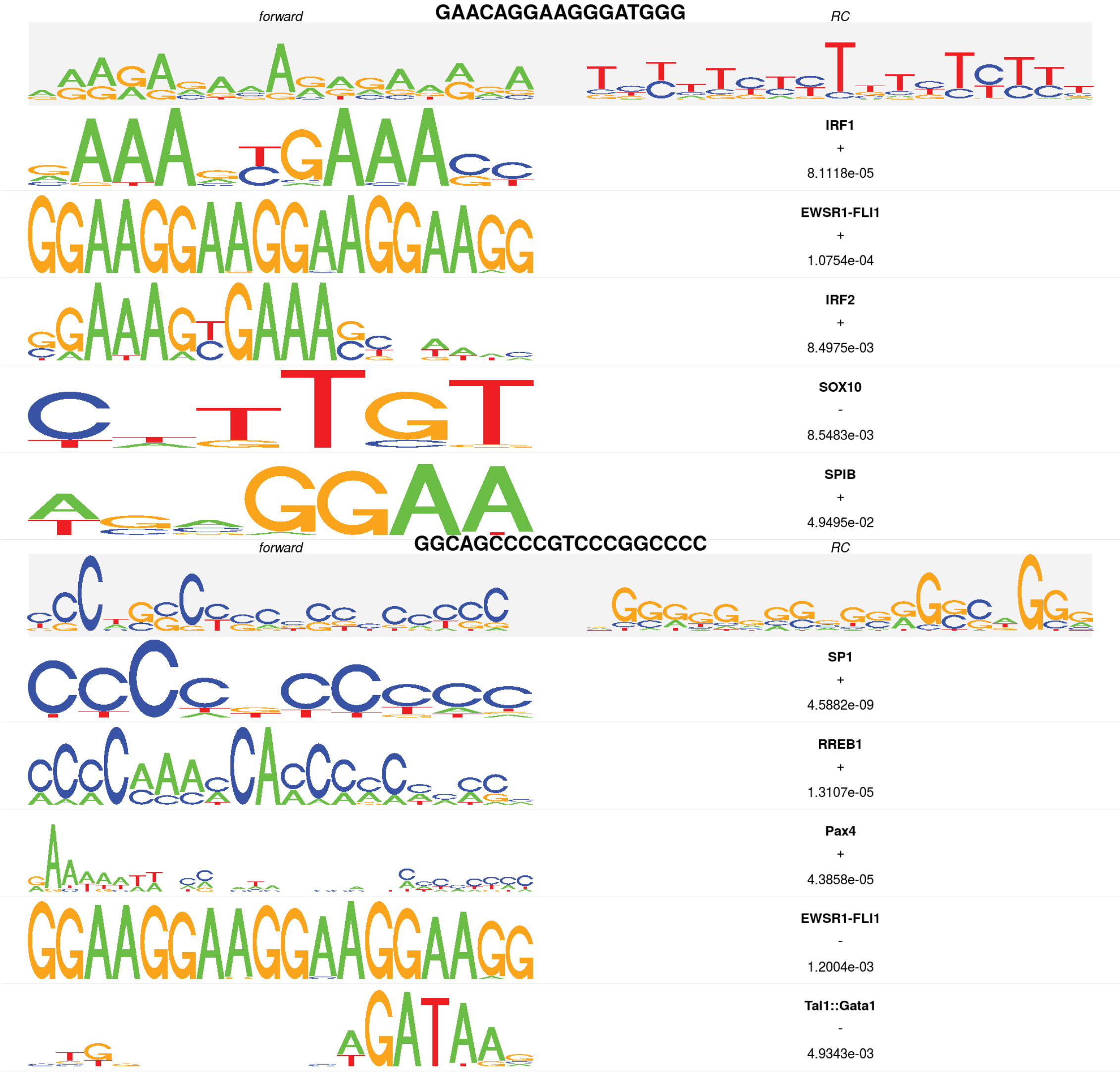


Figure 12. Identified transcription factors: the sequence logo, the name of the motif match and the p-value of the alignment.

of 10^{-2} for the DNA methylation data. The starburst plot for the Listing 23 is shown in Figure 13. While the argument met.p.cut and exp.p.cut controls the black dashed lines, the arguments *diffmean.cut* and *logFC.cut* will be used to highlight the genes that respects these parameters (circled genes in Figure 13). For the example below we set higher p.cuts trying to get the most significant list of pair gene/probes. But for the next sections we will use *exp.p.cut = 0.01* and *logFC.cut = 1* as the previous sections.

```
#------------------- Starburst plot -----------------------------
starburst <- TCGAvisualize_starburst(met.chr9,     # DNA methylation with results
                                     dataDEGs,     # DEG results
                                     group1 = "Glioblastoma Multiforme",
                                     group2 = "Brain Lower Grade Glioma",
                                     filename = "starburst.png",
                                     met.p.cut = 10^-2,
                                     exp.p.cut = 10^-2,
                                     diffmean.cut = 0.25,
                                     logFC.cut = 1,width = 15,height = 10,
                                     names = TRUE)
```

Listing 23. Starburst plot for comparison of DNA methylation and gene expression

ChIP-seq analysis. ChIP-seq is used primarily to determine how transcription factors and other chromatin-associated proteins influence phenotype-affecting mechanisms. Determining how proteins interact with DNA to regulate gene expression is essential for fully understanding many biological processes and disease states. The aim is to explore significant overlap datasets for inferring co-regulation or transcription factor complex for further investigation. A summary of the association of each histone mark is shown in Table 4. Besides, ChIP-seq data exists in the ROADMAP database and can be obtained through the AnnotationHub package[57] or from Roadmap web portal. The Table 5 shows the description for all the Roadmap files that are available through AnnotationHub. After obtaining the ChIP-seq data, we can then identify overlapping regions with the regions identified in the starburst plot. The narrowPeak files are the ones selected for this step.

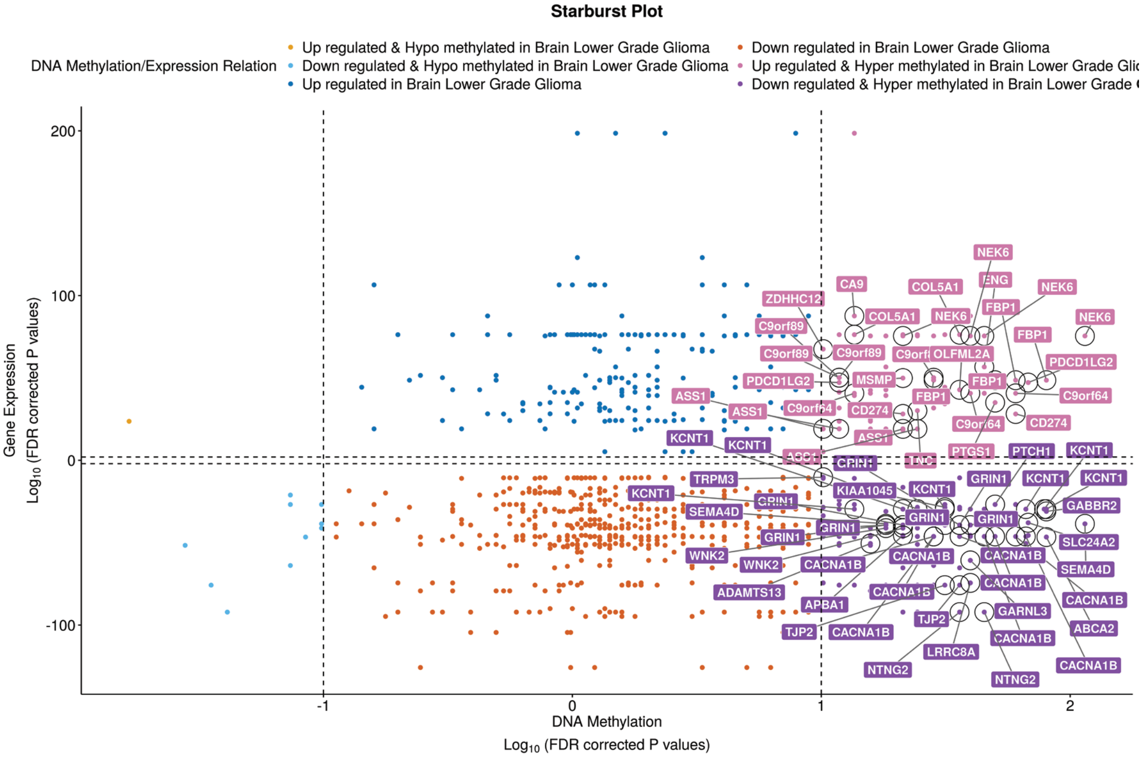


Figure 13. Starburst plot: x-axis is the log10 of the correct P-value for DNA methylation and the y-axis is the log10 of the correct P-value for the expression data. The starburst plot highlights nine distinct quadrants. Highlighted genes might have the potential for activation due to epigenetic alterations.

Table 4. Histone marks that define genomic elements.

Histone marks	Role
Histone H3 lysine 4 trimethylation (H3K4me3)	Promoter regions[58,59]
Histone H3 lysine 4 monomethylation (H3K4me1)	Enhancer regions[58]
Histone H3 lysine 36 trimethylation (H3K36me3)	Transcribed regions
Histone H3 lysine 27 trimethylation (H3K27me3)	Polycomb repression[60]
Histone H3 lysine 9 trimethylation (H3K9me3)	Heterochromatin regions[61]
Histone H3 acetylated at lysine 27 (H3K27ac)	Increase activation of genomic elements[62–64]
Histone H3 lysine 9 acetylation (H3K9ac)	Transcriptional activation[65]

Table 5. ChIP-seq data file types available in AnnotationHub.

File	Description
fc.signal.bigwig	Bigwig File containing fold enrichment signal tracks
pval.signal.bigwig	Bigwig File containing -log10(p-value) signal tracks
hotspot.fdr0.01.broad.bed.gz	Broad domains on enrichment for DNase-seq for consolidated epigenomes
hotspot.broad.bed.gz	Broad domains on enrichment for DNase-seq for consolidated epigenomes
broadPeak.gz	Broad ChIP-seq peaks for consolidated epigenomes
gappedPeak.gz	Gapped ChIP-seq peaks for consolidated epigenomes
narrowPeak.gz	Narrow ChIP-seq peaks for consolidated epigenomes
hotspot.fdr0.01.peaks.bed.gz	Narrow DNasePeaks for consolidated epigenomes
hotspot.all.peaks.bed.gz	Narrow DNasePeaks for consolidated epigenomes
.macs2.narrowPeak.gz	Narrow DNasePeaks for consolidated epigenomes
coreMarks_mnemonics.bed.gz	15 state chromatin segmentations
mCRF_FractionalMethylation.bigwig	MeDIP/MRE(mCRF) fractional methylation calls
RRBS_FractionalMethylation.bigwig	RRBS fractional methylation calls
WGBS_FractionalMethylation.bigwig	Whole genome bisulphite fractional methylation calls

For a complete pipeline with Chip-seq data, bioconductor provides excellent tutorials to work with ChIP-seq and we encourage our readers to review the following article[66].

