





SOFTWARE TOOL ARTICLE

TCGA Workflow: Analyze cancer genomics and epigenomics data using Bioconductor packages [version 1; peer review: 1 approved, 1 approved with reservations]

Tiago C. Silva ¹, Antonio Colaprico^{2,3}, Catharina Olsen^{2,3}, Fulvio D'Angelo^{4,5}, Gianluca Bontempi²⁻⁴, Michele Ceccarelli⁶, Houtan Noushmehr ¹

¹Department of Genetics, Ribeirao Preto Medical School, University of Sao Paulo, Ribeirao Preto, Brazil

²Interuniversity Institute of Bioinformatics in Brussels, Brussels, Belgium

³Machine Learning Group, ULB, Brussels, Belgium

⁴Department of Science and Technology, University of Sannio, Benevento, Italy

⁵Biogem, Istituto di Ricerche Genetiche Gaetano Salvatore, Avellino, Italy

⁶Qatar Computing Research Institute (QCRI), HBKU, Doha, Qatar

v1 First published: 29 Jun 2016, 5:1542
<https://doi.org/10.12688/f1000research.8923.1>




Latest published: 28 Dec 2016, 5:1542
<https://doi.org/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 comprehensive tool that provides a complete integrative analysis harnessing 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 downstream 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 and by using Roadmap and ENCODE data, we provide a workplan to identify candidate 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](#),

Open Peer Review

Approval Status   

	1	2	3
version 2 (revision) 28 Dec 2016			 view
version 1 29 Jun 2016	 view	 view	

1. **Kyle Ellrott**, Oregon Health & Science University, Portland, USA
2. **Elena Papaleo**, Danish Cancer Society Research Center, Copenhagen, Denmark
3. **Charlotte Sonesson** , University of Zurich (UZH), Zürich, Switzerland

Any reports and responses or comments on the article can be found at the end of the article.

GAIA, MINET, RTCGAtoolbox, TCGAbiolinks.

Keywords

Epigenomics, Genomics, Cancer, non-coding, TCGA, ENCODE, Roadmap, Bioinformatics



This article is included in the **Bioconductor** gateway.



This article is included in the **Bioinformatics** gateway.

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

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.

Copyright: © 2016 Silva TC *et al.* This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

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 1; peer review: 1 approved, 1 approved with reservations]** F1000Research 2016, 5:1542 <https://doi.org/10.12688/f1000research.8923.1>

First published: 29 Jun 2016, 5:1542 <https://doi.org/10.12688/f1000research.8923.1>

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. In order to get insight in the 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, the understanding of 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. Here we describe a comprehensive workflow that integrates many Bioconductor packages in order to analyze and integrate the molteplicity of molecular observation layers in large scale cancer dataset.

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 that include: exome (variant analysis), single nucleotide polymorphism (SNP), DNA methylation, transcriptome (mRNA), microRNA (miRNA), proteome and clinical information. Sample types available at TCGA are: primary solid tumors, recurrent solid tumors, blood derived normal and tumor, and solid tissue normal².
- **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 cells³.
- **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 to cancer state at the molecular level and, importantly, it correlates these phenotypes with tissue of origins.

Although there exists a wealth of possibilities⁶ in accessing cancer associated data, **bioconductor** represents 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. Our workflow presents several **bioconductor** packages to work with genomic and epigenomics data.

Methods

Experimental data

TCGA data are accessible via the **TCGA data portal** and the **Broad Institute's GDAC Firehose**. The data are provided as different levels or tiers: Level 1 (Raw Data), Level 2 (Processed Data), Level 3 (Segmented or Interpreted Data) and Level 4 (Region of Interest Data). While the TCGA data portal provides level 1 to 3 data, Firehose only provides level 3 and 4. An explanation of the different levels can be found at **TCGA Wikipedia**. The data provided by TCGA data portal 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 TCGAAbiolinks & RTCGAtoolbox to download clinical, genomics, transcriptomics, epigenomics data, as well as subtype information and GISTIC results (identified genes targeted by somatic copy-number alterations (SCNAs) that drive cancer growth). Just to reiterate, the data used in this workflow are published data and freely available.

Downloading data from TCGA data portal. The Bioconductor package [TCGAAbiolinks](#)⁷ has three main functions **TCGAquery**, **TCGAdownload** and **TCGAprepare** that should sequentially be used to respectively search, download and load the data as an R object.

TCGAquery searches in a pre-processed TCGA database and returns a summary table with the found files, samples, version and other useful information. The most important *TCGAquery* arguments are *tumor* which receives one or multiple tumor types (USC, LGG, SKCM, KICH, CHO, etc), *platform* which receives the platform (HumanMethylation27, Genome_Wide_SNP_6, IlluminaHiSeq_RNASeqV2, etc), *version* which receives the version of the data to be downloaded if the user wants an older version and *samples* which receives a list of TCGA barcodes (ex. "TCGA-CS-4938") to filter the search results. A complete list of possible entries for arguments can be found in the [TCGAAbiolinks vignette](#). Lines 6 and 13 of [Listing 1](#) show an example of this function.

After searching, the user will be able to download the data with *TCGAdownload*. An important feature of this function is the ability to filter the data using the arguments *type* if the user wants to specify file tumor type and *samples* if user wants to specify samples (list of TCGA barcodes). For example, lines 15 and 18 of [Listing 1](#) are used to select a specific tumor type to download and prepare the data respectively. The platforms and their possible inputs for the type argument is shown below:

- **RNASeqV2:** junction_quantification, rsem.genes.results, rsem.isoforms.results, rsem.genes.normalized_results, rsem.isoforms.normalized_results, bt.exon_quantification
- **RNASeq:** exon.quantification, spljxn.quantification, gene.quantification
- **genome_wide_snp_6:** hg18.seg, hg19.seg, nocnv_hg18.seg, nocnv_hg19.seg
- **IlluminaHiSeq_miRNASeq:** hg19.mirbase20.mirna.quantification, hg19.mirbase20.isoform.quantification, mirna.quantification, isoform.quantification

Finally, *TCGAprepare* transforms the downloaded data into a [summarizedExperiment](#) object or a data frame. If `summarizedExperiment` is set to TRUE, TCGAAbiolinks will add metadata to the object in order to help the user when working with the data. Also, if the user sets the argument `add.subtype` to TRUE the `summarizedExperiment` will receive subtype information defined by The Cancer Genome Atlas (TCGA) Research Network reports (the full list of papers can be seen in [TCGAquery_subtype section](#) in TCGAAbiolinks vignette), Likewise, if the user sets the argument `add.clinical` to TRUE the `summarizedExperiment` will receive clinical information. Lines 8–11 and 18–22 of [Listing 1](#) illustrates this function.

```

1 library(TCGAAbiolinks)
2
3 # Download the DNA methylation data: HumanMethylation450 LGG and GBM.
4 path <- "."
5
6 query.met <- TCGAquery(tumor = c("LGG","GBM"),"HumanMethylation450", level = 3)
7 TCGAdownload(query.met, path = path )
8 met <- TCGAprepare(query = query.met, dir = path,
9                   add.subtype = TRUE, add.clinical = TRUE,
10                  summarizedExperiment = TRUE,
11                  save = TRUE, filename = "lgg_gbm_met.rda")
12
13 # Download the expression data: IlluminaHiSeq_RNASeqV2 LGG and GBM.
14 query.exp <- TCGAquery(tumor = c("lgg","gbm"), platform = "IlluminaHiSeq_
RNASeqV2", level = 3)
15
16 TCGAdownload(query.exp,path = path, type = "rsem.genes.normalized_results")

```

```

17
18 exp <- TCGAprepare(query = query.exp, dir = path,
19                   summarizedExperiment = TRUE,
20                   add.subtype = TRUE, add.clinical = TRUE,
21                   type = "rsem.genes.normalized_results",
22                   save = T,filename = "lgg_gbm_exp.rda")

```

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

If a summarizedExperiment object was chosen, the data can be accessed with three different accessors: *assay* for the data information, *rowRanges* to get the range of values in each row and *colData* to get the sample information (patient, batch, sample type, etc)^{8,9}. An example is shown in [Listing 2](#).

```

1 library(summarizedExperiment)
2 # get expression matrix
3 data <- assay(exp)
4
5 # get sample information
6 sample.info <- colData(exp)
7
8 # get genes information
9 genes.info <- rowRanges(exp)

```

Listing 2. summarizedExperiment accessors

Clinical data can be obtained using the function *TCGAquery_clinical* which can be used as described in [Listing 3](#). This function has three arguments *tumor*, *clinical_data_type* and *samples*. The *clinical_data_type* argument is always required and should be accompanied by at least one of the other two parameters. Examples for the argument *clinical_data_type* are: “clinical_drug”, “clinical_patient”, and “clinical_radiation” (a complete list and description can be found in the section ‘Working with clinical data.’ of the TCGAbiolinks [vignette](#)).

An important note about the clinical data is that follow-up data for TCGA patients are contained in the ‘clinical_follow_up’ files for each cancer type and to obtain all available disease progression information, the users should use all the follow_up files in your analyses, not just the latest version.

```

1 # get clinical patient data for GBM samples
2 gbm_clin <- TCGAquery_clinic("gbm","clinical_patient")
3
4 # get clinical patient data for LGG samples
5 lgg_clin <- TCGAquery_clinic("lgg","clinical_patient")
6
7 # Bind the results, as the columns might not be the same,
8 # we will will plyr::rbind.fill , to have all columns from both files
9 clinical <- plyr::rbind.fill(gbm_clin , lgg_clin)
10
11 # Other clinical files can be downloaded,
12 # Use ?TCGAquery_clinic for more information
13 clin_radiation <- TCGAquery_clinic("lgg","clinical_radiation")
14
15 # Also, you can get clinical information from different tumor types.
16 # For example sample 1 is GBM, sample 2 and 3 are TGCT
17 data <- TCGAquery_clinic(clinical_data_type = "clinical_patient",
18                         samples = c("TCGA-06-5416-01A-01D-1481-05",
19                                     "TCGA-2G-AAEW-01A-11D-A42Z-05",
20                                     "TCGA-2G-AAEX-01A-11D-A42Z-05"))

```

Listing 3. Downloading clinical data with TCGAbiolinks

Mutation information is stored in Mutation Annotation Format (MAF) files which contain different mutation types (somatic or germline) and states (validated or putative). A summary of all the Mutation Annotation Format (MAF) can be accessed at [TCGA wiki](#). To download these data using [TCGAbiolinks](#), `TCGAquery_maf` function is provided. It will download the non-obsolete tables from TCGA wiki, remove the protected entries and ask the user which file s/he wants to download (see [Listing 4](#)). It will then download and return a data frame with the data.

```

1 > mutation <- TCGAquery_maf(tumor = "lgg")
2 Getting maf tables
3 Source: https://wiki.nci.nih.gov/display/TCGA/TCGA+MAF+Files
4 We found these maf files below:
5
6                                     MAF.File.Name
7 2 hgsc.bcm.edu_LGG.IlluminaGA_DNASeq.1.somatic.maf
8
9 3 LGG_FINAL_ANALYSIS.aggregated.capture.tcga.uuid.curated.somatic.maf
10
11                                     Archive.Name Deploy.Date
12 2 hgsc.bcm.edu_LGG.IlluminaGA_DNASeq_automated.Level_2.1.0.0 10-DEC-13
13 3 broad.mit.edu_LGG.IlluminaGA_DNASeq_curated.Level_2.1.3.0 24-DEC-14
14 Please, select the line that you want to download: 3

```

Listing 4. Downloading mutation data with TCGAbiolinks

Finally, the Cancer Genome Atlas (TCGA) Research Network has reported integrated genome-wide studies of various diseases, in what is called ‘PanCan’. `TCGAqueryPrepare` function can automatically import the subtypes defined by these reports and incorporate them into a `summarizedExperiment` object. The subtypes can also be accessed using `TCGAquery_subtype` function. The subtypes include: LGG¹⁰, GBM¹⁰, STAD¹¹, BRCA¹², READ¹³, COAD¹³ and LUAD¹⁴.

```

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

```

Listing 5. summarizedExperiment accessors

Downloading data from Broad TCGA GDAC. The Bioconductor package [RTCGAtoolbox](#)¹⁵ 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 are listed in `getFirehoseDatasets` function.
- `runDate` - Stdtdata 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 6](#). The `getData` function allow users to access the downloaded data (see lines 22–24 of [Listing 6](#)) as a `S4Vector` object.

```

1 library(RTCGAToolbox)
2
3 # Get the last run dates
4 lastRunDate <- getFirehoseRunningDates()[1]
5 lastAnalyseDate <- getFirehoseAnalyzeDates(1)
6
7 # get DNA methylation data, RNAseq2 and clinical data for LGG
8 lgg.data <- getFirehoseData(dataset = "LGG",
9                             gistic2_Date = getFirehoseAnalyzeDates(1),
runDate = lastRunDate,

```

```

10             Methylation = TRUE, RNAseq2_Gene_Norm = TRUE,
Clinic = TRUE,
11             Mutation = T,
12             fileSizeLimit = 10000)
13
14 # get DNA methylation data , RNAseq2 and clinical data for GBM
15 gbm.data <- getFirehoseData(dataset = "GBM",
16             runDate = lastDate, gistic2_Date =
getFirehoseAnalyzeDates(1),
17             Methylation = TRUE, Clinic = TRUE, RNAseq2_Gene_Norm =
TRUE,
18             fileSizeLimit = 10000)
19
20 # To access the data you should use the getData function
21 # or simply access with @ (for example gbm.data@Clinical)
22 gbm.mut <- getData(gbm.data,"Mutations")
23 gbm.clin <- getData(gbm.data,"Clinical")
24 gbm.gistic <- getData(gbm.data,"GISTIC")

```

Listing 6. Downloading TCGA data files with RTCGAtoolbox

Finally, RTCGAtoolbox can access level 4 data, which can be handy when the user requires GISTIC results. GISTIC is used to identify genes targeted by somatic copy-number alterations (SCNAs)¹⁶ (see Listing 7).

```

1 # Download GISTIC results
2 gistic <- getFirehoseData("GBM",gistic2_Date ="20141017" )
3
4 # get GISTIC results
5 gistic.allbygene <- gistic@GISTIC@AllByGene
6 gistic.thresholdedbygene <- gistic@GISTIC@ThresholdedByGene

```

Listing 7. Using RTCGAToolbox to get the GISTIC results

Genomic analysis

Copy number variations (CNV) has 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. CNV 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. Data for selected samples were downloaded and prepared in two separate rse objects (RangedSummarizedExperiment).

```

1 #####
2 ## CNV data pre-processing ##
3 #####
4 library(TCGAbiolinks)
5

```



```

6 # Select available copy number platform for GBM and LGG.
7 PanCancer <- c("LGG","GBM")
8 PlatformCancer <- "Genome_Wide_SNP_6"
9 dataType <- "nocnv_hg19"
10
11 for(tumor in PanCancer){
12   pathCancer <- paste0("../data",tumor)
13
14   datQuery <- TCGAquery(tumor = tumor, platform = PlatformCancer, level = "3")
15   lsSample <- TCGAquery_samplesfilter(query = datQuery)
16
17   # Select primary solid tumor ("TP" 01)
18   selected <- TCGAquery_SampleTypes(barcode = lsSample$Genome_Wide_SNP_6,
19   typesample = "TP")
20   TCGAdownload(data = datQuery, path = pathCancer, type = dataType, samples =
21   selected)
22   dataAssay <- TCGAprepare(query = datQuery, dir = pathCancer, type = dataType,
23   save = TRUE, summarizedExperiment = TRUE,
24   samples = selected)
25   save(PlatformCancer, tumor, pathCancer, selected, dataAssay,
26   file = paste0(tumor, "_", PlatformCancer, ".rda"))
27 }

```

Listing 8. Searching, downloading and preparing CNV data with TCGAblinks

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 GAIA¹⁷, 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.

```

1 #####
2 ## Recurrent CNV identification ##
3 #####
4
5 for(cancer in c("LGG","GBM")){
6
7   load(paste0(cancer, "_Genome_Wide_SNP_6.rda"))
8
9   # Prepare CNV matrix
10  cnvMatrix <- dataAssay
11  # Add label (0 for loss, 1 for gain)
12  cnvMatrix <- cbind(cnvMatrix,Label=NA)
13  cnvMatrix[cnvMatrix["Segment_Mean"] < -0.3,"Label"] <- 0
14  cnvMatrix[cnvMatrix["Segment_Mean"] > 0.3,"Label"] <- 1
15  cnvMatrix <- cnvMatrix[!is.na(cnvMatrix$Label),]
16  # Remove "Segment_Mean" and change col.names
17  cnvMatrix <- cnvMatrix[, -6]
18  colnames(cnvMatrix) <- c("Sample.Name", "Chromosome", "Start", "End",
19  "Num.of.Markers", "Aberration")
20  # Substitute Chromosomes "X" and "Y" with "23" and "24"
21  xidx <- which(cnvMatrix$Chromosome=="X")
22  yidx <- which(cnvMatrix$Chromosome=="Y")
23  cnvMatrix[xidx,"Chromosome"] <- 23
24  cnvMatrix[yidx,"Chromosome"] <- 24
25  cnvMatrix$Chromosome <- sapply(cnvMatrix$Chromosome,as.integer)

```



```

25
26   # Recurrent CNV identification with GAIA
27
28   # Retrieve probes meta file from broadinstitute website
29   # Recurrent CNV identification with GAIA
30   gdac.root <- "ftp://ftp.broadinstitute.org/pub/GISTIC2.0/hg19_support/"
31   # Retrieve probes meta file from broadinstitute website
32   markersMatrix <- read.delim(paste0(gdac.root,"genome.info.6.0_hg19.na31_
minus_frequent_nan_probes_sorted_2.1.txt") ,
33                               as.is=TRUE, header=FALSE)
34   colnames(markersMatrix) <- c("Probe.Name", "Chromosome", "Start")
35   unique(markersMatrix$Chromosome)
36   xidx <- which(markersMatrix$Chromosome=="X")
37   yidx <- which(markersMatrix$Chromosome=="Y")
38   markersMatrix[xidx,"Chromosome"] <- 23
39   markersMatrix[yidx,"Chromosome"] <- 24
40   markersMatrix$Chromosome <- sapply(markersMatrix$Chromosome,as.integer)
41   markerID <- apply(markersMatrix,1,function(x) paste0(x[2],":",x[3]))
42   table(duplicated(markerID))
43   ## FALSE      TRUE
44   ## 1831041     186
45   # There are 186 duplicated markers
46   table(duplicated(markersMatrix$Probe.Name))
47   ## FALSE
48   ## 1831227
49   # ... with different names!
50   # Removed duplicates
51   markersMatrix <- markersMatrix[~which(duplicated(markerID)),]
52   #Filter markersMatrix for common CNV
53   markerID <- apply(markersMatrix,1,function(x) paste0(x[2],":",x[3]))
54   commonCNV <- read.delim(paste0(gdac.root,"CNV.hg19.bypos.111213.txt"),
as.is=TRUE)
55   commonCNV[,2] <- sapply(commonCNV[,2], as.integer)
56   commonCNV[,3] <- sapply(commonCNV[,3], as.integer)
57   commonID <- apply(commonCNV,1,function(x) paste0(x[2],":",x[3]))
58   table(commonID %in% markerID)
59   table(markerID %in% commonID)
60   markersMatrix_fil <- markersMatrix[!markerID %in% commonID,]
61
62   markers_obj <- load_markers(markersMatrix_fil)
63
64   cnv_obj <- load_cnv(cnvMatrix, markers_obj, length(selected))
65   results <- runGAIA(cnv_obj, markers_obj, output_file_name=paste0("GAIA_",cancer,
"_", PlatformCancer, "_flt.txt"), aberrations = -1,
66                       chromosomes = -1, num_iterations = 10, threshold = 0.25)
67
68   # Set q-value threshold
69   threshold <- 0.0001
70
71   # Plot the results
72   RecCNV <- t(apply(results,1,as.numeric))
73   colnames(RecCNV)<- colnames(results)
74   RecCNV <- cbind(RecCNV, score=0)
75   minval <- format(min(RecCNV[RecCNV[,"q-value"]!=0,"q-value"]),scientific=
FALSE)
76   minval <- substring(minval,1, nchar(minval)-1)
77   RecCNV[RecCNV[,"q-value"]==0,"q-value"] <- as.numeric(minval)
78   RecCNV[,"score"] <- sapply(RecCNV[,"q-value"],function(x) -log10(as.numeric(x)))
79   RecCNV[RecCNV[,"q-value"]==as.numeric(minval),]
80

```

```

81     source("gaiaCNVplot.R")
82     gaiaCNVplot(RecCNV,cancer,threshold)
83
84     save(results, RecCNV, threshold, file = paste0(cancer, "_CNV_results.rda"))
85 }

```

Listing 9. 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.

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.

```

1 #####
2 ## Recurrent CNV annotation ##
3 #####
4 for(cancer in c("LGG","GBM")){
5   load(paste0(cancer, "_CNV_results.rda" ))
6   mart <- useMart(biomart="ensembl", dataset="hsapiens_gene_ensembl")
7   genes <- getBM(attributes = c("hgnc_symbol", "chromosome_name", "start_
position", "end_position"), mart=mart)
8   genes <- genes[genes[,1]!="X" & genes[,2]%in%c(1:22,"X","Y"),]
9   xidx <- which(genes[,2]=="X")
10  yidx <- which(genes[,2]=="Y")
11  genes[xidx, 2] <- 23
12  genes[yidx, 2] <- 24
13  genes[,2] <- sapply(genes[,2], as.integer)
14  genes <- genes[order(genes[,3]),]
15  genes <- genes[order(genes[,2]),]
16  colnames(genes) <- c("GeneSymbol", "Chr", "Start", "End")
17  genes_GR <- makeGRangesFromDataFrame(genes, keep.extra.columns = TRUE)
18
19  sCNV <- RecCNV[RecCNV["q-value"]<=threshold,c(1:4,6)]
20  sCNV <- sCNV[order(sCNV[,3]),]
21  sCNV <- sCNV[order(sCNV[,1]),]
22  colnames(sCNV) <- c("Chr", "Aberration", "Start", "End", "q-value")
23  sCNV_GR <- makeGRangesFromDataFrame(sCNV, keep.extra.columns = TRUE)
24
25  hits <- findOverlaps(genes_GR, sCNV_GR, type="within")
26  sCNV_ann <- cbind(sCNV[subjectHits(hits),], genes[queryHits(hits),])
27  AberrantRegion <- paste0(sCNV_ann[,1], ":", sCNV_ann[,3], "-", sCNV_
ann[,4])
28  GeneRegion <- paste0(sCNV_ann[,7], ":", sCNV_ann[,8], "-", sCNV_ann[,9])
29  AmpDel_genes <- cbind(sCNV_ann[,c(6,2,5)], AberrantRegion, GeneRegion)
30  AmpDel_genes[AmpDel_genes[,2]==0,2] <- "Del"
31  AmpDel_genes[AmpDel_genes[,2]==1,2] <- "Amp"
32  rownames(AmpDel_genes) <- NULL
33
34   save(RecCNV, AmpDel_genes, file = paste0(cancer, "_CNV_results.rda"))
35 }

```

Listing 10. 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`¹⁸. The Listing 11 shows how to download mutation data using `TCGAquery_maf` (line 4), then we filtered the genes to obtain genes with

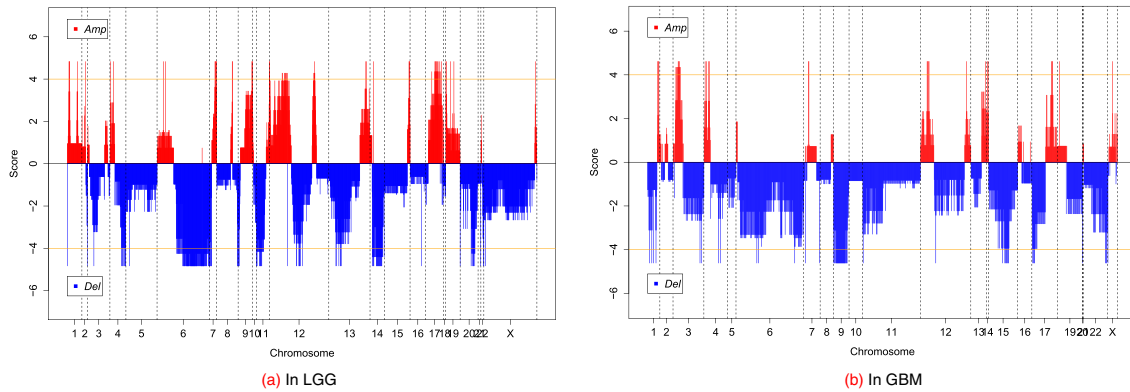


Figure 1. Recurrent CNV ($|score\ threshold| = 4$).

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

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. In the columns case, if an insertion is made at the top, will remove the barplot. The final result for adding the annotation to the bottom is highlighted in [Figure 2](#).

```

1 library(ComplexHeatmap) # Version 1.10.2
2 library(TCGAbiolinks)
3
4 LGGmut <- TCGAquery_maf(tumor = "LGG", archive.name="LGG.IlluminaGA_DNASeq_curated.Level_
2.1.4.0")
5 GBMmut <- TCGAquery_maf(tumor = "GBM", archive.name = "ucsc.edu_GBM.IlluminaGA_DNASeq_
automated.Level_2.1.1.0")
6 mut <- plyr::rbind.fill(LGGmut, GBMmut)
7
8 # Filtering mutations in gliomas
9 EA_pathways <- TCGAbiolinks::listEA_pathways
10 Glioma_pathways <- EA_pathways[grepl("glioma", tolower(EA_pathways$Pathway)),]
11 Glioma_signaling <- Glioma_pathways[Glioma_pathways$Pathway == "Glioma Signaling",]
12 Glioma_signaling_genes <- unlist(strsplit(as.character(Glioma_signaling$
Molecules), ","))
13
14 mut <- mut[mut$Hugo_Symbol %in% Glioma_signaling_genes,]
15
16 samples <- unique(mut$Tumor_Sample_Barcode)

```

```

17 genes <- unique(mut$Hugo_Symbol)
18 mat <- Matrix(0,length(genes),length(samples))
19 colnames(mat) <- samples
20 rownames(mat) <- genes
21
22 pb <- txtProgressBar(min = 0, max = nrow(mat), style = 3)
23
24 for (i in 1:nrow(mat)) {
25   curGene <- rownames(mat)[i]
26   setTxtProgressBar(pb, i)
27   for (j in 1:ncol(mat)) {
28     curSample <- colnames(mat)[j]
29
30     if (length(intersect(mut$Tumor_Sample_Barcode, curSample))==1){
31       mat1 <- mut[mut$Tumor_Sample_Barcode == curSample,]
32       if (length(intersect(mat1$Hugo_Symbol, curGene))==1){
33         mat3 <- mat1[mat1$Hugo_Symbol == curGene,]
34         mat[curGene,curSample]<- as.character(mat3$Variant_Type)[1]
35       }
36     }
37   }
38 }
39 close(pb)
40
41 mat[mat==0] <- ""
42 colnames(mat) <- substr(colnames(mat),1,12)
43
44 mat[is.na(mat)] = ""
45 #mat = t(as.Matrix(mat))
46 mat[1:3, 1:3]
47
48 alter_fun = list(
49   background = function (x, y, w, h) {
50     grid.rect(x, y, w-unit(0.5, "mm"), h-unit(0.5, "mm"), gp = gpar(fill = "#CCCCCC",
51   col = NA))
52   },
53   SNP = function(x, y, w, h) {
54     grid.rect(x, y, w-unit(0.5, "mm"), h-unit(0.5, "mm"), gp = gpar(fill = "blue",
55   col = NA))
56   },
57   DEL = function(x, y, w, h) {
58     grid.rect(x, y, w-unit(0.5, "mm"), h-unit(0.5, "mm"), gp = gpar(fill = "red",
59   col = NA))
60   },
61   INS = function(x, y, w, h) {
62     grid.rect(x, y, w-unit(0.5, "mm"), h*0.33, gp = gpar(fill = "#008000",
63   col = NA))
64   }
65 )
66
67 col = c("INS" = "#008000", "DEL" = "red", "SNP" = "blue")

```

```

64
65 clin.gbm <- TCGAquery_clinic("gbm", "clinical_patient")
66 clin.lgg <- TCGAquery_clinic("lgg", "clinical_patient")
67 clinical <- plyr::rbind.fill(clin.lgg,clin.gbm)
68 annotation <- clinical[match(colnames(mat),clinical$bcr_patient_barcode),
c("disease","radiation_therapy")]
69 annotation <- HeatmapAnnotation(annotation_height = rep(unit(0.3, "cm"),ncol(annotation)),
70                                df = annotation,
71                                col = list(disease = c("LGG"="green", "GBM"="orange"),
72                                             radiation_therapy = c("YES"="blue", "NO"=
"red", "[Unknown]"="yellow", "[Not Available]"="grey")),
73                                annotation_legend_param = list(title_gp = gpar(fontsize =
16, fontface = "bold"),
74                                                                labels_gp = gpar(fontsize =
16), # size labels
75                                                                grid_height = unit(8, "mm")))
76
77 pdf("LGG_GBM_oncoprint.pdf",width = 20,height = 20)
78 p <- oncoPrint(mat, get_type = function(x) strsplit(x, ";")[[1]],
79               remove_empty_columns = FALSE,
80               column_order = NULL, # Do not sort the columns
81               alter_fun = alter_fun, col = col,
82               row_names_gp = gpar(fontsize = 16), # set size for row names
83               pct_gp = gpar(fontsize = 16), # set size for percentage labels
84               axis_gp = gpar(fontsize = 16),# size of axis
85               column_title = "OncoPrint for TCGA LGG, genes in Glioma signaling",
86               column_title_gp = gpar(fontsize = 22),
87               pct_digits = 2,
88               row_barplot_width = unit(4, "cm"), #size barplot
89               bottom_annotation = annotation,
90               heatmap_legend_param = list(title = "Mutations", at = c("DEL", "INS", "SNP"),
91                                         labels = c("DEL", "INS", "SNP"),
92                                         title_gp = gpar(fontsize = 16, fontface = "bold"),
93                                         labels_gp = gpar(fontsize = 16), # size labels
94                                         grid_height = unit(8, "mm")
95               )
96 )
97 draw(p, annotation_legend_side = "bottom")
98 dev.off()

```

Listing 11. Oncoprint

Overview of genomic alterations by circo plot

Genomic alterations in cancer, including CNV and mutations, can be represented in an effective overview plot named circo. We used [circlize](#) CRAN package to represent significant CNV (resulting from GAIA analysis) and recurrent mutations (selecting curated genetic variations retrieved from TCGA that are identified in at least two tumor samples) in LGG (see [Listing 13](#)). Circo plot can illustrate molecular alterations genome-wide or only in one or more selected chromosomes. The [Figure 3](#) shows the resulting circo plot for all chromosomes, while the [Figure 4](#) shows the plot for only the chromosome 17.