The first step shown in Listing 24 is to download the chip-seq data. The function *query* received as argument the annotationHub database (ah) and a list of keywords to be used for searching the data, *EpigenomeRoadmap* is selecting the roadmap database, *consolidated* is selecting only the consolidate epigenomes, *brain* is selecting the brain samples, E068 is one of the epigenomes for brain (keywords can be seen in the summary table) and narrowPeak is selecting the type of file. The data downloaded is a processed data from an integrative Analysis of 111 reference human epigenomes[67].

```
library(AnnotationHub)
library(pbapply)
#------------------- Working with ChipSeq data ---------------
# Step 1: download histone marks for a brain samples.
#------------------------------------------------------------
ah = AnnotationHub() # loading annotation hub database

```

```
# Searching for brain consolidated epigenomes in the roadmap database
bpChipEpi_brain <- query(ah, c("EpigenomeRoadMap", "narrowPeak", "chip", "consolidated",
"brain", "E068"))

# Get chip-seq data
histone.marks <- pblapply(names(bpChipEpi_brain), function(x){ah[[x]]})
names(histone.marks) <- names(bpChipEpi_brain)
```

Listing 24. Download chip-seq data

The Chipseeker package[68] implements functions that uses Chip-seq data to retrieve the nearest genes around the peak, to annotate genomic region of the peak, among others. Also, it provides several visualization functions to summarize the coverage of the peak, average profile and heatmap of peaks binding to TSS regions, genomic annotation, distance to TSS and overlap of peaks or genes.

After downloading the histone marks (see Listing 24), it is useful to verify the average profile of peaks binding to hypomethylated and hypermethylated regions, which will help the user understand better the regions found. Listing 25 shows an example of code to plot the average profile. Figure 14 shows the result.

To help the user better understand the regions found in the DMR analysis, we downloaded histone marks specific for brain tissue using the AnnotationHub package that can access the Roadmap database (Listing 24). Next, the Chipseeker was used to visualize how histone modifications are enriched onto hypomethylated and hypermethylated regions, (Listing 25). The enrichment heatmap and the average profile of peaks binding to those regions is shown in Figure 14 and Figure 15 respectively.

The hypomethylated and hypermethylated regions are enriched for H3K4me3, H3K9ac, H3K27ac and H3K4me1 which indicates regions of enhancers, promoters and increased activation of genomic elements. However, these regions are associated neither with transcribed regions nor Polycomb repression as the H3K36me3 and H3K27me3 heatmaps does not show an enrichment nearby the position 0, and the average profile also does not show a peak at position 0.

```
library(ChIPseeker)
library(pbapply)
library(SummarizedExperiment)
library(GenomeInfoDb)
library(ggplot2)
library(AnnotationHub)

# Create a GR object based on the hypo/hypermethylated probes.
probes <- keepStandardChromosomes(rowRanges(met.chr9)[values(met.chr9)[,"status.Glioblastoma
        Multiforme.Brain Lower Grade Glioma"] %in% c("Hypermethylated", "Hypomethylated"),])
# Defining a window of 3kbp - 3kbp_probe_3kbp
probes@ranges <- IRanges(start = c(probes@ranges@start - 3000), end = c(probes@ranges@start
+ 3000))

### Profile of ChIP peaks binding to TSS regions
# First of all, for calculate the profile of ChIP peaks binding to TSS regions, we should
# prepare the TSS regions, which are defined as the flanking sequence of the TSS sites.
# Then align the peaks that are mapping to these regions, and generate the tagMatrix.
tagMatrixList <- pblapply(histone.marks, function(x) {
    getTagMatrix(keepStandardChromosomes(x), windows = probes, weightCol = "score")
})
names(tagMatrixList) <- basename(bpChipEpi_brain$title)
names(tagMatrixList) <- gsub(".narrowPeak.gz","",names(tagMatrixList)) # remove file type from name
names(tagMatrixList) <- gsub("E068-","",names(tagMatrixList)) # remove file type from name

```

```
pdf("chip_heatmap.pdf", height = 5, width = 10)
tagHeatmap(tagMatrixList, xlim=c(-3000, 3000),color = NULL)
dev.off()

p <- plotAvgProf(tagMatrixList, xlim = c(-3000,3000), xlab = "Genomic Region (5'->3', centered on CpG)")
# We are centreing in the CpG instead of the TSS. So we'll change the labels manually
p <- p + scale_x_continuous(breaks=c(-3000,-1500,0,1500,3000),labels=c(-3000,-1500,"CpG",1500,3000))
library(ggthemes)
pdf("chip-seq.pdf", height = 5, width = 7)
p + theme_few() + scale_colour_few(name="Histone marks") +guides(colour = guide_legend(override.aes =
      list(size=4)))
dev.off()
```

Listing 25. Average profile plot

To annotate the location of a given peak in terms of genomic features, annotatePeak assigns peaks to genomic annotation in “annotation” column of the output, which includes whether a peak is in the TSS, Exon, 5’ UTR, 3’ UTR, Intronic or Intergenic location.

```
require(TxDb.Hsapiens.UCSC.hg19.knownGene)
txdb <- TxDb.Hsapiens.UCSC.hg19.knownGene
peakAnno <- annotatePeak(probes, tssRegion=c(-3000, 3000), TxDb=txdb, annoDb="org.Hs.eg.db")
plotAnnoPie(peakAnno)
```

Listing 26. Annotate the location of a given peak in terms of genomic features

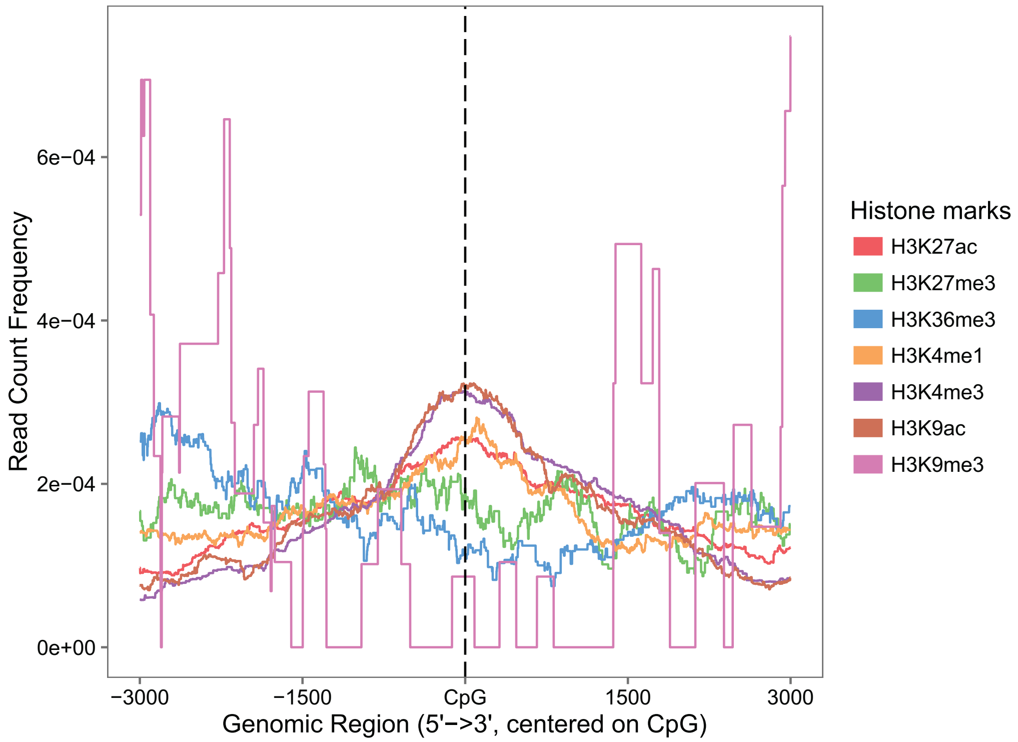


Figure 14. Average profiles for histone markers H3K27ac, H3K27me3, H3K36me3, H3K4me1, H3K4me3, H3K9ac, and H3K9me3. The figure indicates that the differentially methylated regions overlaps regions of enhancers, promoters and increased activation of genomic elements.

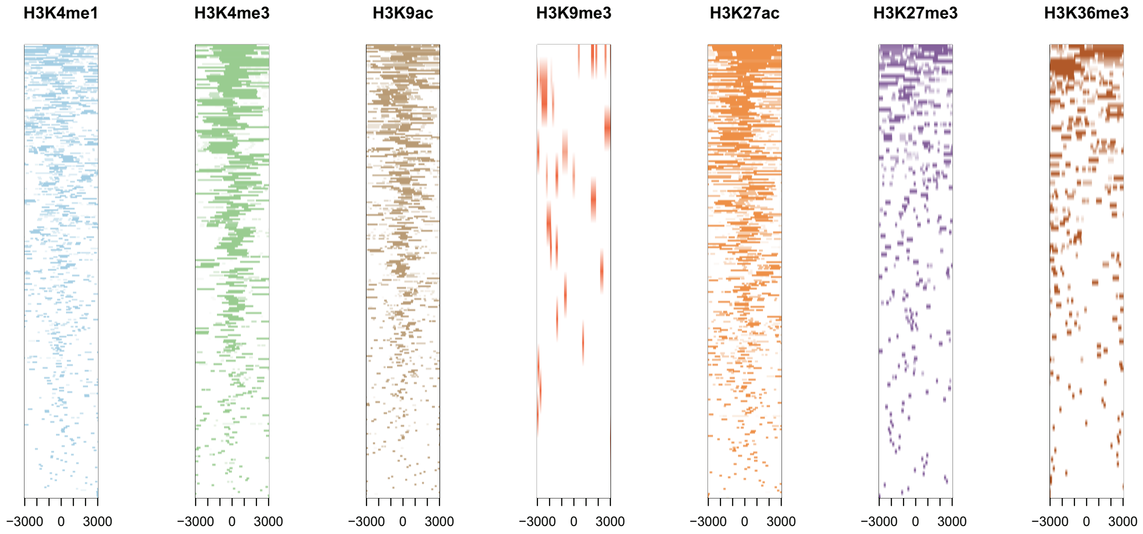


Figure 15. Heatmap of histone marks H3K4me1, H3K4me3, H3K27ac, H3K9ac, H3K9me3, H3K27me3 and H3K36me3 for brain tissues.

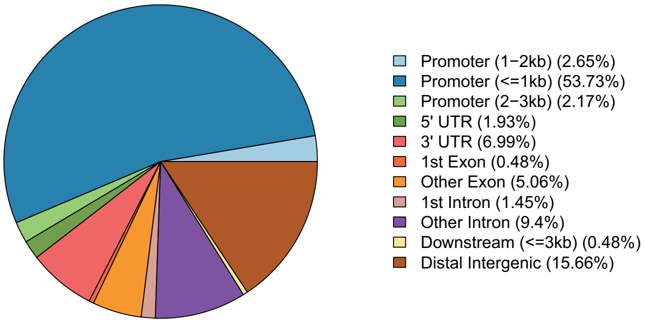


Figure 16. Feature distribution: annotation the region of the probes.

Identification of Regulatory Enhancers. Recently, many studies suggest that enhancers play a major role as regulators of cell-specific phenotypes leading to alteration in transcriptomes realated to diseases[69–72]. In order to investigate regulatory enhancers that can be located at long distances upstream or downstream of target genes Bioconductor offer the Enhancer Linking by Methylation/Expression Relationship (ELMER) package. This package is designed to combine DNA methylation and gene expression data from human tissues to infer multi-level cis-regulatory networks. It uses DNA methylation to identify enhancers and correlates their state with expression of nearby genes to identify one or more transcriptional targets. Transcription factor (TF) binding site analysis of enhancers is coupled with expression analysis of all TFs to infer upstream regulators. This package can be easily applied to TCGA public available cancer data sets and custom DNA methylation and gene expression data sets[73].

ELMER analysis have 5 main steps:

1. Identify distal enhancer probes on HM450K.
2. Identify distal enhancer probes with significantly different DNA methylation level in control group and experiment group.
3. Identify putative target genes for differentially methylated distal enhancer probes.
4. Identify enriched motifs for the distal enhancer probes which are significantly differentially methylated and linked to putative target gene.
5. Identify regulatory TFs whose expression associate with DNA methylation at motifs.

This section shows how to use ELMER to analyze TCGA data using as example LGG and GBM samples.

Preparing the data for ELMER package. After downloading the data with TCGAbiolinks package, some steps are still required to use TCGA data with ELMER. These steps can be done with the function *TCGAprepare_elmer*. This function for the DNA methylation data will remove probes with NA values in more than 20% samples and remove the annotation data, for RNA expression data it will take the log2(expression + 1) of the expression matrix in order to linearize the relation between DNA methylation and expression. Also, it will prepare the row names of the matrix as required by the package.

The Listing 27 shows how to use TCGAbiolinks[8] to search, download and prepare the data for the ELMER package. Due to time and memory constraints, we will use in this example only data from 10 LGG patients and 10 GBM patients that have both DNA methylation and gene expression data. This samples are the same used in the previous steps.

```
#----------- 8.3  Identification of Regulatory Enhancers -------
library(TCGAbiolinks)
library(stringr)
# Samples: primary solid tumor w/ DNA methylation and gene expression
matched_met_exp <- function(project, n = NULL){
    # get primary solid tumor samples: DNA methylation
    message("Download DNA methylation information")
    met450k <- GDCquery(project = project,
                        data.category = "DNA methylation",
                        platform = "Illumina Human Methylation 450",
                        legacy = TRUE,
                        sample.type = c("Primary solid Tumor"))
    met450k.tp <-  met450k$results[[1]]$cases

    # get primary solid tumor samples: RNAseq
    message("Download gene expression information")
    exp <- GDCquery(project = project,
                    data.category = "Gene expression",
                    data.type = "Gene expression quantification",
                    platform = "Illumina HiSeq",
                    file.type =  "results",
                    sample.type = c("Primary solid Tumor"),
                    legacy = TRUE)
     exp.tp <-  exp$results[[1]]$cases
     print(exp.tp[1:10])
     # Get patients with samples in both platforms
     patients <- unique(substr(exp.tp,1,15)[substr(exp.tp,1,12) %in% substr(met450k.tp,1,12)])
     if(!is.null(n)) patients <- patients[1:n] # get only n samples
     return(patients)
}
lgg.samples <- matched_met_exp("TCGA-LGG", n = 10)
gbm.samples <- matched_met_exp("TCGA-GBM", n = 10)
samples <- c(lgg.samples,gbm.samples)

#-----------------------------------
# 1 -  Methylation
#----------------------------------
query.lgg <- GDCquery(project = "TCGA-LGG",
                      data.category = "DNA methylation",
                      platform = "Illumina Human Methylation 450",
                      legacy = TRUE,
                      barcode = lgg.samples)
GDCdownload(query.lgg)
met.lgg <-GDCprepare(query.lgg, save = FALSE)

query.gbm <- GDCquery(project = "TCGA-GBM",
                      data.category = "DNA methylation",
                      platform = "Illumina Human Methylation 450",
                      legacy = TRUE,
                      barcode = gbm.samples)
GDCdownload(query.gbm)
met.gbm <- GDCprepare(query.gbm, save = FALSE)
met.elmer <- SummarizedExperiment::cbind(met.lgg, met.gbm)
met.elmer <- TCGAprepare_elmer(met.elmer,
                               platform = "HumanMethylation450")

#-----------------------------------
# 2 - Expression
# -----------------------------------
query.exp.lgg <- GDCquery(project = "TCGA-LGG",
                data.category = "Gene expression",
                data.type = "Gene expression quantification",
                platform = "Illumina HiSeq",
                file.type = "results",
                legacy = TRUE, barcode = lgg.samples )
GDCdownload(query.exp.lgg)
exp.lgg <- GDCprepare(query.exp.lgg, save = FALSE)

query.exp.gbm <- GDCquery(project = "TCGA-GBM",
                data.category = "Gene expression",
                data.type = "Gene expression quantification",
                platform = "Illumina HiSeq",
                file.type = "results",
                legacy = TRUE, barcode = gbm.samples)
GDCdownload(query.exp.gbm)
exp.gbm <-GDCprepare(query.exp.gbm, save = FALSE)
exp.elmer <- SummarizedExperiment::cbind(exp.lgg, exp.gbm)
exp.elmer <- TCGAprepare_elmer(exp.elmer, platform = "IlluminaHiSeq_RNASeqV2")
```

Listing 27. Preparing TCGA data for ELMER's mee object

Finally, the ELMER input is a mee object that contains a DNA methylation matrix, an gene expression matrix, a probe information GRanges, the gene information GRanges and a data frame summarizing the data. It should be highlighted that samples without both the gene expression and DNA methylation data will be removed from the mee object.

By default the function *fetch.mee* that is used to create the mee object will separate the samples into two groups, the control group (normal samples) and the experiment group (tumor samples), but the user can relabel the samples to compare different groups. For the next sections, we will work with both the experimental group (LGG) and control group (GBM).

```
library(ELMER)
geneAnnot <- txs()
geneAnnot$GENEID <- paste0("ID",geneAnnot$GENEID)
geneInfo <- promoters(geneAnnot,upstream = 0, downstream = 0)
probe <- get.feature.probe()

# create mee object, use @ to access the matrices inside the object
mee <- fetch.mee(meth = met.elmer, exp = exp.elmer,
                 TCGA = TRUE, probeInfo = probe, geneInfo = geneInfo)

# Relabel GBM samples in the mee object: GBM is control
mee@sample$TN[mee@sample$ID %in% lgg.samples] <- "Control"
```

Listing 28. Creating mee object with TCGA data to be used in ELMER

ELMER analysis. After preparing the data into a *mee* object, we executed the five ELMER steps for both the hypo (distal enhancer probes hypomethylated in the LGG group) and hyper (distal enhancer probes hypermethylated in the LGG group) direction. The code is shown below. A description of how these distal enhancer probes are identified is found in the ELMER.data vignette.

```r
library(parallel)
# Available directions are hypo and hyper, we will use only hypo
# due to speed constraint
direction <- c("hyper")

for (j in direction){
    print(j)
    dir.out <- paste0("elmer/",j)
    dir.create(dir.out, recursive = TRUE)
    #--------------------------------------
    # STEP 3: Analysis                     |
    #--------------------------------------
    # Step 3.1: Get diff methylated probes |
    #--------------------------------------
    Sig.probes <- get.diff.meth(mee, cores=detectCores(),
                                dir.out =dir.out,
                                diff.dir=j,
                                pvalue = 0.01)

    #-------------------------------------------------------------
    # Step 3.2: Identify significant probe-gene pairs            |
    #-------------------------------------------------------------
    # Collect nearby 20 genes for Sig.probes
    nearGenes <- GetNearGenes(TRange=getProbeInfo(mee, probe=Sig.probes$probe),
                              cores=detectCores(),
                              geneAnnot=getGeneInfo(mee))

    pair <- get.pair(mee=mee,
                     probes=na.omit(Sig.probes$probe),
                     nearGenes=nearGenes,
                     permu.dir=paste0(dir.out,"/permu"),
                     dir.out=dir.out,
                     cores=detectCores(),
                     label= j,
                     permu.size=100, # For significant results use 10000
                     Pe = 0.01) # For significant results use 0.001

    Sig.probes.paired <- fetch.pair(pair=pair,
                                    probeInfo = getProbeInfo(mee),
                                    geneInfo = getGeneInfo(mee))
    Sig.probes.paired <-read.csv(paste0(dir.out,
                                        "/getPair.",j,
          ".pairs.significant.csv"),
                                stringsAsFactors=FALSE)[,1]


    #-------------------------------------------------------------
    # Step 3.3: Motif enrichment analysis on the selected probes |
    #-------------------------------------------------------------
    if(length(Sig.probes.paired) > 0){
        #-------------------------------------------------------------
        # Step 3.3: Motif enrichment analysis on the selected probes |
        #-------------------------------------------------------------
        enriched.motif <- get.enriched.motif(probes=Sig.probes.paired,
                                             dir.out=dir.out, label=j,
                                             background.probes = probe$name)
        motif.enrichment <- read.csv(paste0(dir.out,
                                    "/getMotif.",j,
              ".motif.enrichment.csv"),
                                    stringsAsFactors=FALSE)
        if(length(enriched.motif) > 0){
            #-------------------------------------------------------------
            # Step 3.4: Identifying regulatory TFs                        |
            #-------------------------------------------------------------
           print("get.TFs")

           TF <- get.TFs(mee = mee,
                         enriched.motif = enriched.motif,
                         dir.out = dir.out,
                         cores = detectCores(), label = j)
           TF.meth.cor <- get(load(paste0(dir.out,
                                    "/getTF.",j,
            ".TFs.with.motif.pvalue.rda")))
           save(TF, enriched.motif, Sig.probes.paired,
                pair, nearGenes, Sig.probes,
                motif.enrichment, TF.meth.cor,
                file=paste0(dir.out,"/ELMER_results_",j,".rda"))
        }
    }
}
```

Listing 29. Running ELMER analysis

When ELMER Identifies the enriched motifs for the distal enhancer probes which are significantly differentially methylated and linked to putative target gene, it will plot the Odds Ratio (x axis) for the each motifs found. The list of motifs for the hyper direction (probes hypermethylated in LGG group compared to the GBM group) is found in the Figure 17. We selected motifs that had a minimum incidence of 10 in the given probes set and the smallest lower boundary of 95% confidence interval for Odds Ratio of 1.1. These both values are the default from the ELMER package.

The analysis found 16 enriched motifs for the hyper direction and no enriched motifs for the hypo direction. After finding the enriched motifs, ELMER identifies regulatory transcription factors (TFs) whose expression is associated with DNA methylation at motifs. ELMER automatically creates a TF ranking plot for each enriched motif. This plot shows the TF ranking plots based on the association score $(-log(Pvalue))$ between TF expression and DNA methylation of the motif. We can see in Figure 18 that the top three TFs that are associated with that FOX motif are FOXD3, HMGA2 and HIST1H2BH.