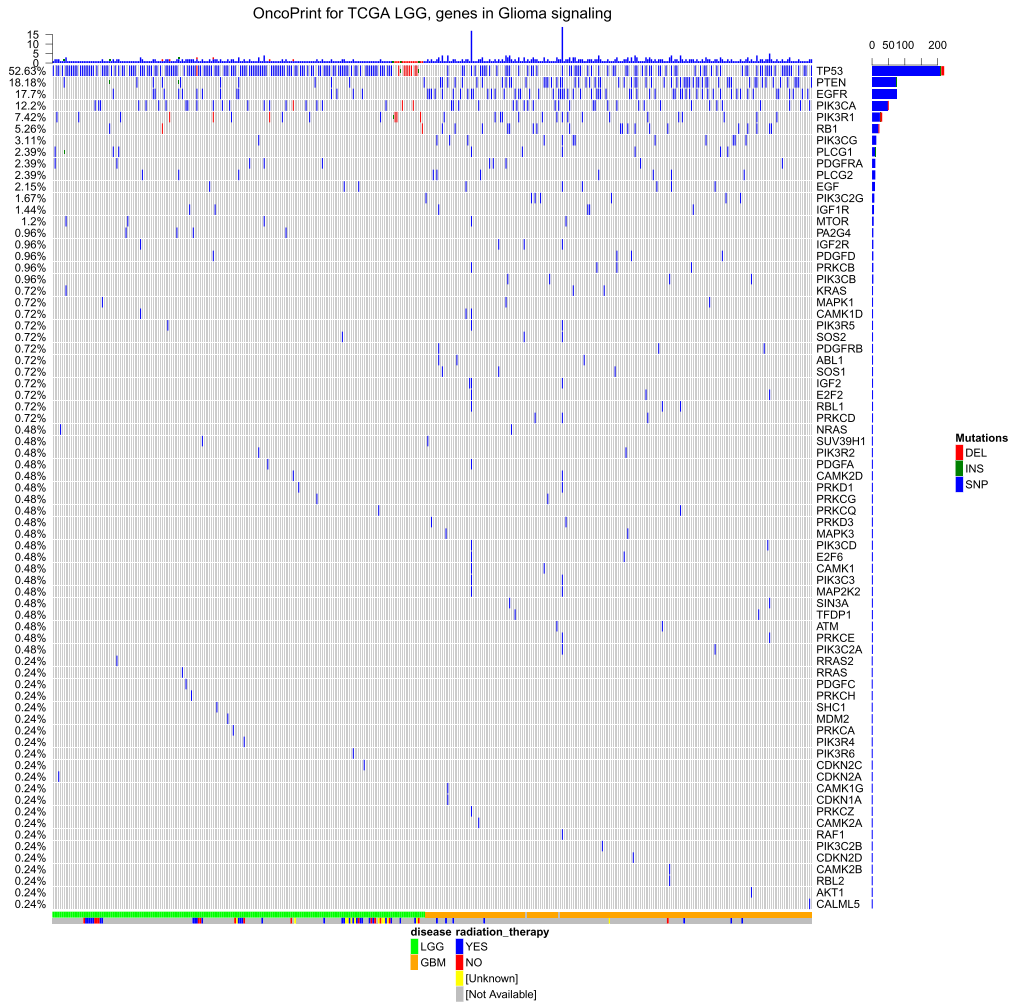


Figure 2. OncoPrint for LGG samples. Blue defines SNP, green defines insertions and red defines deletions. The upper barplot shows the number of these genetic mutation for each patient, while the right barplot shows the number of genetic mutations for each gene. The bottom bar shows the group of each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

```

1 #####
2 ## Genomic aberration overview - Circos plot ##
3 #####
4
5 # Retrieve curated mutations for selected cancer (e.g. "LGG")
6 library(TCGAbiolinks)
7 mut <- TCGAquery_maf(tumor = "LGG",archive.name="LGG.IlluminaGA_DNASeq_curated.
Level_2.1.4.0")
8 # Select only potentially damaging mutations
9 mut <- mut[mut$Variant_Classification %in% c("Missense_Mutation","Nonsense_Mutation",
"Nonstop_Mutation","Frame_Shift_Del","Frame_Shift_Ins"),]
10 # Select recurrent mutations (identified in at least two samples)
11 mut.id <- paste0(mut$Chromosome,":",mut$Start_position,"-",mut$End_position,"|",mut$
Reference_Allele,"/",mut$Tumor_Seq_Allele2)
12 mut <- cbind(mut.id, mut)
13 numSamples <- table(mut.id)
14 s.mut <- names(which(numSamples>=2))
15 # Prepare selected mutations data for circos plot
16 s.mut <- mut[mut$mut.id %in% s.mut,]
17 s.mut <- s.mut[,c("Chromosome","Start_position","End_position","Variant_Classification",
"Hugo_Symbol")]
18 s.mut <- unique(s.mut)

```

```

19 Chromosome <- sapply(s.mut[,1],function(x) paste0("chr",x))
20 s.mut <- cbind(Chromosome,s.mut[,-1])
21 s.mut[,1] <- as.character(s.mut[,1])
22 s.mut[,4] <- as.character(s.mut[,4])
23 s.mut[,5] <- as.character(s.mut[,5])
24 typeNames <- unique(s.mut[,4])
25 type <- c(4:1)
26 names(type) <- typeNames[1:4]
27 Type <- type[s.mut[,4]]
28 s.mut <- cbind(s.mut,Type)
29 s.mut <- s.mut[,c(1:3,6,4,5)]
30
31 # Load recurrent CNV data for selected cancer (e.g. "LGG")
32 load("LGG_CNV_results.rda")
33 # Prepare selected sample CNV data for circos plot
34 s.cnv <- as.data.frame(RecCNV[RecCNV[,"q-value"]<=10^-4,c(1:4,6)])
35 s.cnv <- s.cnv[,c(1,3,4,2)]
36 xidx <- which(s.cnv$Chromosome==23)
37 yidx <- which(s.cnv$Chromosome==24)
38 s.cnv[xidx,"Chromosome"] <- "X"
39 s.cnv[yidx,"Chromosome"] <- "Y"
40 Chromosome <- sapply(s.cnv[,1],function(x) paste0("chr",x))
41 s.cnv <- cbind(Chromosome, s.cnv[,-1])
42 s.cnv[,1] <- as.character(s.cnv[,1])
43 s.cnv[,4] <- as.character(s.cnv[,4])
44 s.cnv <- cbind(s.cnv,CNV=1)
45 colnames(s.cnv) <- c("Chromosome","Start_position","End_position","Aberration_Kind","CNV")
46
47 # Draw genomic circos plot
48 library(circlize)
49 pdf("CircosPlot.pdf",width=15,height=15)
50 par(mar=c(1,1,1,1), cex=1)
51 circos.initializeWithIdeogram()
52 # Add CNV results
53 colors <- c("forestgreen","firebrick")
54 names(colors) <- c(0,1)
55 circos.genomicTrackPlotRegion(s.cnv, ylim = c(0,1.2),
56                               panel.fun = function(region, value, ...) {
57                                   circos.genomicRect(region, value, ytop.column = 2,
58                                                       col = colors[value[[1]]],
59                                                       border="white")
60                                   cell.xlim = get.cell.meta.data("cell.xlim")
61                                   circos.lines(cell.xlim, c(0, 0), lty = 2, col =
62 "#00000040")
63                               })
64 # Add mutation results
65 colors <- c("blue","green","red","gold")
66 names(colors) <- typeNames[1:4]
67 circos.genomicTrackPlotRegion(s.mut, ylim = c(1.2,4.2),
68                               panel.fun = function(region, value, ...) {
69                                   circos.genomicPoints(region, value, cex = 0.8, pch = 16,
70                                                         col = colors[value[[2]]], ...)
71                               })
72
73 circos.clear()
74
75 legend(-0.2, 0.2, bty="n", y.intersp=1, c("Amp","Del"), pch=15, col=c("firebrick",
76 "forestgreen"), title="CNVs", text.font=3, cex=1.2, title.adj=0)
77 legend(-0.2, 0, bty="n", y.intersp=1, names(colors), pch=16, col=colors, title="Mutations",
78 text.font=3, cex=1.2, title.adj=0)
79 dev.off()
80
81 # Draw single chromosome circos plot (e.g. "Chr 17")
82 pdf("CircosPlotChr17.pdf",width=18,height=13)
83 par(mar=c(1,1,1,1),cex=1.5)
84 circos.par("start.degree" = 90, canvas.xlim = c(0, 1), canvas.ylim = c(0, 1),
85            gap.degree = 270, cell.padding = c(0, 0, 0, 0), track.margin = c(0.005, 0.005))

```



```

82 circos.initializeWithIdeogram(chromosome.index = "chr17")
83 circos.par(cell.padding = c(0, 0, 0, 0))
84 # Add CNV results
85 colors <- c("forestgreen", "firebrick")
86 names(colors) <- c(0,1)
87 circos.genomicTrackPlotRegion(s.cnv, ylim = c(0,1.2),
88                               panel.fun = function(region, value, ...) {
89                                   circos.genomicRect(region, value, ytop.column = 2,
ybottom = 0,
90                                                       col = colors[value[[1]]],
91                                                       border="white")
92                                   cell.xlim = get.cell.meta.data("cell.xlim")
93                                   circos.lines(cell.xlim, c(0, 0), lty = 2, col =
"#00000040")
94                               })
95
96 # Add mutation results representing single genes
97 genes.mut <- paste0(s.mut$Hugo_Symbol, "-", s.mut$Type)
98 s.mutt <- cbind(s.mut, genes.mut)
99 n.mut <- table(genes.mut)
100 idx <- !duplicated(s.mutt$genes.mut)
101 s.mutt <- s.mutt[idx,]
102 s.mutt <- cbind(s.mutt, num=n.mut[s.mutt$genes.mut])
103 genes.num <- paste0(s.mutt$Hugo_Symbol, "(", s.mutt$num, ")")
104 s.mutt <- cbind(s.mutt[, -c(6:8)], genes.num)
105 s.mutt[, 6] <- as.character(s.mutt[, 6])
106 s.mutt[, 4] <- s.mutt[, 4]/2
107
108 colors <- c("blue", "green", "red", "gold")
109 names(colors) <- typeNames[1:4]
110 circos.genomicTrackPlotRegion(s.mutt, ylim = c(0.3,2.2), track.height = 0.05,
111                               panel.fun = function(region, value, ...) {
112                                   circos.genomicPoints(region, value, cex = 0.8, pch = 16,
col = colors[value[[2]]], ...)
113                               })
114
115 circos.genomicTrackPlotRegion(s.mutt, ylim = c(0, 1), track.height = 0.1, bg.border =
NA)
116 i_track = get.cell.meta.data("track.index")
117
118 circos.genomicTrackPlotRegion(s.mutt, ylim = c(0,1),
119                               panel.fun = function(region, value, ...) {
120                                   circos.genomicText(region, value,
121                                                       y = 1,
122                                                       labels.column = 3,
123                                                       col = colors[value[[2]]],
124                                                       facing = "clockwise", adj =
c(1, 0.5),
125                                                       postTransform = postTransform.text,
cex = 1.5, niceFacing = T)
126                               }, track.height = 0.1, bg.border = NA)
127
128 circos.genomicPosTransformLines(s.mutt,
129                                postTransform = function(region, value)
130                                postTransform.text(region,
131                                                    y = 1,
132                                                    labels = value[[3]],
133                                                    cex = 0.8, track.index =
i_track+1),
134                                direction = "inside", track.index = i_track)
135
136 circos.clear()
137
138 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)
139 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)
140 dev.off()

```

Listing 12. Genomic aberration overview by circos plot

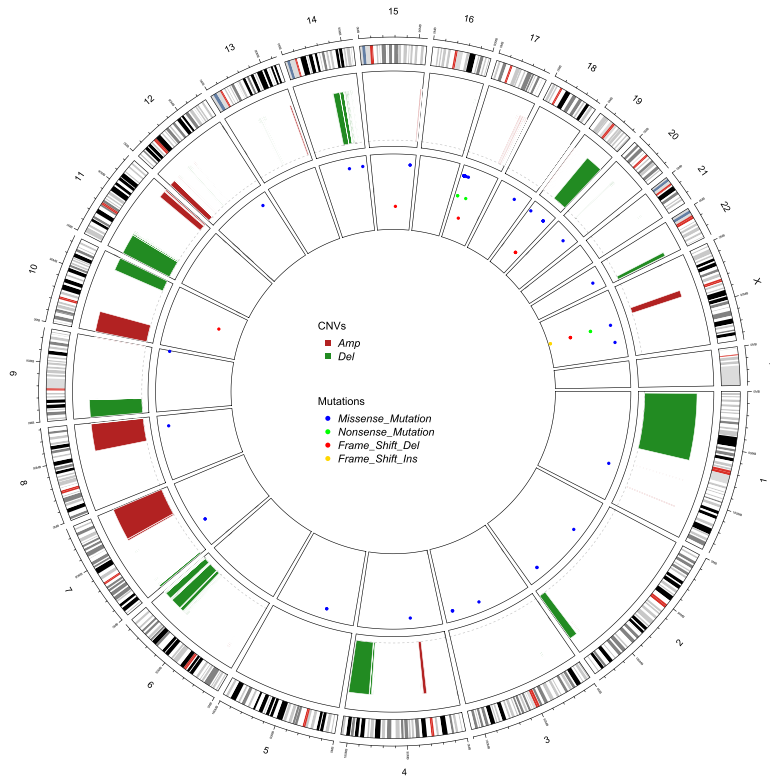


Figure 3. Circos plot of recurrent CNV and mutations in LGG.

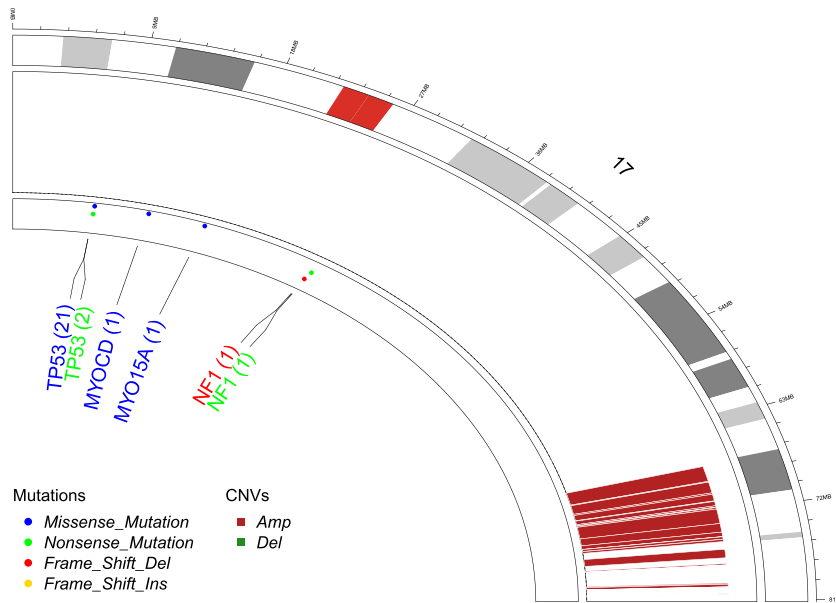


Figure 4. Circos plot of chromosome 17 recurrent CNV and mutations in LGG.