The output of this step is a data frame with the following columns:

1. motif: the names of motif.
2. top.potential.TF: the highest ranking upstream TFs which are known recognized the motif.
3. potential.TFs: TFs which are within top 5% list and are known recognized the motif. top5percent: all TFs which are within top 5% list considered candidate upstream regulators

Also, for each motif we can take a look at the three most relevant TFs. For example, Figure 19 shows the average DNA methylation level of sites with the FOX motif plotted against the expression of the TFs FOXD3, HIST1H1D, HIST1H2BH and HMGA2. We can see that the experiment group (LGG samples) has a lower average methylation level of sites with the FOX motif plotted and a higher average expression of the TFs.

```
scatter.plot(mee, category="TN", save=T, lm_line=TRUE,
               byTF=list(TF=c("FOXD3","HIST1H1D","HIST1H2BH","HMGA2"),
               probe=enriched.motif[["FOX"]]))
```

Listing 30. Visualizing the average DNA methylation level of sites with a chosen motif vs TF expression

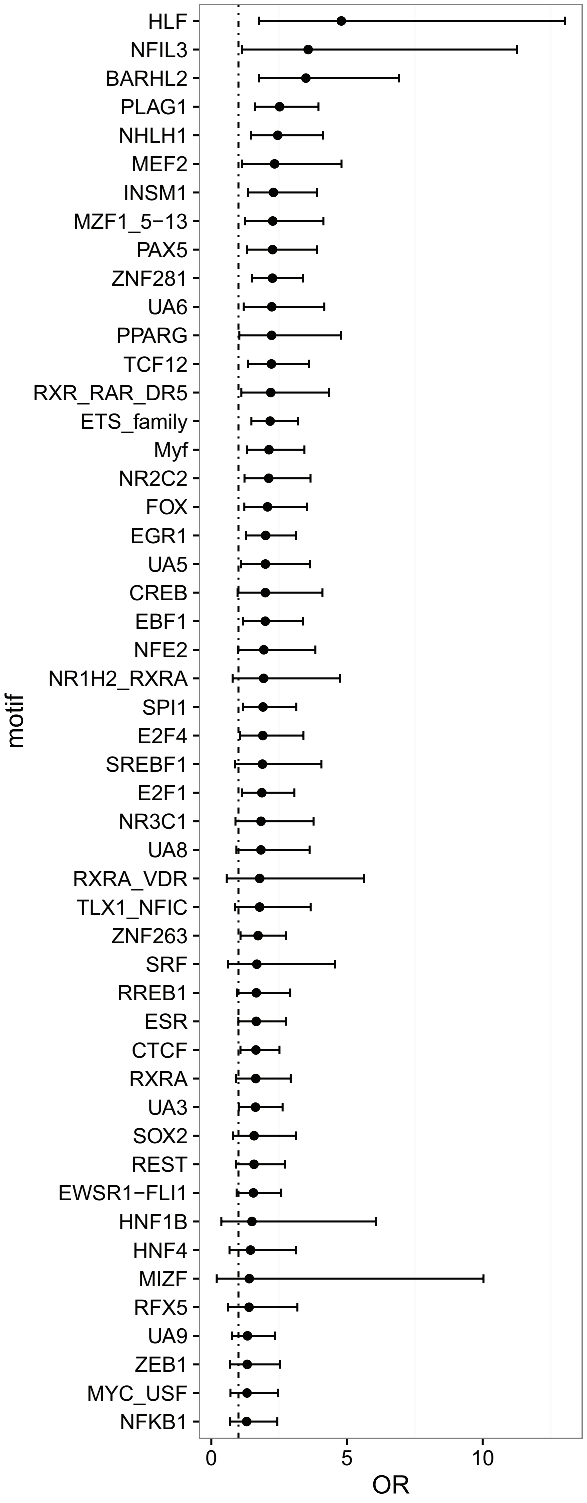


Figure 17. The plot shows the Odds Ratio (x axis) for the selected motifs. The range shows the 95% confidence interval for each Odds Ratio.

And for each relevant TF we will use the clinical data to analyze the survival curves for the 30% patients with higher expression of that TF versus the 30% with lower expression. The code below shows that analysis.

```r
TCGAsurvival_TFplot <- function(TF,mee, clinical, percentage = 0.3){

   # For the transcription factor, gets it getGeneID
   gene <- getGeneID(mee,symbol=TF)
   # Get the expression values for the genes.
   # (getExp is a ELMER function)
   exp <- getExp(mee,geneID=gene)

   # Get the names of the 30% patients with lower expression
   g1 <- names(sort(exp)[1:ceiling(length(exp) * percentage)])

   # Get the names of the 30% patients with higher expression
   g2 <- names(sort(exp, decreasing = T)[1:ceiling(length(exp) * percentage)])

   # get the data of only these patients
   idx <-  clinical$bcr_patient_barcode %in% substr(c(g1, g2),1,12)
   clinical <- clinical[idx,]
   # Create the labels for each sample
   clinical$tf_groups <- "high"
   low.idx <- clinical$bcr_patient_barcode %in%  substr(c(g1),1,12)
   clinical[low.idx,]$tf_groups <- "low"

   # Use TCGAbiolinks to create the survival curve
   TCGAanalyze_survival(clinical, "tf_groups",
                        legend=paste0(TF," Exp level"),
                        filename = paste0(TF,".pdf"))
}

# get clinical patient data for GBM samples
gbm_clin <- GDCquery_clinic("TCGA-GBM","Clinical")

# get clinical patient data for LGG samples
lgg_clin <- GDCquery_clinic("TCGA-LGG","Clinical")

# Bind the results, as the columns might not be the same,
# we will will plyr rbind.fill, to have all columns from both files
clinical <- plyr::rbind.fill(gbm_clin,lgg_clin)
# Call the function we created
TCGAsurvival_TFplot("ELF4",mee,clinical)
TCGAsurvival_TFplot("HMGA2",mee,clinical)
TCGAsurvival_TFplot("FOXD3",mee,clinical)
```

Listing 31. Survival analysis for samples with both lower and higher expression of regulatory TFs.

The Figure 20, shows that the samples with lower expression of some of these TFs have a better survival than those with higher expression.

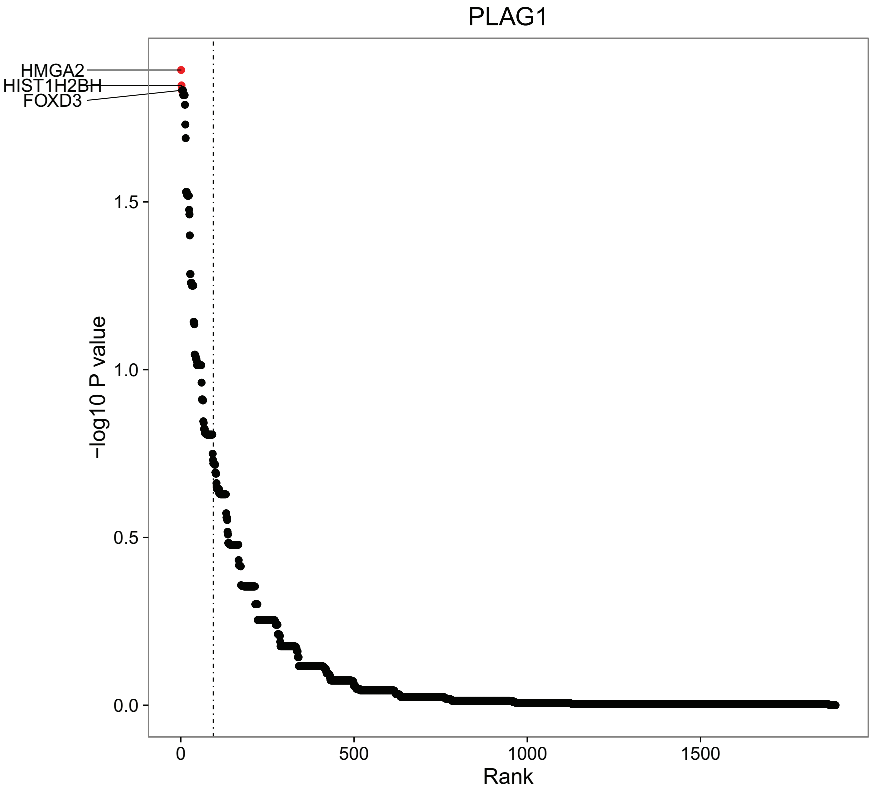


Figure 18. **TF ranking plots based on the score (*–log*(*Pvalue*)) of association between TF expression and DNA methylation of the FOX motif.** The dashed line indicates the boundary of the top 5% association scores and the TFs within this boundary were considered candidate upstream regulators. The top three associated TFs and the TF family members (dots in red) that are associated with that specific motif are labeled in the plot.

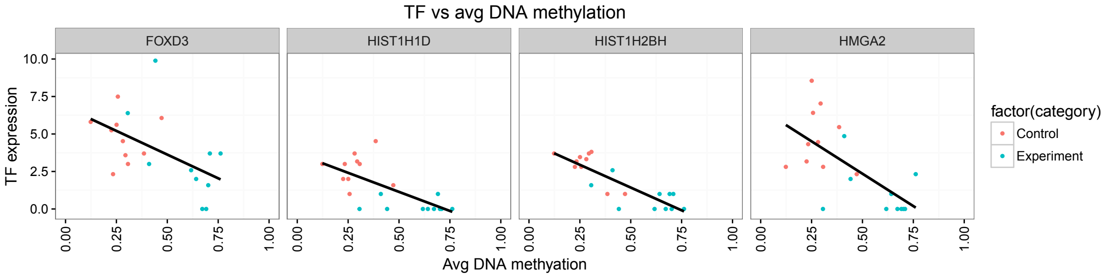


Figure 19. **Each scatter plot shows the average DNA methylation level of sites with the FOX motif plotted against the expression of the TFs FOXD3, HIST1H1D, HIST1H2BH, HMGA2 respectively.**

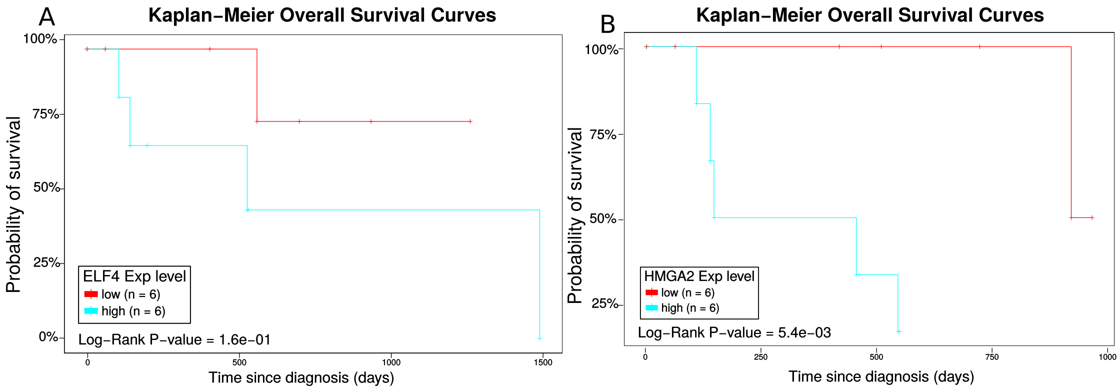


Figure 20. **A**) Survival plot for the 30% patients with high expression and low expression of ELF4 TF. **B**) Survival plot for the 30% patients with high expression and low expression of HGMA2 TF.

Conclusion

This workflow outlines how one can use specific Bioconductor packages for the analysis of cancer genomics and epigenomics data derived from the TCGA. In addition, we highlight the importance of using ENCODE and Roadmap data to inform on the biology of the non-coding elements defined by functional roles in gene regulation. We introduced TCGAbiolinks and RTCGAToolbox Bioconductor packages in order to illustrate how one can acquire TCGA specific data, followed by key steps for genomics analysis using GAIA package, for transcriptomic analysis using TCGAbiolinks, dnet, pathview packages and for DNA methylation analysis using TCGAbiolinks package. An inference of gene regulatory networks was also introduced by MINET package. Finally, we introduced Bioconductor packages AnnotationHub, ChIPSeeker, ComplexHeatmap, and ELMER to illustrate how one can acquire ENCODE/Roadmap data and integrate these data with the results obtained from analyzing TCGA data to identify and characterize candidate regulatory enhancers associated with cancer.

Data and software availability

This workflow depends on various packages from version 3.2 of the Bioconductor project, running on R version 3.2.2 or higher. It requires a number of software packages, including AnnotationHub, ChIPSeeker, ELMER, ComplexHeatmap, GAIA, rGADEM, MotIV, MINET, RTCGAToolbox and TCGAbiolinks. Version numbers for all packages used are in section “Session Information”. Listing 32 shows how to install all the required packages.

```
source("https://bioconductor.org/biocLite.R")
packages <- c("TCGAbiolinks","ELMER","gaia","ChIPseeker",
              "AnnotationHub", "ComplexHeatmap",
              "clusterProfiler", "RTCGAToolbox",
              "minet","biomaRt","pathview", "MotifDb",
              "MotIV","motifStack","rGADEM")
new.packages <- packages[!(packages %in% installed.packages()[,"Package"])]
if(length(new.packages)) biocLite(new.packages)
if(!require("dnet")) install.packages("dnet")
if(!require("circlize")) install.packages("circlize")
if(!require("VennDiagram")) install.packages("VennDiagram")
if(!require("c3net")) install.packages("c3net")
if(!require("pbapply")) install.packages("pbapply")
if(!require("gplots")) install.packages("gplots")
```

Listing 32. Installing packages

All data used in this workflow is freely available and some of them can be accessed using a R/Bioconductor package. There are three main sources of data: The NCI Genomic Data Commons (GDC), a supplementary data repository with processed datasets from the Roadmap Epigenomics Project and from The Encyclopedia of DNA Elements (ENCODE)[67] and the BioGRID database with physical and genetic interactions information. For the first, a summary of the data available can be seen in https://gdc-portal.nci.nih.gov/ and its data can be accessed using the R/Bioconductor TCGAbiolinks package. For the second, a summary of the data available can be found in this spread sheet and the data can be accessed using the R/Bioconductor AnnotationHub package. For the third, the data should be directly download from the data portal.

Session information

```
R version 3.3.0 (2016-05-03)
Platform: x86_64-pc-linux-gnu (64-bit )
Running under: Ubuntu 16.04.1 LTS

locale:
 [1] LC_CTYPE=pt_BR.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=pt_BR.UTF-8        LC_COLLATE=en_US.UTF-8
 [5] LC_MONETARY=pt_BR.UTF-8    LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=pt_BR.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=pt_BR.UTF-8 LC_IDENTIFICATION=C

```

```
attached base packages:
 [1] stats4    parallel  grid      stats     graphics  grDevices utils
 [8] datasets  methods   base

other attached packages:
 [1] rGADEM_2.20.0
 [2] seqLogo_1.38.0
 [3] BSgenome_1.40.1
 [4] rtracklayer_1.32.1
 [5] motifStack_1.16.2
 [6] ade4_1.7-4
 [7] grImport_0.9-0
 [8] XML_3.98-1.4
 [9] MotIV_1.28.0
[10] pathview_1.12.0
[11] biomaRt_2.28.0
[12] minet_3.30.0
[13] clusterProfiler_3.0.4
[14] DOSE_2.10.7
[15] AnnotationHub_2.4.2
[16] ChIPseeker_1.8.7
[17] ELMER_1.4.2
[18] ELMER.data_1.2.2
[19] Homo.sapiens_1.3.1
[20] TxDb.Hsapiens.UCSC.hg19.knownGene_3.2.2
[21] org.Hs.eg.db_3.3.0
[22] GO.db_3.3.0
[23] OrganismDbi_1.14.1
[24] GenomicFeatures_1.24.5
[25] AnnotationDbi_1.34.4
[26] IlluminaHumanMethylation450kanno.ilmn12.hg19_0.2.1
[27] minfi_1.18.2
[28] bumphunter_1.12.0
[29] locfit_1.5-9.1
[30] iterators_1.0.8
[31] foreach_1.4.3
[32] Biostrings_2.40.2
[33] XVector_0.12.1
[34] SummarizedExperiment_1.2.3
[35] GenomicRanges_1.24.2
[36] GenomeInfoDb_1.8.3
[37] IRanges_2.6.1
[38] S4Vectors_0.10.2
[39] lattice_0.20-33
[40] Biobase_2.32.0
[41] BiocGenerics_0.18.0
[42] ComplexHeatmap_1.10.2
[43] dnet_1.0.9
[44] supraHex_1.10.0
[45] hexbin_1.27.1
[46] igraph_1.0.1
[47] circlize_0.3.7
[48] pbapply_1.2-1
[49] gplots_3.0.1
[50] gaia_2.16.0
[51] readr_0.2.2
[52] downloader_0.4
```

```
[53] TCGAbiolinks_2.1.0

loaded via a namespace (and not attached):
  [1] SparseM_1.7                   ggthemes_3.2.0
  [3] prabclus_2.2-6                GGally_1.2.0
  [5] R.methodsS3_1.7.1             pkgmaker_0.22
  [7] tidyr_0.5.1                   ggplot2_2.1.0
  [9] knitr_1.13                    aroma.light_3.2.0
 [11] multcomp_1.4-6                R.utils_2.3.0
 [13] data.table_1.9.6              hwriter_1.3.2
 [15] KEGGREST_1.12.2               RCurl_1.95-4.8
 [17] GEOquery_2.38.4               doParallel_1.0.10
 [19] preprocessCore_1.34.0         cowplot_0.6.2
 [21] TH.data_1.0-7                 RSQLite_1.0.0
 [23] chron_2.3-47                  xml2_1.0.0
 [25] httpuv_1.3.3                  assertthat_0.1
 [27] BiocInstaller_1.22.3          DEoptimR_1.0-6
 [29] caTools_1.17.1                dendextend_1.2.0
 [31] Rgraphviz_2.16.0              DBI_0.4-1
 [33] geneplotter_1.50.0            reshape_0.8.5
 [35] stringdist_0.9.4.1            EDASeq_2.6.2
 [37] matlab_1.0.2                  dplyr_0.5.0
 [39] trimcluster_0.1-2             annotate_1.50.0
 [41] gridBase_0.4-7                robustbase_0.92-6
 [43] GenomicAlignments_1.8.4       mclust_5.2
 [45] mnormt_1.5-4                  cluster_2.0.4
 [47] ape_3.5                       genefilter_1.54.2
 [49] edgeR_3.14.0                  nlme_3.1-128
 [51] nnet_7.3-12                   diptest_0.75-7
 [53] sandwich_2.3-4                registry_0.3
 [55] affyio_1.42.0                 matrixStats_0.50.2
 [57] graph_1.50.0                  rngtools_1.2.4
 [59] base64_2.0                    Matrix_1.2-6
 [61] boot_1.3-18                   zoo_1.7-13
 [63] whisker_0.3-2                 GlobalOptions_0.0.10
 [65] png_0.1-7                     rjson_0.2.15
 [67] bitops_1.0-6                  R.oo_1.20.0
 [69] ConsensusClusterPlus_1.36.0   KernSmooth_2.23-15
 [71] doRNG_1.6                     shape_1.4.2
 [73] stringr_1.0.0                 qvalue_2.4.2
 [75] nor1mix_1.2-1                 coin_1.1-2
 [77] ShortRead_1.30.0              scales_0.4.0
 [79] GSEABase_1.34.0               magrittr_1.5
 [81] plyr_1.8.4                    gdata_2.17.0
 [83] zlibbioc_1.18.0               RColorBrewer_1.1-2
 [85] illuminaio_0.14.0             plotrix_3.6-3
 [87] KEGGgraph_1.30.0              lme4_1.1-12
 [89] Rsamtools_1.24.0              affy_1.50.0
 [91] MASS_7.3-45                   stringi_1.1.1
 [93] GOSemSim_1.30.3               latticeExtra_0.6-28
 [95] ggrepel_0.5                   tools_3.3.0
 [97] gridExtra_2.2.1               sjPlot_2.0.1
 [99] digest_0.6.9                  shiny_0.13.2
[101] quadprog_1.5-5                fpc_2.1-10
[103] Rcpp_0.12.6                   siggenes_1.46.0
[105] httr_1.2.1                    psych_1.6.6
[107] kernlab_0.9-24                effects_3.1-1
[109] sjstats_0.2.0                 colorspace_1.2-6
[111] rvest_0.3.2                   topGO_2.24.0
```

```
[113] splines_3.3.0                 RBGL_1.48.1
[115] multtest_2.28.0               flexmix_2.3-13
[117] xtable_1.8-2                  jsonlite_1.0
[119] nloptr_1.0.4                  UpSetR_1.2.2
[121] modeltools_0.2-21             R6_2.1.2
[123] htmltools_0.3.5               mime_0.5
[125] minqa_1.2.4                   BiocParallel_1.6.3
[127] DESeq_1.24.0                  class_7.3-14
[129] interactiveDisplayBase_1.10.3 beanplot_1.2
[131] codetools_0.2-14              mvtnorm_1.0-5
[133] tibble_1.1                    gtools_3.5.0
[135] openssl_0.9.4                 survival_2.39-5
[137] limma_3.28.17                 munsell_0.4.3
[139] DO.db_2.9                     GetoptLong_0.1.3
[141] sjmisc_1.8                    haven_0.2.1
[143] reshape2_1.4.1                gtable_0.2.0
```

Author contributions

HN conceived the study. HN, MC and GB provided direction on the design of the Transcriptomics, Genomics, master regulatory networks and DNA methylation workflows. TCS developed and tested sections “Experimental data”, “DNA methylation analysis”, “Motif analysis” and “Integrative analysis”. AC developed and tested section “Transcriptomic analysis”. CO developed and tested the section “Inference of gene regulatory networks”. FDA developed and tested section “Genomic analysis”. TCS, AC, CO, and FDA prepared the first draft of the manuscript. All authors were involved in the revision of the draft manuscript and have agreed to the final content. Also, AC, TS, CO, MC, GB and HN are authors of the TCGAbiolinks package and MC is author of the GAIA package.

Competing interests

No competing interests were disclosed.

Grant information

The project was supported by the São Paulo Research Foundation (FAPESP) (2015/02844-7 and 2016/01389-7 to T.C.S. & H.N. and 2015/07925-5 to H.N.), the BridgeIRIS project, funded by INNOVIRIS, Region de Bruxelles Capitale, Brussels, Belgium, and by GENomic profiling of Gastrointestinal Inflammatory-Sensitive CANcers (GENGISCAN), Belgian FNRS PDR (T100914F to G.B.). Funding for open access charge: São Paulo Research Foundation (FAPESP) (2015/07925-5).

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Acknowledgments

We are grateful to all the authors of the packages used in this article. Also, we would like to thank The GDC Support Team for the provided help, which was necessary in order to update the TCGAbiolinks package to use GDC API.