Transcriptomic analysis

Pre-Processing Data. The **LGG** and **GBM** data used for following transcriptomic analysis were downloaded using TCGA Biolinks. 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 object (RangedSummarizedExperiment) saving them as an R object with a filename including both the name of the cancer and the name of the platform used for gene expression data (see Listing 13).

```

1 library(TCGA Biolinks)
2
3 # defining common parameters
4 PanCancer <- c("LGG", "GBM")
5 PlatformCancer <- "IlluminaHiSeq_RNASeqV2"
6 dataType <- "rsem.genes.results"
7
8 for(tumor in PanCancer){
9   pathCancer <- paste0("../ data", tumor)
10  datQuery <- TCGAquery(tumor = tumor, platform = PlatformCancer, level = "3")
11
12  # get only primary solid Tumor
13  lsSample <- TCGAquery_samplesfilter(query = datQuery)
14  dataSmTP <- TCGAquery_SampleTypes(barcode = lsSample$IlluminaHiSeq_RNASeqV2,
15  typesample = "TP")
16  TCGAdownload(data = datQuery , path = pathCancer, type = dataType, samples =
17  c(dataSmTP))
18  dataAssy <- TCGAprepare(query = datQuery , dir = pathCancer, type = dataType,
19  save = TRUE, summarizedExperiment = TRUE,
20  samples = c(dataSmTP),
21  filename = paste0(tumor, "_", PlatformCancer, ".rda")
22 }
```

Listing 13. Searching, downloading and preparing RNA-seq data with TCGA Biolinks

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 14). In this way 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 ($\text{cor.cut} = 0.6$) 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.

This function implements Within-lane normalization procedures to adjust for GC-content effect (or other gene-level effects) on read counts: loess robust local regression, global-scaling, and full-quantile normalization¹⁹ and between-lane normalization procedures to adjust for distributional differences between lanes (e.g., sequencing depth): global-scaling and full-quantile normalization²⁰.

```

1 library(TCGA Biolinks)
2
3 # loading LGG and GBM rse data
4
5 load("LGG_IlluminaHiSeq_RNASeqV2.rda")
6 cancer <- "LGG"
7 pathCancer <- paste0("../data", cancer)
8 PlatformCancer <- "IlluminaHiSeq_RNASeqV2"
9
10 dataClin_LGG <- TCGAquery_clinic(tumor = "LGG",
11  clinical_data_type = "clinical_patient")
```

```

12
13
14 dataPrep_LGG <- TCGAanalyze_Preprocessing(object = rse,
15                                         cor.cut = 0.6,
16                                         filename = "LGG_IlluminaHiSeq_RNASeqV2.png")
17
18 load("GBM_IlluminaHiSeq_RNASeqV2.rda")
19 cancer <- "GBM"
20 pathCancer <- paste0("../data",cancer)
21 PlatformCancer <- "IlluminaHiSeq_RNASeqV2"
22
23 dataClin_GBM <- TCGAquery_clinic(tumor = "GBM",
24                                 clinical_data_type = "clinical_patient")
25
26 dataPrep_GBM <- TCGAanalyze_Preprocessing(object = rse,
27                                         cor.cut = 0.6,
28                                         filename = "GBM_IlluminaHiSeq_RNASeqV2.png")
29
30 dataNorm <- TCGAanalyze_Normalization(tabDF = cbind(dataPrep_LGG, dataPrep_GBM),
31                                       geneInfo = geneInfo,
32                                       method = "gcContent") #18323 672
33
34 dataFilt <- TCGAanalyze_Filtering(tabDF = dataNorm,
35                                  method = "quantile",
36                                  qnt.cut = 0.25) #13742 672
37
38 save(dataFilt, file = paste0("LGG_GBM_Norm_",PlatformCancer, ".rda"))
39
40 dataFiltLGG <- subset(dataFilt, select = substr(colnames(dataFilt),1,12) %in% dataClin_
LGG$bcr_patient_barcode)
41 dataFiltGBM <- subset(dataFilt, select = substr(colnames(dataFilt),1,12) %in% dataClin_
GBM$bcr_patient_barcode)
42
43 dataDEGs <- TCGAanalyze_DEA(mat1 = dataFiltLGG,
44                             mat2 = dataFiltGBM,
45                             Cond1type = "LGG",
46                             Cond2type = "GBM",
47                             fdr.cut = 0.01 ,
48                             logFC.cut = 1,
49                             method = "glmLRT")

```

Listing 14. Normalizing mRNA transcripts and differentially 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 from DEGs we performed an enrichment analysis using *TCGAanalyze_EA_complete* function (see Listing 15).

```

1 ansEA <- TCGAanalyze_EAcomplete(TFname="DEA genes LGG Vs GBM", RegulonList =
rownames(dataDEGs))
2
3 TCGAvisualize_EAbarplot(tf = rownames(ansEA$ResBP),
4                          GOBPTab = ansEA$ResBP, GOCCTab = ansEA$ResCC,
5                          GOMFTab = ansEA$ResMF, PathTab = ansEA$ResPat,
6                          nRGTab = rownames(dataDEGs),
7                          nBar = 20)

```

Listing 15. 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 (GO:biological process, GO:cellular component, and GO:molecular function).

The [Figure 5](#) shows canonical pathways significantly overrepresented (enriched) by the DEGs. The most statistically significant canonical pathways identified in DEGs list are listed 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*²¹ can be used. [Listing 16](#) 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.

```

1 GenelistComplete <- rownames(assay(rse,1))
2
3 # DEGs TopTable
4 dataDEGsFiltLevel <- TCGAanalyze_LevelTab(dataDEGs, "LGG", "GBM",
5                                           dataFilt[, colnames(dataFiltLGG)],
6                                           dataFilt[, colnames(dataFiltGBM)])
7
8 dataDEGsFiltLevel$GeneID <- 0
9
10 # Converting Gene symbol to geneID
11 library(clusterProfiler)
12 eg = as.data.frame(bitr(dataDEGsFiltLevel$mRNA,
13                         fromType="SYMBOL",
14                         toType="ENTREZID",
15                         annoDb="org.Hs.eg.db"))
16 eg <- eg[!duplicated(eg$SYMBOL),]
17
18 dataDEGsFiltLevel <- dataDEGsFiltLevel[dataDEGsFiltLevel$mRNA %in% eg$SYMBOL,]
19
20 dataDEGsFiltLevel <- dataDEGsFiltLevel[order(dataDEGsFiltLevel$mRNA,decreasing=
FALSE),]
21 eg <- eg[order(eg$SYMBOL,decreasing=FALSE),]
22
23 # table(eg$SYMBOL == dataDEGsFiltLevel$mRNA) should be TRUE
24 all(eg$SYMBOL == dataDEGsFiltLevel$mRNA)
25 dataDEGsFiltLevel$GeneID <- eg$ENTREZID
26
27 dataDEGsFiltLevel_sub <- subset(dataDEGsFiltLevel, select = c("GeneID", "logFC"))
28 genelistDEGs <- as.numeric(dataDEGsFiltLevel_sub$logFC)
29 names(genelistDEGs) <- dataDEGsFiltLevel_sub$GeneID
30
31 require("pathview")
32 # pathway.id : hsa05214 is the glioma pathway
33 # limit : sets the limit for gene expression legend and color
34 hsa05214 <- pathview(gene.data = genelistDEGs,
35                     pathway.id = "hsa05214",
36                     species = "hsa",
37                     limit = list (gene=as.integer(max(abs(genelistDEGs)))))

```

Listing 16. Pathways enrichment analysis with pathview package

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

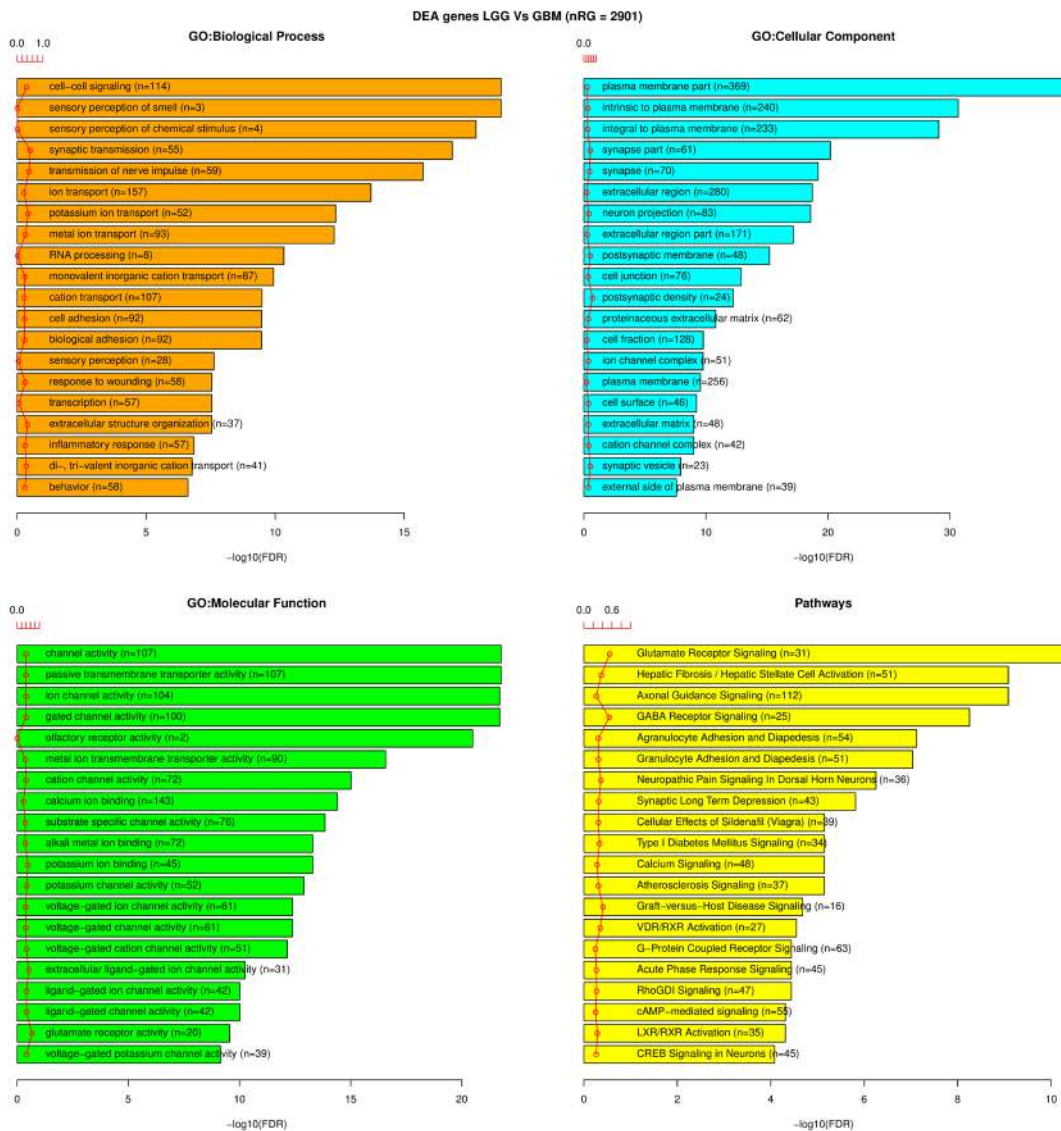


Figure 5. The plot shows canonical pathways significantly overrepresented (enriched) by the DEGs (differentially expressed genes) with the number of genes for the main categories of three ontologies (GO:biological process, GO:cellular component, and GO:molecular function, respectively). The most statistically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

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: ARACNE²², CLR²³, MRNET²⁴ and C3NET²⁵. 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 (MINET²⁶ and c3net²⁵, respectively).

Many gene regulatory interactions have been experimentally validated and published. These ‘known’ interactions can be accessed using different tools and databases such as BioGrid²⁷ or GeneMANIA²⁸. 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. 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

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

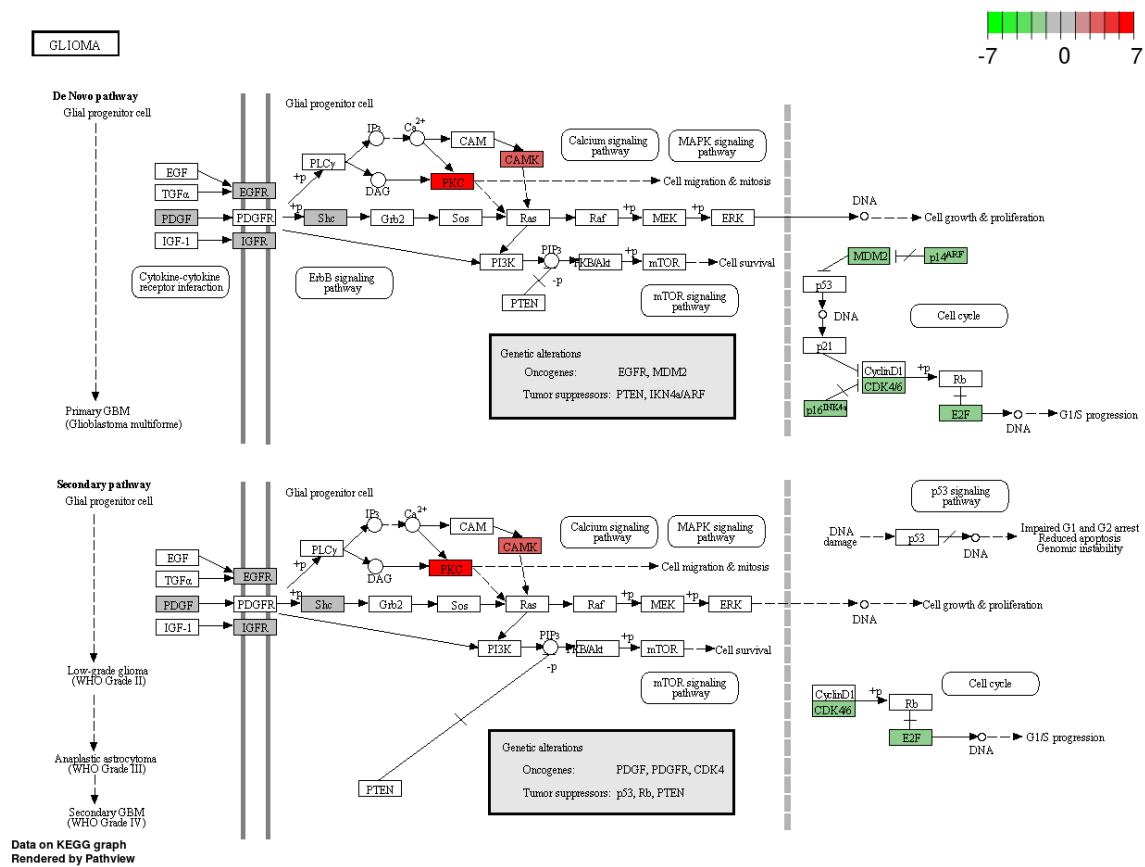


Figure 6. Pathways enrichment analysis : glioma pathway. Red defines genes that are up-regulated and green defines genes that are down-regulated.