References

1. Hawkins RD, Hon GC, Ren B: **Next-generation genomics: an integrative approach.** *Nat Rev Genet.* 2010; **11**(7): 476–486.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**
2. Cancer Genome Atlas Research Network, Weinstein JN, Collisson EA, *et al.*: **The Cancer Genome Atlas Pan-Cancer analysis project.** *Nat Genet.* 2013; **45**(10): 1113–1120.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**
3. ENCODE Project Consortium: **A user’s guide to the encyclopedia of DNA elements (ENCODE).** *PLoS Biol.* 2011; **9**(4): e1001046.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**
4. Fingerman IM, McDaniel L, Zhang X, *et al.*: **NCBI Epigenomics: a new public resource for exploring epigenomic data sets.** *Nucleic Acids Res.* 2011; **39**(Database issue): D908–912.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**
5. Bernstein BE, Stamatoyannopoulos JA, Costello JF, *et al.*: **The NIH Roadmap Epigenomics Mapping Consortium.** *Nat Biotechnol.* 2010; **28**(10): 1045–1048.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**
6. Kannan L, Ramos M, Re A, *et al.*: **Public data and open source tools for multi-assay genomic investigation of disease.** *Brief Bioinform.* 2016; **17**(4): 603–15.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**
7. Wilks C, Cline MS, Weiler E, *et al.*: **The Cancer Genomics Hub (CGHub): overcoming cancer through the power of torrential data.** *Database (Oxford).* 2014; **2014**: pii: bau093.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**
8. Colaprico A, Silva TC, Olsen C, *et al.*: ***TCGAbiolinks*: an R/Bioconductor package for integrative analysis of TCGA data.** *Nucleic Acids Res.* 2016; **44**(8): e71.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

9. Huber W, Carey VJ, Gentleman R, *et al.*: **Orchestrating high-throughput genomic analysis with Bioconductor.** *Nat Methods.* 2015; **12**(2): 115–121.
PubMed Abstract | Publisher Full Text | Free Full Text

10. Morgan M, Obenchain V, Hester J, *et al.*: **Summarizedexperiment: Summarizedexperiment container.** R package version 1.1.0.
Reference Source

11. Zheng S, Cherniack AD, Dewal N, *et al.*: **Comprehensive Pan-Genomic Characterization of Adrenocortical Carcinoma.** *Cancer Cell.* 2016; **29**(5): 723–736.
PubMed Abstract | Publisher Full Text | Free Full Text

12. Cancer Genome Atlas Network: **Comprehensive molecular portraits of human breast tumours.** *Nature.* 2012; **490**(7418): 61–70.
PubMed Abstract | Publisher Full Text | Free Full Text

13. Cancer Genome Atlas Network: **Comprehensive molecular characterization of human colon and rectal cancer.** *Nature.* 2012; **487**(7407): 330–337.
PubMed Abstract | Publisher Full Text | Free Full Text

14. Ceccarelli M, Barthel FP, Malta TM, *et al.*: **Molecular Profiling Reveals Biologically Discrete Subsets and Pathways of Progression in Diffuse Glioma.** *Cell.* 2016; **164**(3): 550–563.
PubMed Abstract | Publisher Full Text | Free Full Text

15. Cancer Genome Atlas Network: **Comprehensive genomic characterization of head and neck squamous cell carcinomas.** *Nature.* 2015; **517**(7536): 576–582.
PubMed Abstract | Publisher Full Text | Free Full Text

16. Davis CF, Ricketts CJ, Wang M, *et al.*: **The somatic genomic landscape of chromophobe renal cell carcinoma.** *Cancer Cell.* 2014; **26**(3): 319–330.
PubMed Abstract | Publisher Full Text | Free Full Text

17. Cancer Genome Atlas Research Network: **Comprehensive molecular characterization of clear cell renal cell carcinoma.** *Nature.* 2013; **499**(7456): 43–49.
PubMed Abstract | Publisher Full Text | Free Full Text

18. Cancer Genome Atlas Research Network, Linehan WM, Spellman PT, *et al.*: **Comprehensive Molecular Characterization of Papillary Renal-Cell Carcinoma.** *N Engl J Med.* 2016; **374**(2): 135–145.
PubMed Abstract | Publisher Full Text | Free Full Text

19. Cancer Genome Atlas Research Network: **Comprehensive molecular profiling of lung adenocarcinoma.** *Nature.* 2014; **511**(7511): 543–550.
PubMed Abstract | Publisher Full Text | Free Full Text

20. Cancer Genome Atlas Research Network: **Comprehensive genomic characterization of squamous cell lung cancers.** *Nature.* 2012; **489**(7417): 519–525.
PubMed Abstract | Publisher Full Text | Free Full Text

21. Cancer Genome Atlas Research Network: **The Molecular Taxonomy of Primary Prostate Cancer.** *Cell.* 2015; **163**(4): 1011–1025.
PubMed Abstract | Publisher Full Text | Free Full Text

22. Cancer Genome Atlas Network: **Genomic Classification of Cutaneous Melanoma.** *Cell.* 2015; **161**(7): 1681–1696.
PubMed Abstract | Publisher Full Text | Free Full Text

23. Cancer Genome Atlas Research Network: **Comprehensive molecular characterization of gastric adenocarcinoma.** *Nature.* 2014; **513**(7517): 202–209.
PubMed Abstract | Publisher Full Text | Free Full Text

24. Cancer Genome Atlas Research Network: **Integrated genomic characterization of papillary thyroid carcinoma.** *Cell.* 2014; **159**(3): 676–690.
PubMed Abstract | Publisher Full Text | Free Full Text

25. Samur MK: **RTCGAToolbox: a new tool for exporting TCGA Firehose data.** *PLoS One.* 2014; **9**(9): e106397.
PubMed Abstract | Publisher Full Text | Free Full Text

26. Beroukhim R, Getz G, Nghiemphu L, *et al.*: **Assessing the significance of chromosomal aberrations in cancer: methodology and application to glioma.** *Proc Natl Acad Sci U S A.* 2007; **104**(50): 20007–20012.
PubMed Abstract | Publisher Full Text | Free Full Text

27. Mermel CH, Schumacher SE, Hill B, *et al.*: **GISTIC2.0 facilitates sensitive and confident localization of the targets of focal somatic copy-number alteration in human cancers.** *Genome Biol.* 2011; **12**(4): R41.
PubMed Abstract | Publisher Full Text | Free Full Text

28. Beroukhim R, Mermel CH, Porter D, *et al.*: **The landscape of somatic copy-number alteration across human cancers.** *Nature.* 2010; **463**(7283): 899–905.
PubMed Abstract | Publisher Full Text | Free Full Text

29. Morganella S, Pagnotta SM, Ceccarelli M: **Gaia: Genomic analysis of important aberrations.**
Reference Source

30. Gu Z: **Complexheatmap: Making complex heatmaps.** R package version 1.7.1.
Reference Source

31. Parman C, Halling C, Gentleman R: **affyQCReport: QC Report Generation for affyBatch objects.** R package version 1.50.0. 2016.
Reference Source

32. Risso D, Schwartz K, Sherlock G, *et al.*: **GC-content normalization for RNA-Seq data.** *BMC Bioinformatics.* 2011; **12**(1): 480.
PubMed Abstract | Publisher Full Text | Free Full Text

33. Bullard JH, Purdom E, Hansen KD, *et al.*: **Evaluation of statistical methods for normalization and differential expression in mRNA-seq experiments.** *BMC Bioinformatics.* 2010; **11**(1): 94.
PubMed Abstract | Publisher Full Text | Free Full Text

34. Luo W, Brouwer C: **Pathview: an R/Bioconductor package for pathway-based data integration and visualization.** *Bioinformatics.* 2013; **29**(14): 1830–1831.
PubMed Abstract | Publisher Full Text | Free Full Text

35. Margolin AA, Nemenman I, Basso K, *et al.*: **ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context.** *BMC Bioinformatics.* 2006; **7**(Suppl 1): S7.
PubMed Abstract | Publisher Full Text | Free Full Text

36. Faith JJ, Hayete B, Thaden JT, *et al.*: **Large-scale mapping and validation of *Escherichia coli* transcriptional regulation from a compendium of expression profiles.** *PLoS Biol.* 2007; **5**(1): e8.
PubMed Abstract | Publisher Full Text | Free Full Text

37. Meyer PE, Kontos K, Lafitte F, *et al.*: **Information-theoretic inference of large transcriptional regulatory networks.** *EURASIP J Bioinform Syst Biol.* 2007; **2007**(1): 79879.
PubMed Abstract | Publisher Full Text | Free Full Text

38. Altay G, Emmert-Streib F: **Inferring the conservative causal core of gene regulatory networks.** *BMC Syst Biol.* 2010; **4**(1): 132.
PubMed Abstract | Publisher Full Text | Free Full Text

39. Meyer PE, Lafitte F, Bontempi G: ***minet*: A R/Bioconductor package for inferring large transcriptional networks using mutual information.** *BMC Bioinformatics.* 2008; **9**(1): 461.
PubMed Abstract | Publisher Full Text | Free Full Text

40. Stark C, Breitkreutz BJ, Reguly T, *et al.*: **BioGRID: a general repository for interaction datasets.** *Nucleic Acids Res.* 2006; **34**(Database issue): D535–D539.
PubMed Abstract | Publisher Full Text | Free Full Text

41. Montojo J, Zuberi K, Rodriguez H, *et al.*: **GeneMANIA Cytoscape plugin: fast gene function predictions on the desktop.** *Bioinformatics.* 2010; **26**(22): 2927–2928.
PubMed Abstract | Publisher Full Text | Free Full Text

42. Phillips T: **The role of methylation in gene expression.** *Nat Educ.* 2008; **1**(1): 116.
Reference Source

43. Deaton AM, Bird A: **CpG islands and the regulation of transcription.** *Genes Dev.* 2011; **25**(10): 1010–1022.
PubMed Abstract | Publisher Full Text | Free Full Text

44. Robertson KD: **DNA methylation and human disease.** *Nat Rev Genet.* 2005; **6**(8): 597–610.
PubMed Abstract | Publisher Full Text

45. Marabita F, Almgren M, Lindholm ME, *et al.*: **An evaluation of analysis pipelines for DNA methylation profiling using the Illumina HumanMethylation450 BeadChip platform.** *Epigenetics.* 2013; **8**(3): 333–346.
PubMed Abstract | Publisher Full Text | Free Full Text

46. Droit A, Gottardo R, Robertson G, *et al.*: **rgadem: *De novo* motif discovery.** 2015.
Reference Source

47. Li L: **GADEM: a genetic algorithm guided formation of spaced dyads coupled with an EM algorithm for motif discovery.** *J Comput Biol.* 2009; **16**(2): 317–329.
PubMed Abstract | Publisher Full Text | Free Full Text

48. Ou J, Brodsky M, Wolfe S, *et al.*: **motifstack: Plot stacked logos for single or multiple DNA, RNA and amino acid sequence**. 2013.
Reference Source

49. Mercier E, Gottardo R: **Motiv: Motif identification and validation.** 2014.
Reference Source

50. Mahony S, Auron PE, Benos PV: **DNA familial binding profiles made easy: comparison of various motif alignment and clustering strategies.** *PLoS Comput Biol.* 2007; **3**(3): e61.
PubMed Abstract | Publisher Full Text | Free Full Text

51. Mahony S, Benos PV: **STAMP: a web tool for exploring**

DNA-binding motif similarities. *Nucleic Acids Res.* 2007; **35**(Web Server issue): W253–W258.
PubMed Abstract | Publisher Full Text | Free Full Text