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

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

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

We obtained a set of known interactions from the [BioGrid database](#).

```

1 get.adjacency.biogrid <- function(tmp.biogrid, names.genes = NULL){
2
3   if(is.null(names.genes)){
4     names.genes <- sort(union(unique(tmp.biogrid[, "Official.Symbol.Interactor.A"]),
5     unique(tmp.biogrid[, "Official.Symbol.Interactor.B"])))
6     ind <- seq(1, nrow(tmp.biogrid))
7   }else{
8     ind.A <- which(tmp.biogrid[, "Official.Symbol.Interactor.A"] %in% names.genes)
9     ind.B <- which(tmp.biogrid[, "Official.Symbol.Interactor.B"] %in% names.genes)
10
11     ind <- intersect(ind.A, ind.B)
12   }
13
14   mat.biogrid <- matrix(0, nrow=length(names.genes), ncol=length(names.genes),
15   dimnames=list(names.genes, names.genes))
16   for(i in ind){
17     mat.biogrid[tmp.biogrid[i, "Official.Symbol.Interactor.A"], tmp.biogrid[i, "Official.
18     Symbol.Interactor.B"]] <- mat.biogrid[tmp.biogrid[i, "Official.Symbol.Interactor.B"],
19     tmp.biogrid[i, "Official.Symbol.Interactor.A"]] <- 1
20   }
21   diag(mat.biogrid) <- 0
22   return(mat.biogrid)
23 }
```

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 for the LGG data set using one gene set.

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

```

1  ### plot details (colors & symbols)
2  mycols<-c('#e41a1c','#377eb8','#4daf4a','#984ea3','#ff7f00','#ffff33','#a65628')
3
4  ### load network inference libraries
5  library(minet)
6  library(c3net)
7
8  ### differentially identified genes using TCGAbiolinks
9  names.genes.de <- rownames(dataDEGs)
10
11 ### read biogrid info
12 library(downloader)
13 file <- "http://the.biogrid.org/downloads/archives/Release%20Archive/BIOGRID-3.4.133/
BIOGRID-ALL-3.4.133.tab2.zip"
14 download(file ,basename(file))
15 unzip(basename(file),junkpaths =T)
16 tmp.biogrid <- read.csv(gsub("zip","txt",basename(file)), header=TRUE, sep="\t",
stringsAsFactors=FALSE)
17 net.biogrid.de <- get.adjacency.biogrid(tmp.biogrid, names.genes.de)
18
19 for (cancertype in c("LGG", "GBM")) {
20
21   if(cancertype == "GBM"){
22     mydata <- dataFiltGBM[names.genes.de, ]
23   } else if(cancertype == "LGG"){
24     mydata <- dataFiltLGG[names.genes.de, ]
25   }
26   ### infer networks
27   net.aracne <- minet(t(mydata), method = "aracne")
28   net.mrnet <- minet(t(mydata))
29   net.clr <- minet(t(mydata), method = "clr")
30   net.c3net <- c3net(mydata)
31
32   ### validate compared to biogrid network
33   tmp.val <- list(validate(net.aracne, net.biogrid.de), validate(net.mrnet, net.
biogrid.de),
34                   validate(net.clr, net.biogrid.de), validate(net.c3net, net.
biogrid.de))
35
36   ### plot roc and compute auc for the different networks
37   dev1 <- show.roc(tmp.val[[1]],cex=0.3,col=mycols[1],type="l")
38   res.auc <- auc.roc(tmp.val[[1]])
39   for(count in 2:length(tmp.val)){
40     show.roc(tmp.val[[count]],device=dev1,cex=0.3,col=mycols[count],type="l")
41     res.auc <- c(res.auc, auc.roc(tmp.val[[count]]))
42   }
43
44   legend("bottomright", legend=paste(c("aracne","mrnet","clr","c3net"), signif(res.
auc,4), sep=": "),
45         col=mycols[1:length(tmp.val)],lty=1, bty="n" )
46   dev.copy2pdf(width=8,height=8,device = dev1, file = paste0("roc_biogrid_",cancertype,
".pdf"))
47   save(net.aracne, net.mrnet, net.clr, net.c3net, file=paste0("nets_",cancertype,
".RData"))
48
49 }

```

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.

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 stability²⁹.

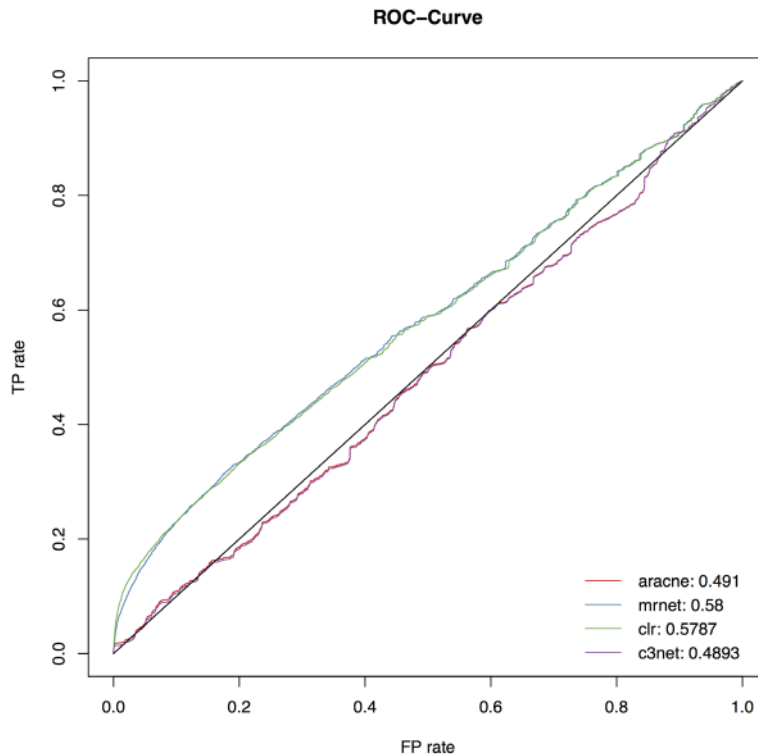


Figure 7. ROC with corresponding AUC for inferred GBM networks compared to BioGrid interactions using 2901 genes.

In mammals, DNA methylation is found sparsely but globally, distributed in definite CpG sequences throughout the entire genome. There is however an exception, the CpG islands (CGIs) which are short interspersed DNA sequences which are enriched for GC. These CpG islands are normally found in sites of transcription initiation and their methylation can lead to gene silencing³⁰

Thus, the investigation of the DNA methylation is crucial to understanding regulatory gene networks in cancer as the DNA methylation represses transcription³¹. Therefore, the DMR (differentially Methylated Region) detection can help us investigate regulatory gene networks.

This section describes the analysis of DNA methylation using the bioconductor package [TCGAbiolinks](#)⁷. 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 1–7 of the [Listing 17](#) describes how to make the data acquisition). We started by checking the mean DNA methylation of different groups of samples, then a DMR (Differentially methylated region) analysis is performed in which we search for regions that have possible biological significance, for example, 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 have 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 [illumina manual](#). Therefore, the rs probes were removed (see [Listing 17](#) lines 43). Also, probes in chromosomes X, Y were removed to eliminate potential artifacts originating from the presence of a different proportion of males and females³². The last pre-processing steps were to remove probes with at least one NA value (see [Listing 17](#) lines 40).

After this pre-processing step, using the function `TCGAvisualize_meanMethylation` provided we can take a 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 17](#) lines 46–50).

```

1 #-----
2 # Obtaining DNA methylation
3 #-----
4 library(TCGAbiolinks)
5 library(stringr)
6 # Samples
7 matched_met_exp <- function(tumor, n = NULL){
8   # get primary solid tumor samples: DNA methylation
9   met450k <- TCGAquery(tumor = tumor,"HumanMethylation450", level = 3)
10  met450k.tp <- TCGAquery_SampleTypes(unique(unlist(stringr::str_split(met450k$
barcode,","))),c("TP"))
11
12  # get primary solid tumor samples: RNAseq
13  rnaseq <- TCGAquery(tumor = tumor,c("IlluminaHiSeq_RNASeqV2"),level = 3)
14  rnaseq.tp <- TCGAquery_SampleTypes(unique(unlist(stringr::str_split(rnaseq$
barcode,","))),c("TP"))
15
16  # Get patients with samples in both platforms
17  patients <- unique(substr(rnaseq.tp,1,15)[substr(rnaseq.tp,1,12) %in%
substr(met450k.tp,1,12)] )
18  if(!is.null(n)) patients <- patients[1:n] # get only n samples
19  return(patients)
20 }
21 lgg.samples <- matched_met_exp("LGG", n = 10)
22 gbm.samples <- matched_met_exp("GBM", n = 10)
23 samples <- c(lgg.samples,gbm.samples)
24
25 #-----
26 # 1 - methylation
27 # -----
28 # For methylation it is quicker in this case to download the tar.gz file
29 # and get the samples we want instead of downloading files by files
30 query.met <- TCGAquery(tumor = c("GBM","LGG"), platform = "HumanMethylation450",
level = 3, samples = samples)
31 TCGAdownload(query.met, samples = samples)
32 met <- TCGAprepare(query.met,dir = ".", save = FALSE,samples = samples)
33 #-----
34 # Mean methylation
35 #-----
36 # Plot a barplot for the groups in the disease column in the
37 # summarizedExperiment object
38
39 # remove probes with NA (similar to na.omit)
40 met <- subset(met,subset = (rowSums(is.na(assay(met))) == 0))
41
42 # remove probes in chromosomes X, Y and NA
43 met <- subset(met,subset = !as.character(seqnames(met)) %in% c("chrNA","chrX","chrY"))
44
45

```

```

46 TCGAvisualize_meanMethylation(met,
47                               groupCol = "disease",
48                               group.legend = "Groups",
49                               filename = "mean_lgg_gbm.png",
50                               print.pvalue = TRUE)

```

Listing 17. 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 = 6.1×10^{-6}).

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 18**). 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, it calculates the difference between the mean DNA methylation (mean of the beta-values) of each group for each probe. Second, it tests 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} .

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 18**. This plot aids the user in selecting relevant thresholds, as we search for candidate biological DMRs.

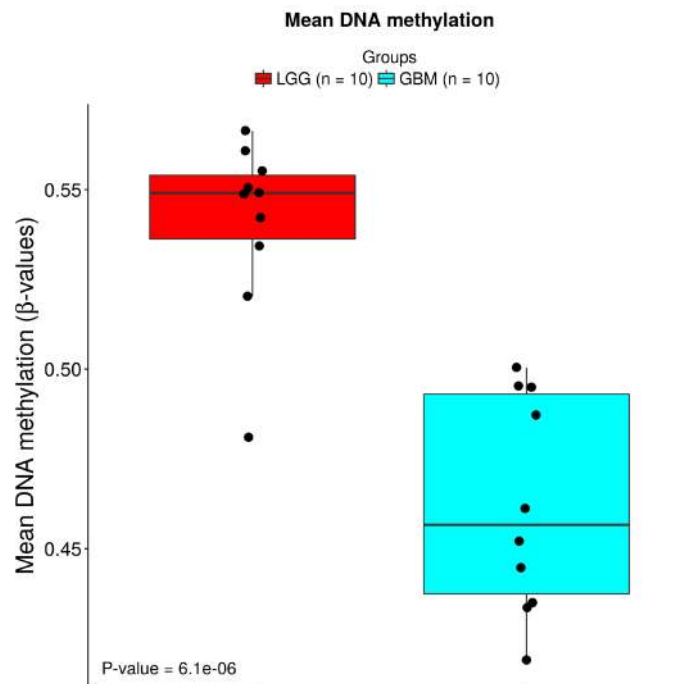


Figure 8. Boxplot of mean DNA methylation of each sample (black dots).

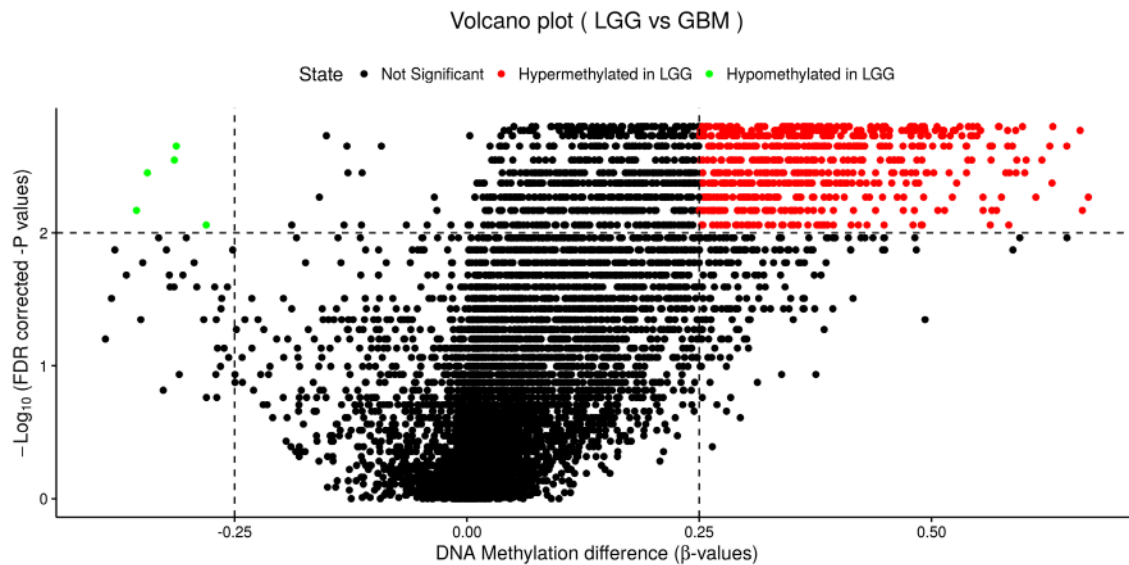


Figure 9. Volcano plot: searching for differentially methylated CpG sites (x-axis: difference of mean DNA methylation, y-axis: statistical significance).

```

1 # Becareful! Depending on the number of probes and samples this function might take
some days.
2 # To make this example faster we used only the chromosome 9
3 # This should take some minutes
4 met.chr9 <- subset(met, subset = as.character(seqnames(met)) %in% c("chr9"))
5
6 met.chr9 <- TCGAanalyze_DMR(met.chr9,
7   groupCol = "disease", # a column in the colData matrix
8   group1 = "GBM", # a type of the the groupCol column
9   group2="LGG", # a type of the the groupCol column
10  p.cut = 10^-2,
11  diffmean.cut = 0.25,
12  legend = "State",
13  plot.filename = "LGG_GBM_metvolcano.png",
14  cores = 1 # if set to 1 there will be a progress bar
15 )

```

Listing 18. 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`¹⁸. 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 19 shows the code to produce the heatmap of a DNA methylation data (Figure 10).

```

1 #-----
2 # DNA methylation heatmap
3 #-----
4 library(ComplexHeatmap)
5
6 # get the probes that are Hypermethylated or Hypomethylated
7 # met is the same object of the section 'DNA methylation analysis'
8 sig.met <- met.chr9[values(met.chr9)$status.GBM.LGG %in% c("Hypermethylated",
" Hypomethylated"),]
9

```

```

10 # To speed up the example, let take a look on the first 100 probes
11 sig.met.100 <- sig.met[1:100,]
12
13 # top annotation, which samples are LGG and GBM
14 # We will add clinical data as annotation of the samples
15 # we will sort the clinical data to have the same order of the DNA methylation matrix
16 clinical.order <- clinical[match(substr(colnames(sig.met.100),1,12),clinical$bcr_patient_
barcode),)]
17 ta = HeatmapAnnotation(df = clinical.order[,c("disease","gender","icd_o_3_histology",
"tumor_tissue_site")],
18                       col = list(disease = c("LGG" = "grey", "GBM" = "black"),
19                                gender = c("MALE"="blue", "FEMALE"="pink")),
20
21 # row annotation: add the status for LGG in relation to GBM
22 # For example: status.gbm.lgg Hypomethylated means that the mean DNA methylation of
23 # probes for lgg are hypomethylated compared to GBM ones.
24 ra = rowAnnotation(df = values(sig.met.100)[ "status.GBM.LGG" ],
25                   col = list(status.GBM.LGG = c("Hypomethylated" = "orange",
26                                                "Hypermethylated" = "darkgreen")),
27                   width = unit(1, "cm"))
28
29 heatmap <- Heatmap(assay(sig.met.100),
30                   name = "DNA methylation",
31                   top_annotation = ta,
32                   col = matlab::jet.colors(200),
33                   show_row_names = F,
34                   cluster_rows = T,
35                   cluster_columns = F,
36                   show_column_names = F,
37                   column_title = "DNA methylation") + ra
38 png("heatmap.png",width = 600, height = 400); draw(heatmap); dev.off()

```

Listing 19. 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 sequence short nucleotide sequences (6–15 bp) might have a biological significance as it can be used to control the expression of genes. These sequences are called Regulatory motifs. The bioconductor package **rGADEM**^{33,34} 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 in the Listing 20). An object will be returned which contains all relevant information about your motif analysis (sequence consensus, pwm, chromosome, pvalue...).

Using bioconductor package **motifStack**³⁵ it is possible to generate a graphic representation of multiple motifs with different similarity scores (see Figure 11).

```

1 library(rGADEM)
2 library(BSgenome.Hsapiens.UCSC.hg19)
3 library(motifStack)
4
5 probes <- rowRanges(met.chr9)[values(met.chr9)$status.GBM.LGG %in%
c("Hypermethylated", "Hypomethylated") ,]
6 # Get hypo/hyper methylated probes and make a 200bp window surrounding each probe.
7 sequence <- RangedData(space=as.character(probes@seqnames),
8                       IRanges(start=probes@ranges@start - 100,
9                               end=probes@ranges@start + 100), strand="*")
10 #look for motifs
11 gadem <- GADEM(sequence, verbose=1, genome=Hsapiens)
12
13 # How many motifs were found?
14 length(gadem@motifList)

```



```

15
16 # get the number of occurrences
17 nOccurrences(gadem)
18
19 # view all sequences consensus
20 consensus(gadem)
21
22 # print the first two motif logos.
23 plot(gadem@motifList[[1]])
24 pwm <- getPWM(gadem)
25 pfm <- new("pfm",mat=pwm[[1]],name="Novel Site 1")
26 plotMotifLogo(pfm)
27
28 # Number of instances of motif 1?
29 length(gadem@motifList[[1]]@alignList)

```

Listing 20. rGADEM: de novo motif discovery

After rGADEM returns its results, the user can use MotIV package³⁶⁻³⁹ to start the motif matching analysis (line 4 of Listing 21). 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, main="", bysim=TRUE, cex=0.3)
6
7 # visualize the quality of the results around the alignments
8 # E-value give an estimation of the match.
9 alignment <- viewAlignments(analysis.jaspar)
10 print(alignment[[1]])

```

Listing 21. MotIV: motifs matches analysis

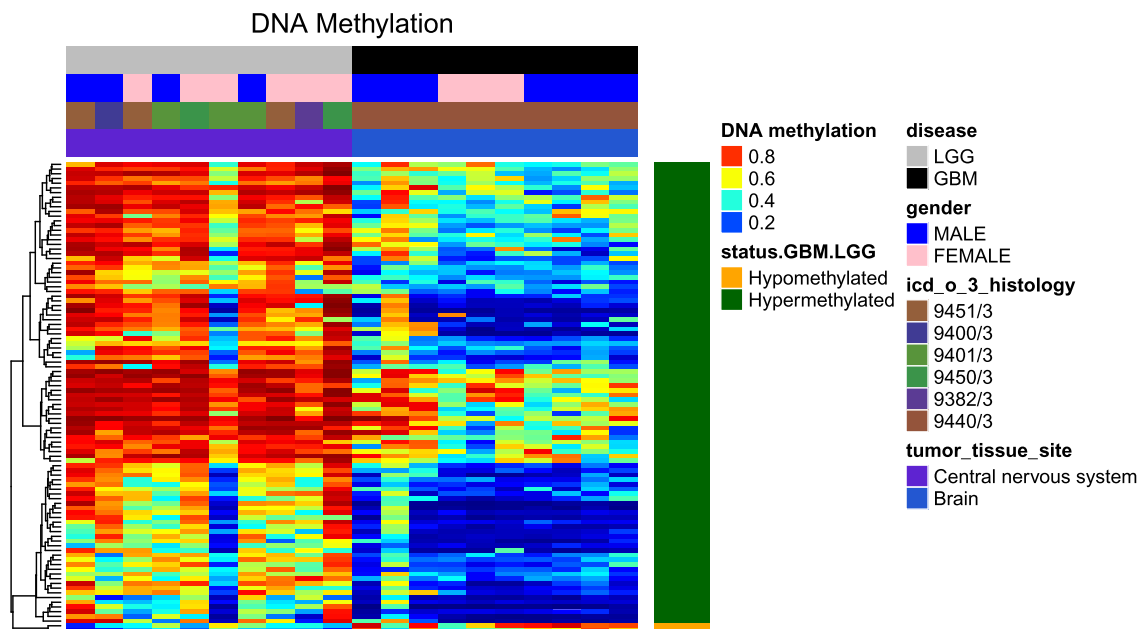


Figure 10. Heatmap of DNA methylation in probes. Rows are probes and columns are samples (patients). The DNA methylation values ranges from 0.0 (completely DNA unmethylated, blue) to 1.0 (completely DNA methylated, red). The groups of each sample were annotated in the top bar and the DNA methylation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

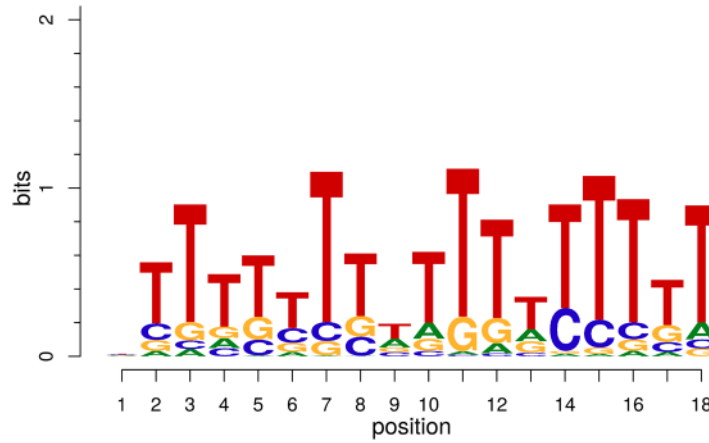


Figure 11. Motif logos found during *de-novo* motif analysis.

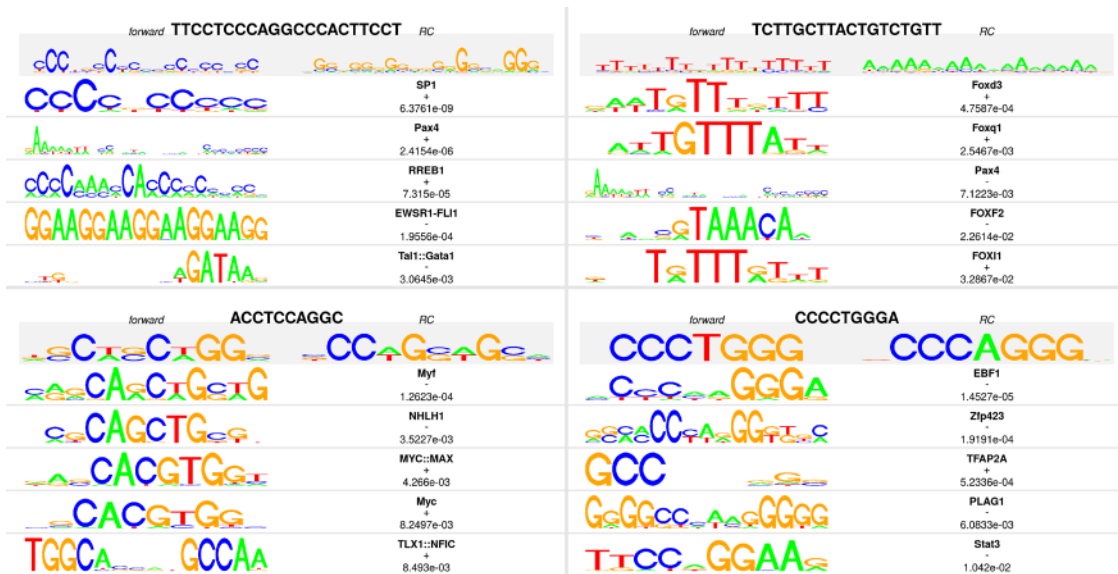


Figure 12. Identified transcription factors: the sequence logo, the name of the motif match and the p-value of the alignment.

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 throughput data^{29,40,41}. 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 possible question one might ask is whether nearby genes also undergo a change in its expression, either an increase or a decrease. DNA methylation at promoters of genes have 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 applied for a study of DNA methylation and gene expression⁴². 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 samples⁴³.

The function `TCGAVisualize_starburst` creates a Starburst plot for comparison of DNA methylation and gene expression. 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 of 10^{-2} for the DNA methylation data. The Starburst plot for the Listing 22 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.

```

1 #----- Starburst plot -----
2 starburst <- TCGAVisualize_starburst(met.chr9,          # DNA methylation with
results
3
4                               dataDEGs,              # DEG results
5                               "GBM", "LGG",          # Groups
6                               filename = "starburst.png",
7                               met.p.cut = 10^-2,
8                               exp.p.cut = 10^-2,
9                               diffmean.cut = 0.25,
10                              logFC.cut = 1,
11                              width = 15, height = 10,
names = TRUE)

```

Listing 22. Starburst plot for comparison of DNA methylation and gene expression

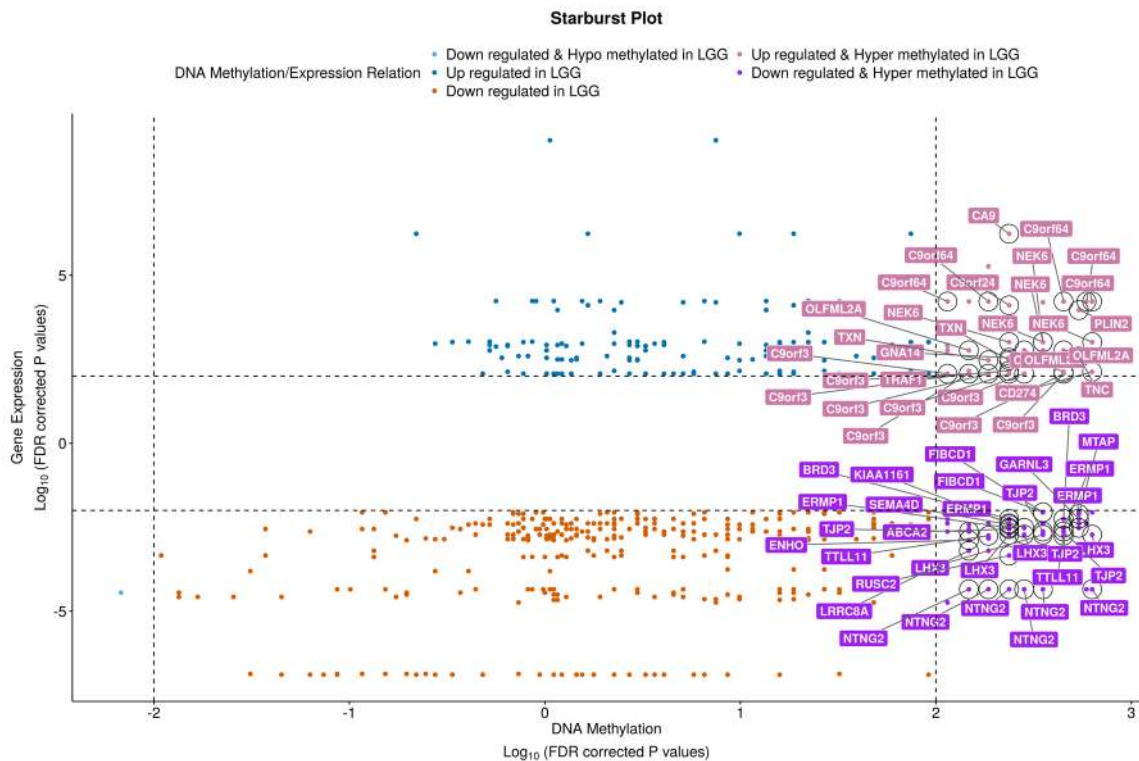


Figure 13. Starburst plot: x-axis is the log10 of the correct P-value for DNA methylation and the y-axis is the log10 of the correct P-value for the expression data. The starburst plot highlights nine distinct quadrants. Highlighted genes might have the potential for activation due to epigenetic alterations.

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 exist in the ROADMAP database and can be obtained through the [AnnotationHub](#) package⁴⁴ 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.

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 article⁵³.

The first step shown in [Listing 23](#) 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 (a table for the list is found in this [summary table](#))⁵⁴.and narrowPeak is select ing the type of file. The data downloaded are processed data from an integrative Analysis of 111 reference human epigenomes⁵⁴.

```

1 library(AnnotationHub)
2 library(pbapply)
3 #----- Working with ChipSeq data -----
4 # Step 1: download histone marks for a brain samples.
5 #-----
6 ah = AnnotationHub() # loading annotation hub database
7
8 # Searching for brain consolidated epigenomes in the roadmap database
9 bpChipEpi_brain <- query(ah, c("EpigenomeRoadMap", "narrowPeak", "chip", "consolidated",
10 "brain", "E068"))
11 # Get chip-seq data
12 histone.marks <- pblapply(names(bpChipEpi_brain), function(x){ah[[x]]})
13 names(histone.marks) <- names(bpChipEpi_brain)

```

Listing 23. Download chip-seq data

Table 4. Histone marks that define genomic elements.

Histone marks	Role
Histone H3 lysine 4 trimethylation (H3K4me3)	Promoter regions ^{45,46}
Histone H3 lysine 4 monomethylation (H3K4me1)	Enhancer regions ⁴⁵
Histone H3 lysine 36 trimethylation (H3K36me3)	Transcribed regions
Histone H3 lysine 27 trimethylation (H3K27me3)	Polycomb repression ⁴⁷
Histone H3 lysine 9 trimethylation (H3K9me3)	Heterochromatin regions ⁴⁸
Histone H3 acetylated at lysine 27 (H3K27ac)	Increase activation of genomic elements ⁴⁹⁻⁵¹
Histone H3 lysine 9 acetylation (H3K9ac)	Transcriptional activation ⁵²

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 $-\log_{10}(\text{p-value})$ 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

The [ChIPseeker](#) package⁵⁵ 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 23](#), 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 24](#) shows an example of code to plot the average profile. [Figure 14](#) shows the result.

To help the user understand better the regions found in the DMR analysis, we downloaded histone marks specific for brain tissue, which was done using the AnnotationHub package that can access Roadmap datababse ([Listing 23](#)). After, the ChIPseeker was used to visualize how histone modifications are enriched to hypomethylated and hypermethylated regions, ([Listing 24](#)). The enrichment heatmap and the average profile of peaks binding to those region 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 not 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.