52. Mercier E, Droit A, Li L, *et al.*: **An integrated pipeline for the genome-wide analysis of transcription factor binding sites from ChIP-Seq.** *PLoS One.* 2011; **6**(2): e16432.
PubMed Abstract | Publisher Full Text | Free Full Text

53. Shi X, Liu J, Huang J, *et al.*: **Integrative analysis of high-throughput cancer studies with contrasted penalization.** *Genet Epidemiol.* 2014; **38**(2): 144–151.
PubMed Abstract | Publisher Full Text | Free Full Text

54. Rhodes DR, Chinnaiyan AM: **Integrative analysis of the cancer transcriptome.** *Nat Genet.* 2005; **37**(Suppl): S31–S37.
PubMed Abstract | Publisher Full Text

55. Noushmehr H, Weisenberger DJ, Diefes K, *et al.*: **Identification of a CpG island methylator phenotype that defines a distinct subgroup of glioma.** *Cancer Cell.* 2010; **17**(5): 510–522.
PubMed Abstract | Publisher Full Text | Free Full Text

56. Siegmund KD: **Statistical approaches for the analysis of DNA methylation microarray data.** *Hum Genet.* 2011; **129**(6): 585–595.
PubMed Abstract | Publisher Full Text | Free Full Text

57. Tenenbaum D, Morgan M, Carlson M, *et al.*: **Annotationhub: Client to access annotationhub resources.** R package version 2.2.2.
Reference Source

58. Heintzman ND, Stuart RK, Hon G, *et al.*: **Distinct and predictive chromatin signatures of transcriptional promoters and enhancers in the human genome.** *Nat Genet.* 2007; **39**(3): 311–318.
PubMed Abstract | Publisher Full Text

59. Bernstein BE, Kamal M, Lindblad-Toh K, *et al.*: **Genomic maps and comparative analysis of histone modifications in human and mouse.** *Cell.* 2005; **120**(2): 169–181.
PubMed Abstract | Publisher Full Text

60. Bonasio R, Tu S, Reinberg D: **Molecular signals of epigenetic states.** *Science.* 2010; **330**(6004): 612–616.
PubMed Abstract | Publisher Full Text | Free Full Text

61. Peters AH, Kubicek S, Mechtler K, *et al.*: **Partitioning and plasticity of repressive histone methylation states in mammalian chromatin.** *Mol Cell.* 2003; **12**(6): 1577–1589.
PubMed Abstract | Publisher Full Text

62. Heintzman ND, Hon GC, Hawkins RD, *et al.*: **Histone modifications at human enhancers reflect global cell-type-specific gene expression.** *Nature.* 2009; **459**(7243): 108–112.
PubMed Abstract | Publisher Full Text | Free Full Text

63. Rada-Iglesias A, Bajpai R, Swigut T, *et al.*: **A unique chromatin signature uncovers early developmental enhancers in humans.** *Nature.* 2011; **470**(7333): 279–283.
PubMed Abstract | Publisher Full Text | Free Full Text

64. Creyghton MP, Cheng AW, Welstead GG, *et al.*: **Histone H3K27ac separates active from poised enhancers and predicts developmental state.** *Proc Natl Acad Sci U S A.* 2010; **107**(50): 21931–21936.
PubMed Abstract | Publisher Full Text | Free Full Text

65. Nishida H, Suzuki T, Kondo S, *et al.*: **Histone H3 acetylated at lysine 9 in promoter is associated with low nucleosome density in the vicinity of transcription start site in human cell.** *Chromosome Res.* 2006; **14**(2): 203–211.
PubMed Abstract | Publisher Full Text

66. Pekowska A, Anders S: **ChIP-seq analysis basics.** 2015; 6.
Reference Source

67. Roadmap Epigenomics Consortium, Kundaje A, Meuleman W, *et al.*: **Integrative analysis of 111 reference human epigenomes.** *Nature.* 2015; **518**(7539): 317–330.
Publisher Full Text | Free Full Text

68. Yu G, Wang LG, He QY, *et al.*: **ChIPseeker: an R/Bioconductor package for ChIP peak annotation, comparison and visualization.** *Bioinformatics.* 2015; **31**(14): 2382–3.
PubMed Abstract | Publisher Full Text

69. Giorgio E, Robyr D, Spielmann M, *et al.*: **A large genomic deletion leads to enhancer adoption by the lamin B1 gene: a second path to autosomal dominant adult-onset demyelinating leukodystrophy (ADLD).** *Hum Mol Genet.* 2015; **24**(11): 3143–54.
PubMed Abstract | Publisher Full Text | Free Full Text

70. Gröschel S, Sanders MA, Hoogenboezem R, *et al.*: **A single oncogenic enhancer rearrangement causes concomitant *EVI1* and *GATA2* deregulation in leukemia.** *Cell.* 2014; **157**(2): 369–381.
PubMed Abstract | Publisher Full Text

71. Sur IK, Hallikas O, Vähärautio A, *et al.*: **Mice lacking a *Myc* enhancer that includes human SNP rs6983267 are resistant to intestinal tumors.** *Science.* 2012; **338**(6112): 1360–1363.
PubMed Abstract | Publisher Full Text

72. Yao L, Berman BP, Farnham PJ: **Demystifying the secret mission of enhancers: linking distal regulatory elements to target genes.** *Crit Rev Biochem Mol Biol.* 2015; **50**(6): 550–573.
PubMed Abstract | Publisher Full Text | Free Full Text

73. Yao L, Shen H, Laird PW, *et al.*: **Inferring regulatory element landscapes and transcription factor networks from cancer methylomes.** *Genome Biol.* 2015; **16**(1): 105.
PubMed Abstract | Publisher Full Text | Free Full Text

Open Peer Review

Current Referee Status: ? ✓ ?

Version 2

Referee Report 06 February 2017

doi:10.5256/f1000research.11063.r19656

?

Charlotte Soneson
Institute of Molecular Life Sciences, University of Zurich (UZH), Zürich, Switzerland

This article provides a comprehensive overview of how to query, download and analyze data from large cancer consortia using R and Bioconductor. Overall, I find it a useful contribution to the field, and it illustrates many of the use cases that will likely be of interest to other researchers. However, as it currently is, I find that it suffers from a few problems that I think that the authors could relatively easily fix.

Most importantly, it should be asserted that all the code actually runs. For example, the code in listings 2 and 4 will not run since the functions are capitalized in the wrong way, and the code in listing 7 will not work since the "lastDate" variable is not defined. Also, the session info does not seem to correspond to the same session as was used in the workflow since, for example, the RTCGAToolbox package is not listed as being loaded.

Also, as the code is currently provided (in listings, with row numbers), I could not find a way to directly cut and paste it into an R session. Given the amount of code included in this workflow, it would be good if that could be resolved.

Since the workflow contains a lot of code, it would be useful to add a bit more comments, explaining what each code chunk, and each part of a chunk, is doing. Moreover, the different code chunks have different formatting styles, and it would be easier to read if this was harmonized. For example, comments are formatted differently, spaces are used inconsistently, and even within a listing (e.g., listing 2) the function calls use different formatting (with or without named arguments). Finally, not all code chunks are in "listings", and it is not clear to me why.

Throughout the text, the authors should also make sure to be consistent in terms of capitalization. In several places there are differences in capitalization between the text and the code (e.g., GDCquery_Maf()), or between different places in the text. There are also discrepancies in whether function names are italicized or not. The function GDCquery_clinic is referred to as GDCquery_clinical in the text. Within one section, the AAIC is spelled out as "Array Array", "Array-array" and "array-array".

Just after listing 2, it is stated that you can use "rowRanges to gets the range of values in each row" of the SummarizedExperiment object. This should be clarified, since it returns a GRanges object describing the features in the rows.

The "p-value corrected FDR" should also be replaced by e.g. "corrected (or adjusted) p-value" or "FDR".

Bioconductor should be capitalized.

The "broadinstitute website" should be changed to "Broad Institute website" and include a link.

The "nce" is not defined as far as I could see, and the sentence just before the "Retrieving known interactions" seems unfinished. It is not clear to me how the information that "There are 3,941 unique interactions between the 2,901 differentially expressed genes" was obtained, since the function in the code chunk above is not run until later.

It would be easier to get an overview of the text of the headers were more consistent. For example, the gene regulatory network section has subheaders that are larger than the main section header.

Finally, as mentioned by one of the other reviewers, the paper would benefit from modifications to the text in order to make it more readable. There are many grammatical errors, long and complicated sentences, and sentences that are unfinished and thus difficult to understand.

I believe that if the authors can make sure that all the code in the workflow runs as expected, and the text and code is clarified and harmonized throughout the manuscript, this can be a very valuable addition to the literature.

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Competing Interests: No competing interests were disclosed.

Version 1

Referee Report 21 September 2016

doi:10.5256/f1000research.9601.r15858

✓ **Elena Papaleo**

Computational Biology Laboratory, Unit of Statistics, Bioinformatics and Registry, Danish Cancer Society Research Center, Copenhagen, Denmark

As it has been pointed out already by the first reviewer, it is important to verify that the pipeline is updated according to the data migration to the GCD server.

Apart from this, I find this manuscript very nice and the workflow an important contribution in the field, allowing to make accessible large datasets with genomics profiling of cancer patients to the community and even for not advanced R-users. I would like to compliment with the authors to make such a comprehensive workflow available.

Some minor points before final publication is indexed:

- The main issue with this manuscript is that it will need an extensive revision by a native English speaker since some sentences are hard to read and very often they risk to convey the wrong message or to be misinterpreted. One example from the abstract, "we provide a series of biologically focused integrative downstream analyses of different molecular data" includes to many adjectives. Few lines below "we provide a workplan to identify candidate biologically relevant functional epigenomic elements associated with cancer", this sentence does not sound that right in English.

- Be also careful in the style use to call the software or Bioconductor packages in the text since sometimes they are italics and other times they are not.
- There is also a lot of redundancy on some of the statements in the introduction that if removed will improve the readability. For example the sentence at the end of the first paragraph of the introduction starting with " Here we describe a comprehensive workflow that integrates many bioconductor packages..." is probably not needed since the same concept is provided at the end of the introduction in the same page.
- In the Introduction when TCGA data are illustrated I would also mention metastatic samples (they are there, isn't it? or may be I don't recall correctly...)
- I suggest to change the title of the first section of the methods from 'Experimental data' to 'Access to the data' or something similar so that it will better reflect the content of the paragraph.
- I felt that sometimes the manuscript sounds more a user guide than an article, so I would suggest taking as an example the section on methylation to improve the discussion of the analyses outcome and aims in the other paragraphs.
- Also please be careful to make things accessible to any readers for example at page 4, the authors introduce the GISTIC data without explaining them in details.
- The sentence at page 4 "the data used in this workflow are published data and freely available" might be redundant so the authors could remove it.
- Page 4 summarizedExperiment object is missing the reference to the original paper about summarizedExperiment format.
- Page 5: in the text the authors illustrate assay, rowRanges and colData but in the example the order is different, i.e. assay, colData, rowRanges - I would suggest to keep consistency in the order between main text and examples (it also holds for other places in the paper).
- I am not sure I can understand the sentence at page 5, "the users should use all the follow up files in your analyses, not just the latest version".
- Page 6: the sentence beginning with "Finally, the Cancer Genome Atlas (TCGA) Research Network has reported integrated genome-wide studies of various diseases, in what is called 'Pan Cancer'..." I am not sure fits within the paragraph where the main focus is on subtypes. It sounds a bit confusing for the reader.
- Not sure that Listing 5 also fits there... why not mention it before together with the others summ.exp. accessors.