```

1 library(ChIPseeker)
2 library(pbapply)
3 library(SummarizedExperiment)
4 library(GenomeInfoDb)
5 library(ggplot2)
6 library(AnnotationHub)
7
8 # Create a GR object based on the hypo/hypermethylated probes.
9 probes <- keepStandardChromosomes(rowRanges(met.chr9)[values(met.chr9)$status.GBM.LGG
%in% c("Hypermethylated", "Hypomethylated"),])
10 # Defining a window of 3kbp - 3kbp_probe_3kbp
11 probes@ranges <- IRanges(start = c(probes@ranges@start - 3000), end = c(probes@
ranges@start + 3000))

```

```

12
13 ### Profile of ChIP peaks binding to TSS regions
14 # First of all, for calculate the profile of ChIP peaks binding to TSS regions, we should
15 # prepare the TSS regions, which are defined as the flanking sequence of the TSS sites.
16 # Then align the peaks that are mapping to these regions, and generate the tagMatrix.
17 tagMatrixList <- pblapply(histone.marks, function(x) {
18   getTagMatrix(keepStandardChromosomes(x), windows = probes, weightCol = "score")
19 })
20 names(tagMatrixList) <- basename(bpChipEpi_brain$title)
21 names(tagMatrixList) <- gsub(".narrowPeak.gz","",names(tagMatrixList)) # remove file
type from name
22 names(tagMatrixList) <- gsub("E068-", "",names(tagMatrixList)) # remove file type from name
23
24 pdf("chip_heatmap.pdf", height = 5, width = 10)
25 tagHeatmap(tagMatrixList, xlim=c(-3000, 3000),color = NULL)
26 dev.off()
27
28 p <- plotAvgProf(tagMatrixList, xlim = c(-3000,3000), xlab = "Genomic Region (5'→3',
centered on CpG)")
29 # We are centreing in the CpG instead of the TSS. So we'll change the labels manually
30 p <- p + scale_x_continuous(breaks=c(-3000,-1500,0,1500,3000),labels=c(-3000,-1500,"CpG",1500,3000))
31 library(ggthemes)
32 pdf("chip-seq.pdf", height = 5, width = 7)
33 p + theme_few() + scale_colour_few(name="Histone marks") + guides(colour = guide_legend
(override.aes = list(size=4)))
34 dev.off()

```

Listing 24. 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.

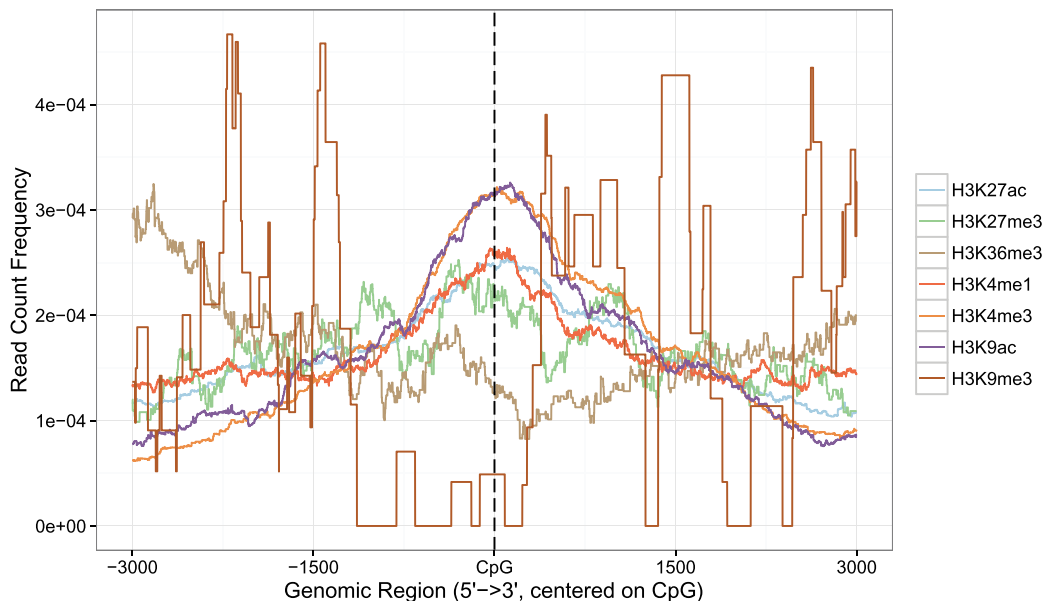


Figure 14. Average profiles for histone marks H3K27ac, H3K27me3, H3K36me3, H3K4me1, H3K4me3, H3K9ac, H3K9me3. The figure indicates that the differentially methylated regions overlaps regions of enhancers, promoters and increased activation of genomic elements.

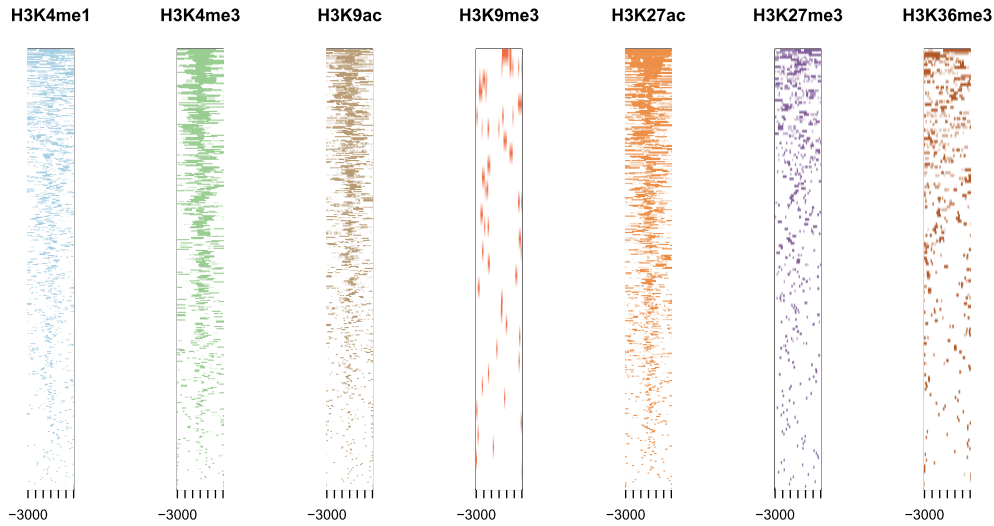


Figure 15. Heatmap of histone marks H3K4me1, H3K4me3, H3K27ac, H3K9ac, H3K9me3, H3K27me3 and H3K36me3 for brain tissues. The figure indicates that the most of the peaks that overlaps the probes are not brain specific.

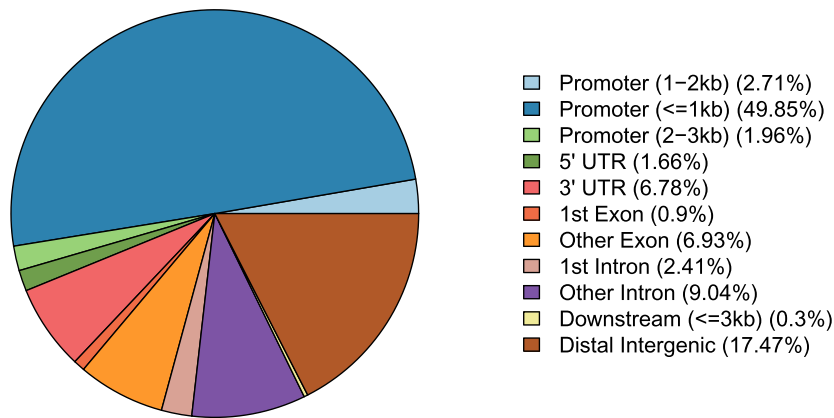


Figure 16. Feature distribution: annotation the region of the probes closer to the H3K4me3 peaks.

```

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

```

Listing 25. Annotate the location of a given peak in terms of genomic features

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 related to diseases⁵⁶⁻⁵⁹. In order to investigate regulatory enhancers that can be located at long distances upstream or downstream of target genes bioconductor offers 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 sets⁶⁰.

ELMER analysis has five 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 TCGAAbiolinks 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 $\log_2(\text{expression} + 1)$ 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 26** shows how to use TCGAAbiolinks⁷ 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. These samples are the same used in the previous steps.

```

1 #----- 8.3 Identification of Regulatory Enhancers -----
2 library(TCGAAbiolinks)
3 library(stringr)
4 # Samples: primary solid tumor with both DNA methylation and gene expression
5 matched_met_exp <- function(tumor, n = NULL){
6   # get primary solid tumor samples: DNA methylation
7   met450k <- TCGAquery(tumor = tumor, "HumanMethylation450", level = 3)
8   met450k.tp <- TCGAquery_SampleTypes(unique(unlist(stringr::str_split(met450k$
barcode, ","))), c("TP"))
9
10  # get primary solid tumor samples: RNAseq
11  rnaseq <- TCGAquery(tumor = tumor, c("IlluminaHiSeq_RNASeqV2"), level = 3)
12  rnaseq.tp <- TCGAquery_SampleTypes(unique(unlist(stringr::str_split(rnaseq$
barcode, ","))), c("TP"))
13
14  # Get samples with samples in both platforms
15  samples <- unique(substr(rnaseq.tp, 1, 15) [substr(rnaseq.tp, 1, 12) %in%
substr(met450k.tp, 1, 12)] )
16  if(!is.null(n)) samples <- samples [1:n] # get only n samples
17  return(samples)
18 }
19 lgg.samples <- matched_met_exp("LGG", n = 10)
20 gbm.samples <- matched_met_exp("GBM", n = 10)
21 samples <- c(lgg.samples, gbm.samples)
22

```

```

23 #-----
24 # 1 - Methylation
25 # -----
26 # For methylation it is quicker in this case to download the tar.gz file
27 # and get the samples we want instead of downloading files by files
28 query.met <- TCGAquery(tumor = c("GBM","LGG"), platform = "HumanMethylation450",
level = 3, samples = samples)
29 TCGAdownload(query.met, samples = samples)
30 met.elmer <- TCGAprepare(query.met, dir = ".", save = FALSE, samples = samples)
31 met.elmer <- TCGAprepare_elmer(met.elmer, platform = "HumanMethylation450")
32
33 #-----
34 # 2 - Expression
35 # -----
36 query.rna <- TCGAquery(tumor = c("GBM","LGG"), platform = "IlluminaHiSeq_RNASeqV2",
level = 3)
37 TCGAdownload(query.rna, samples = samples, type = "rsem.genes.normalized_results")
38 exp.elmer <- TCGAprepare(query.rna, dir = ".", ,type = "rsem.genes.normalized_results",
39 samples = samples, save = FALSE)
40 exp.elmer <- TCGAprepare_elmer(exp.elmer, platform = "IlluminaHiSeq_RNASeqV2")

```

Listing 26. Preparing TCGA data for ELMER's mee object

Finally, the ELMER input is a mee object that contains a DNA methylation matrix, a 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 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 two groups the experiment group (LGG) and control samples (GBM).

```

1 library(ELMER)
2 geneAnnot <- txs()
3 geneAnnot$GENEID <- paste0("ID",geneAnnot$GENEID)
4 geneInfo <- promoters(geneAnnot,upstream = 0, downstream = 0)
5 probe <- get.feature.probe()
6
7 # create mee object, use @ to access the matrices inside the object
8 mee <- fetch.mee(meth = met, exp = exp, TCGA = TRUE, probeInfo = probe, geneInfo =
geneInfo)
9
10 # Get gbm barcodes
11 samples <- unlist(str_split(TCGAquery("gbm", "HumanMethylation450", level =
3)$barcode, ","))
12
13 # Relabel GBM samples in the mee object: GBM is control
14 mee@sample$TN[mee@sample$ID %in% lgg.samples] <- "Control"

```

Listing 27. 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.

```

1 library(parallel)
2 # Available directions are hypo and hyper, we will use only hypo
3 # due to speed constraint
4 direction <- c("hypo")
5
6 for(j in direction){
7   print(j)
8   dir.out <- paste0("elmer/",j)
9   dir.create(dir.out, recursive = TRUE)
10  #-----
11  # STEP 3: Analysis |
12  #-----
13  # Step 3.1: Get diff methylated probes |
14  #-----
15  Sig.probes <- get.diff.meth(mee, cores=detectCores(),
16                             dir.out =dir.out,
17                             diff.dir=j,
18                             pvalue = 0.01)
19
20  #-----
21  # Step 3.2: Identify significant probe-gene pairs |
22  #-----
23  # Collect nearby 20 genes for Sig.probes
24  nearGenes <- GetNearGenes(TRange=getProbeInfo(mee, probe=Sig.probes$probe),
25                             cores=detectCores(),
26                             geneAnnot=getGeneInfo(mee))
27
28  pair <- get.pair(mee=mee,
29                  probes=Sig.probes$probe,
30                  nearGenes=nearGenes,
31                  permu.dir=paste0(dir.out,"/permu"),
32                  dir.out=dir.out,
33                  cores=detectCores(),
34                  label= j,
35                  permu.size=100, # For significant results use 10000
36                  Pe = 0.01) # For significant results use 0.001
37
38  Sig.probes.paired <- fetch.pair(pair=pair,
39                                 probeInfo = getProbeInfo(mee),
40                                 geneInfo = getGeneInfo(mee))
41  Sig.probes.paired <-read.csv(paste0(dir.out,"/getPair.",j,".pairs.significant.csv"),
42                              stringsAsFactors=FALSE)[,1]
43
44
45  #-----
46  # Step 3.3: Motif enrichment analysis on the selected probes |
47  #-----
48  if(length(Sig.probes.paired) > 0 ){
49    #-----
50    # Step 3.3: Motif enrichment analysis on the selected probes |
51    #-----
52    enriched.motif <- get.enriched.motif(probes=Sig.probes.paired,
53                                         dir.out=dir.out, label=j,
54                                         background.probes = probe$name)
55    motif.enrichment <- read.csv(paste0(dir.out,"/getMotif.",j,".motif.
enrichment.csv"),
56                                 stringsAsFactors=FALSE)
57    if(length(enriched.motif) > 0){

```

```

58     #-----
59     # Step 3.4: Identifying regulatory TFs |
60     #-----
61     print("get.TFs")
62
63     TF <- get.TFs(mee = mee,
64                 enriched.motif = enriched.motif,
65                 dir.out = dir.out,
66                 cores = detectCores(), label = j)
67     TF.meth.cor <- get(load(paste0(dir.out, "/getTF.", j, ".TFs.with.motif.
pvalue.rda")))
68     save(TF, enriched.motif, Sig.probes.paired,
69         pair, nearGenes, Sig.probes, motif.enrichment, TF.meth.cor,
70         file=paste0(dir.out, "/ELMER_results_", j, ".rda"))
71     }
72 }
73 }

```

Listing 28. 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 found for the hyper direction (probes hypomethylated in LGG group compared to the GBM group) is found in [Figure 17](#). To select the motifs we select the 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 values are the default from the ELMER package.

The analysis found 14 enriched motifs for the hyper direction and no enriched motifs for the hypo direction.

After finding these list of enriched motifs, ELMER identifies regulatory TFs whose expression associate with DNA methylation at motifs and for each enriched motif a TF ranking plot is created automatically by ELMER. This plot shows the TF ranking plots based on the score ($-\log(Pvalue)$) of association between TF expression and DNA methylation of the motif. We can see in [Figure 18](#) that the top three associated TFs that are associated with that AP1 motif are POLR3K, DLX3 and NEUROD2.

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 on the three most relevant transcription factors. For example, for the AP1 motif the average DNA methylation level of sites with the AP1 motif plotted against the expression of the transcription factors WT1, ZNF208, ATF4 and DDX5 is show in [Figure 19](#). We can see that the experiment group (GBM samples) has a lower average methylation level of sites with the AP1 motif plotted and a higher expression of the transcription factors.