- In writing it is important that the authors really pay attention to convey the message that it is an integrative approach ... at the very first reading of the manuscript I felt that it was more a list of tools. and the integrative part of it was missing

- Page 18: a reference is missing for the AAIC method and an explanation to the reader about the choice of the 0.6 correlation cutoff.

- In the list at page 18 with the normalization, I believe that there are some repetitions (i.e. global scaling and full-quantile are appearing twice, also there are some typos).

- Page 19, I might have missed something but I cannot find in maintext any explanation of the DEA pipeline and more comments would be needed there for a new user to really appreciate it and its importance.

- Page 20 on the second line I believe a). is missing.

- On PEA, I was wondering why for example the R package for Reactome was not integrated here and if the authors could comment on this.

- Page 23, why for protein-protein interactions only BioGrid is used and not for example a more comprehensive resource such as I2D (Interologous Interaction Database).

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Competing Interests: No competing interests were disclosed.

Author Response 28 Dec 2016

Tiago Chedraoui Silva, University of São Paulo, Brazil

Dear Elena Papaleo,

First, we would like to thank you for your review and for providing a detailed feedback for our workflow. We have made several changes in this new version (2) of the workflow. The changes and responses of all authors to these points are described below:

- "As it has been pointed out already by the first reviewer, it is important to verify that the pipeline is updated according to the data migration to the GCD server."
  - TCGAbiolinks package was entirely redesigned to query, download and prepare data from the GDC NCI data portal (https://gdc.nci.nih.gov/). All the codes are working and we will submit very soon the workflow to Bioconductor. For the moment, the RMarkdown can be found in our Github repository (https://github.com/BioinformaticsFMRP/f1000_TCGA_Workflow/blob/master/f1000.Rmd).

- "Be also careful in the style use to call the software or Bioconductor packages in the text since sometimes they are italics and other times they are not."
  - We applied the following pattern: name of the packages normal with a link to the package, name of functions and objects in italics.

- "I felt that sometimes the manuscript sounds more a user guide than an article, so I would suggest taking as an example the section on methylation to improve the discussion of the analyses outcome and aims in the other paragraphs."
    - Our main focus was on using the tools (in a reasonable time) rather than analyzing the results. The analysis itself can be verified with some articles of our group, for example, the DNA methylation analysis was described in Ceccarelli, Michele, *et al.* " Molecular profiling reveals biologically discrete subsets and pathways of progression in diffuse glioma." Cell 164.3 (2016): 550-563 as well as the analysis performed by the ELMER tool which is described in Yao, Lijing, *et al.* "Inferring regulatory element landscapes and transcription factor networks from cancer methylomes." Genome biology 16.1 (2015): 1.

- "In writing it is important that the authors really pay attention to convey the message that it is an integrative approach ... at the very first reading of the manuscript I felt that it was more a list of tools. and the integrative part of it was missing"
    - We agree with you. The article starts with specific analysis, which might cover 65% of the content and unfortunately it might lead to the feeling of the missing integrative approach. Also, we decided not to go much deeper into the analyzes because they already exist in the referenced articles and we focus more on the execution of the analyzes with the tools of the bioconductor project, something that was not covered by the cited articles, which might also give the feeling of a "list of tools". But, the integrative analysis is shown strongly in the last two sections. We may highlight that ELMER, for example, uses DNA methylation, gene expression, histone marks, motif information and clinical data for its analysis. Also, after the DNA methylation analysis available in TCGAbiolinks package, we do a motif analysis on those regions and integrated the roadmap histone marks data to evaluate them. Finally, another integrative analysis presented in the TCGAbiolinks package is the starburst plot which integrates the differential expression analysis results with the DNA methylation results and we try to identify if the nearest gene might have been affected by the change of the DNA methylation.

- "Page 18: a reference is missing for the AAIC method and an explanation to the reader about the choice of the 0.6 correlation cutoff."
    - Array-array intensity correlation plot (AAIC) is a re-adaptation of a affyQCReport::correlationPlot working with data from gene expression of class RangedSummarizedExperiment as output of GDCprepare.Reference from https://www.bioconductor.org/packages/devel/bioc/vignettes/affyQCReport/inst/doc/affyQ . AAIC shows an heat map of the array-array Spearman / Pearson rank correlation coefficients. The arrays are ordered using the phenotypic data (if available) in order to place arrays with similar samples adjacent to each other. Self-self correlations are on the diagonal and by definition have a correlation coefficient of 1.0. Data from similar tissues or treatments will tend to have higher coefficients. This plot is useful for detecting outliers, failed hybridizations, or mistracked samples.
    - The 0.6 threshold came out from unsupervised hierarchical clustering with ward methodology that obtained distinct groups of samples and first one had pearson correlation lower than 0.6 considering them as possible outliers.

- "Page 19, I might have missed something but I cannot find in maintext any explanation of the DEA pipeline and more comments would be needed there for a new user to really appreciate it and its importance."
    - For DEA pipeline we used the TCGAanalyze_DEA function from our package TCGAbiolinks, that allows user to perform Differentially expression analysis (DEA), using edgeR package to identify differentially expressed genes (DEGs). It is possible to do a two-class analysis.
    - TCGAanalyze_DEA performs DEA using following functions from edgeR:
        - edgeR::DGEList converts the count matrix into an edgeR object.
        - edgeR::estimateCommonDisp each gene gets assigned the same dispersion estimate.
        - edgeR::exactTest performs pair-wise tests for differential expression between two groups.
        - edgeR::topTags takes the output from exactTest(), adjusts the raw p-values using the False Discovery Rate (FDR) correction, and returns the top differentially expressed genes.
- "On PEA, I was wondering why for example the R package for Reactome was not integrated here and if the authors could comment on this."
    - Thank you for this suggestion we are working to integrate Reactome as source of genes annotated within pathways.
    - Our PEA provided only one plot to show Gene ontologies and pathways enriched by a list of genes to have an overview of top biological functions and pathways altered by molecules inside the gene signature. We focused mainly on Bioconductor packages, but in particular we are interested to add wrapping to some functionalities of the ReactomePA package.

- "Page 23, why for protein-protein interactions only BioGrid is used and not for example a more comprehensive resource such as I2D (Interologous Interaction Database)."
    - Different sources for protein-protein interactions are available. We used BioGrid as an example, but the users can chose their preferred database such as I2D

Regarding the other points:

- We revised the text to correct links, references and remove redundancies.
- We added metastatic samples as one of the available sample types
- We changed the name of the section 'Experimental data' to 'Access to the data'
- We included a paragraph introducing GISTIC data

Competing Interests: No competing interests were disclosed.

Referee Report 18 July 2016

doi:10.5256/f1000research.9601.r14695

?

Kyle Ellrott

Oregon Health & Science University, Portland, OR, USA

This review comes at a very inopportune moment. The entire software pipeline is based on the TCGAbiolinks tool kit, which downloads files from the TCGA DCC service. Unfortunately, just as this paper was being sent for review, the NCI began their migration to the GDC service. This means that all of the data services at DCC TCGA data portal (https://tcga-data.nci.nih.gov) are no longer active, and users are being directed to the GDC NCI data portal (https://gdc.nci.nih.gov/). This means that the TCGAbiolinks tool kit is broken. My attempts to run some of the examples listed in the paper where stopped by this issue. Many parts of the TCGAbiolinks will have to be fixed and/or re-written to deal with this change. This isn't the fault of the authors, but it does render the example and pipelines described in the paper inoperable.

There are a few technical issues that need to be addressed. The package 'RTCGAtoolbox' is mentioned, but actually its name is 'RTCGAToolbox'. (lower-case T to upper-case), and because the bioconductor website is case-sensitive the provided URL (http://bioconductor.org/packages/RTCGAtoolbox/) doesn't work. On page 3, 'TCGA Wikipedia' should be 'TCGA Wiki'. In the methods section on page 3, where the various data levels of TCGA are explained, on the most important aspects about the different levels is of access requirements. Data levels 1 and 2 cannot be accessed without dbGap, while level 3 and 4 are generally available for public access.

There is also the question of whether or not this paper fits the criteria of the Software Tools Article guidelines. The authors have presented a loose set of examples that utilize various existing, and previously published, tools. In the guidelines of F1000Research's Software Tool Articles, the criteria for a paper is "We welcome the description of new software tools. A Software Tool Article should include the rationale for the development of the tool and details of the code used for its construction." The authors refer to the code included in the paper as a workflow, but reads more like a series of point example that the reader can copy and alter for their own research. In introducing a new piece of software, one expects that the authors of the article are responsible for the software being presented. And while this article mentioned an extensive number of R packages, I believe only one of them, TCGAbiolinks, was written by the authors, and was published last year. While the analysis they present is very detailed and covers an expansive number of topics, I'm not sure if this should be classified as 'Software Tool Article'. On my first read through, my assumption was that the authors were responsible for all of the tools mentioned in the abstract. And this wasn't clarified in the text. For example, in the conclusion section on page 43, they include the phrase 'We introduced TCGAbiolinks and RTCGAtoolbox bioconductor packages in order to illustrate how one can acquire TCGA specific data', it wasn't until later, that I realized that the author of the RTCGAToolbox wasn't in the author list of the paper.

Of course, this is at the discretion of the editors. I would feel that this material would better be presented as a review article covering the different methods of TCGA data analysis, with better notation and attributions about the source of different pieces of software.

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Competing Interests: No competing interests were disclosed.

Author Response 28 Dec 2016

Tiago Chedraoui Silva, University of São Paulo, Brazil

Dear Kyle Ellrott,

Thank you for your comments and suggestions. We made several changes in the version 2 of the workflow, some of the changes and answers to your points are below:

- TCGAbiolinks package was entirely redesigned to query, download and prepare data from the GDC NCI data portal (https://gdc.nci.nih.gov/). All the codes are working and we will submit very soon the workflow to Bioconductor. For the moment, the RMarkdown can be found in our Github repository (https://github.com/BioinformaticsFMRP/f1000_TCGA_Workflow/blob/master/f1000.Rmd).

- We also highlighted the difference between open (TCGA level 3 and 4 data) and controlled data (TCGA level 1 and 2 data) and we added some useful sources that can help the user request access to the controlled data.

- Despite the description of a software article, we decided by this type of article simply because there were already other workflows with that choice and we did not find any other possibility that best described this type of article. In addition, our main focus was on using the tools (in a reasonable time) rather than analyzing the results. The analysis itself can be verified with some articles of our group, for example, the DNA methylation analysis was described in **Ceccarelli, Michele, *et al*. "Molecular profiling reveals biologically discrete subsets and pathways of progression in diffuse glioma." Cell 164.3 (2016): 550-563** as well as the analysis performed by the ELMER tool which is described in **Yao, Lijing, *et al.* "Inferring regulatory element landscapes and transcription factor networks from cancer methylomes." Genome biology 16.1 (2015): 1.**

- We added to "Author contributions" that we are the authors from the R/Bioconductor packages TCGAbiolinks and GAIA. Also, I'm working on a new version of the ELMER package.

- We corrected links and references such as RTCGAToolbox.

Competing Interests: No competing interests were disclosed.