```

1 scatter.plot(mee, category="TN", save=T, lm_line=TRUE,
2             byTF=list(TF=c("HOXA5", "TGIF1", "HOXA6", "FOSL1"), probe=enriched.
motif[["AP1"]]))

```

Listing 29. Visualizing the average DNA methylation level of sites with a chosen motif vs TF expression

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 transcription factor versus the 30% with lower expression. The code below shows that analysis.

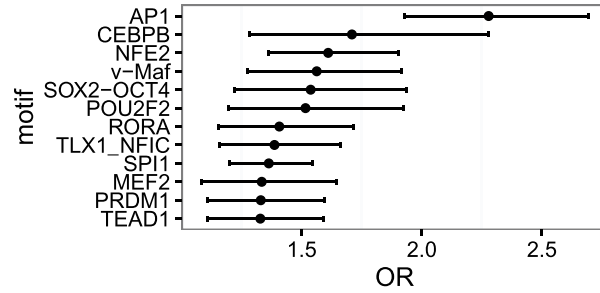


Figure 17. The plot shows the Odds Ratio (x axis) for the selected motifs. The range shows the 95% confidence interval for each Odds Ratio.

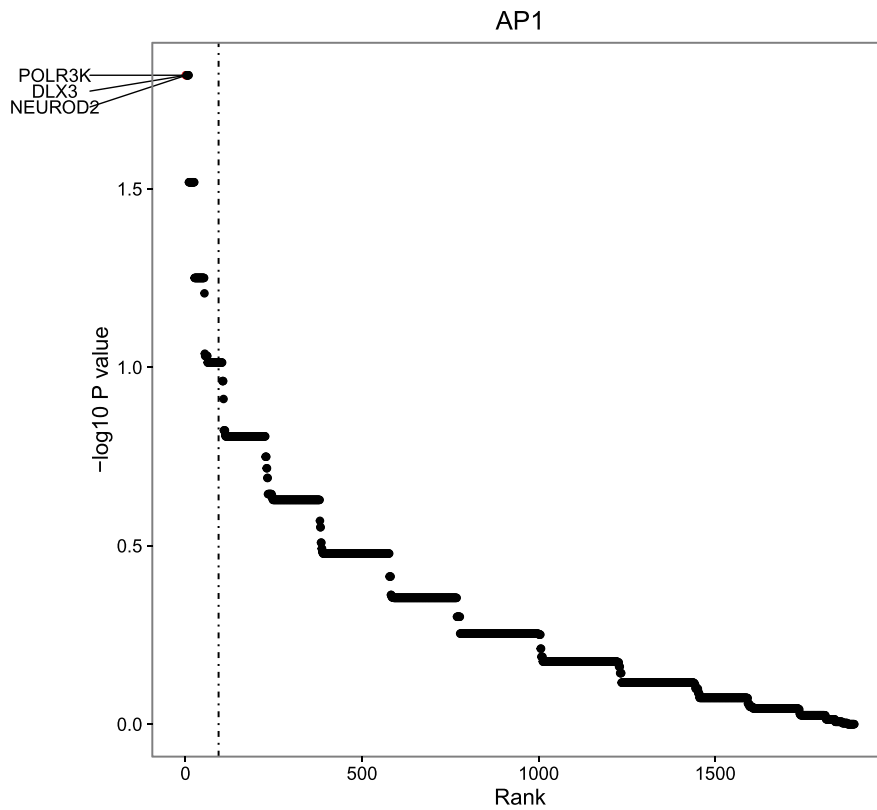


Figure 18. TF ranking plots based on the score ($-\log(Pvalue)$) of association between TF expression and DNA methylation of the AP1 motif. The dashed line indicates the boundary of the top 5% association score and the TF within this boundary were considered candidate upstream regulators. The top 3 associated TFs and the TF family members (dots in red) that are associated with that specific motif are labeled in the plot.

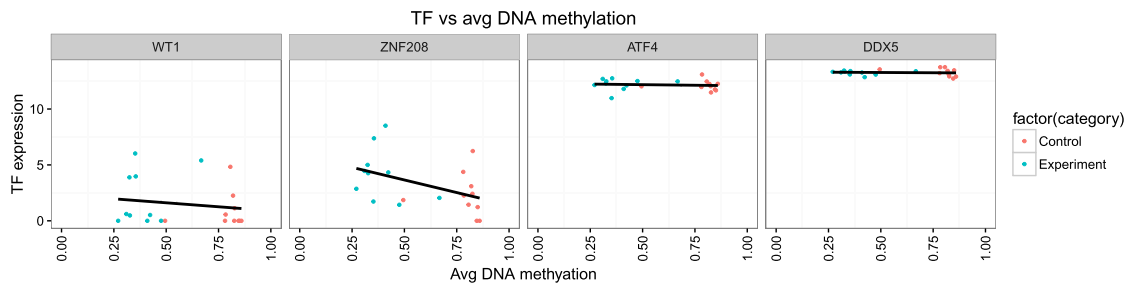


Figure 19. Each scatter plot shows the average DNA methylation level of sites with the AP1 motif plotted against the expression of the transcription factors HOXA5, TGIF1, HOXA6 and FOSL1 respectively.

```

1 TCGAsurvival_TFplot <- function(TF,mee, clinical, percentage = 0.3){
2
3   # For the transcription factor, gets it getGeneID
4   gene <- getGeneID(mee,symbol=TF)
5   # Get the expression values for the genes.
6   #(getExp is a ELMER function)
7   exp <- getExp(mee,geneID=gene)
8
9   # Get the names of the 30% patients with lower expression
10  g1 <- names(sort(exp)[1:ceiling(length(exp) * percentage)])
11
12  # Get the names of the 30% patients with higher expression
13  g2 <- names(sort(exp, decreasing = T)[1:ceiling(length(exp)*percentage)])
14
15  # get the data of only these patients
16  clinical <- clinical[clinical$bcr_patient_barcode %in% substr(c(g1, g2),1,12),]
17
18  # Create the labels for each sample
19  clinical$tf_groups <- "high"
20  clinical[clinical$bcr_patient_barcode %in% substr(c(g1) ,1 ,12),]$tf_groups <- "low"
21
22  # Use TCGAbiolinks to create the survival curve
23  TCGAanalyze_survival(clinical, "tf_groups",
24                       legend=paste0(TF," Exp level"),
25                       filename = paste0(TF,".png"))
26 }
27
28 # get clinical patient data for GBM samples
29 gbm_clin <- TCGAquery_clinic("gbm","clinical_patient")
30
31 # get clinical patient data for LGG samples
32 lgg_clin <- TCGAquery_clinic("lgg","clinical_patient")
33
34 # Bind the results, as the columns might not be the same,
35 # we will will plyr rbind.fill, to have all columns from both files
36 clinical <- plyr::rbind.fill(gbm_clin,lgg_clin)
37 # Call the function we created
38 TCGAsurvival_TFplot("FOXP4",mee,clinical)
39 TCGAsurvival_TFplot("FOXE3",mee,clinical)

```

Listing 30. Survival analysis for samples with lower expression of regulatory TF and higher expression

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

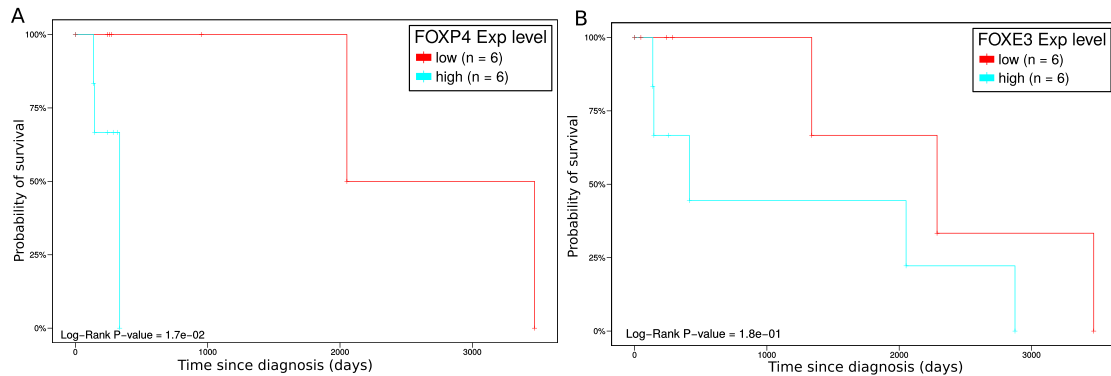


Figure 20. **A)** Survival plot for the 30% patients with high expression and low expression of FOXP4 TF. **B)** Survival plot for the 30% patients with high expression and low expression of FOXE3 TF.

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 with the results obtained from analyzing TCGA data in order 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 31](#) shows how to install all the required packages.

```

1 source("https://bioconductor.org/biocLite.R")
2 packages <- c("TCGAbiolinks", "ELMER", "gaia", "ChIPseeker", "AnnotationHub",
3             "ComplexHeatmap", "clusterProfiler", "RTCGAtoolbox",
4             "minet", "biomaRt", "pathview", "MotifDb", "MotIV", "motifStack"
5             , "rGADEM")
6 new.packages <- packages[!(packages %in% installed.packages()[, "Package"])]
7 if (length(new.packages)) biocLite(new.packages)
8 if (!require("dnet")) install.packages("dnet")
9 if (!require("circlize")) install.packages("circlize")
10 if (!require("VennDiagram")) install.packages("VennDiagram")
11 if (!require("c3net")) install.packages("c3net")
12 if (!require("pbapply")) install.packages("pbapply")
13 if (!require("gplots")) install.packages("gplots")

```

Listing 31. Installing packages

All data used in this workflow is freely available and can be accessed using a R/Bioconductor package. There are two main sources of data: [The Cancer Genome Atlas \(TCGA\)](#) and a [supplementary data repository](#) with processed datasets from the Roadmap Epigenomics Project and from The Encyclopedia of DNA Elements (ENCODE) project⁵⁴. For the first, a summary of the data available can be seen in <https://tcga-data.nci.nih.gov/tcga/> 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.

Session information

```

1 R version 3.3.0 (2016-05-03)
2 Platform: x86_64-pc-linux-gnu (64-bit)
3 Running under: Ubuntu 16.04 LTS
4
5 locale:
6  [1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C
7  [3] LC_TIME=en_US.UTF-8      LC_COLLATE=en_US.UTF-8
8  [5] LC_MONETARY=en_US.UTF-8  LC_MESSAGES=en_US.UTF-8
9  [7] LC_PAPER=en_US.UTF-8     LC_NAME=C
10 [9] LC_ADDRESS=C             LC_TELEPHONE=C
11 [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
12
13 attached base packages:
14 [1] grid      stats4    parallel  stats     graphics  grDevices  utils
15 [8] datasets  methods  base
16
17 other attached packages:
18 [1] knitr_1.13
19 [2] ggthemes_3.0.3
20 [3] scales_0.4.0
21 [4] survival_2.39-4
22 [5] AnnotationHub_2.4.2
23 [6] ggplot2_2.1.0
24 [7] gplots_3.0.1
25 [8] pbapply_1.2-1
26 [9] circlize_0.3.7
27 [10] dnet_1.0.9
28 [11] supraHex_1.10.0
29 [12] hexbin_1.27.1
30 [13] motifStack_1.16.2
31 [14] ade4_1.7-4
32 [15] grImport_0.9-0
33 [16] XML_3.98-1.4
34 [17] rGADEM_2.20.0
35 [18] seqLogo_1.38.0
36 [19] MotIV_1.28.0
37 [20] MotifDb_1.14.0
38 [21] BSgenome.Hsapiens.UCSC.hg19_1.4.0
39 [22] BSgenome_1.40.0
40 [23] rtracklayer_1.32.0
41 [24] pathview_1.12.0
42 [25] biomaRt_2.28.0
43 [26] downloader_0.4
44 [27] c3net_1.1.1
45 [28] igraph_1.0.1
46 [29] minet_3.30.0
47 [30] clusterProfiler_3.0.2
48 [31] DOSE_2.10.2
49 [32] ComplexHeatmap_1.11.1
50 [33] ChIPseeker_1.8.3

```


51 [34] [gaia_2.16.0](#)
 52 [35] [ELMER_1.4.2](#)
 53 [36] [ELMER.data_1.2.2](#)
 54 [37] [Homo.sapiens_1.3.1](#)
 55 [38] [TxDb.Hsapiens.UCSC.hg19.knownGene_3.2.2](#)
 56 [39] [org.Hs.eg.db_3.3.0](#)
 57 [40] [GO.db_3.3.0](#)
 58 [41] [OrganismDbi_1.14.1](#)
 59 [42] [GenomicFeatures_1.24.2](#)
 60 [43] [annotationDbi_1.34.3](#)
 61 [44] [IlluminaHumanMethylation450kanno.ilmn12.hg19_0.2.1](#)
 62 [45] [minfi_1.18.2](#)
 63 [46] [bumphunter_1.12.0](#)
 64 [47] [locfit_1.5-9.1](#)
 65 [48] [iterators_1.0.8](#)
 66 [49] [foreach_1.4.3](#)
 67 [50] [Biostrings_2.40.1](#)
 68 [51] [XVector_0.12.0](#)
 69 [52] [lattice_0.20-33](#)
 70 [53] [SummarizedExperiment_1.2.2](#)
 71 [54] [Biobase_2.32.0](#)
 72 [55] [GenomicRanges_1.24.1](#)
 73 [56] [GenomeInfoDb_1.8.1](#)
 74 [57] [IRanges_2.6.0](#)
 75 [58] [S4Vectors_0.10.1](#)
 76 [59] [BiocGenerics_0.18.0](#)
 77 [60] [stringr_1.0.0](#)
 78 [61] [TCGAbiolinks_1.3.5](#)

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.

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.

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.* 2015; pii bbv080.
[PubMed Abstract](#) | [Publisher Full Text](#)
7. Colaprico A, Silva TC, Olsen C, et al.: **Tcgbiolinks: 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](#)
8. 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](#)
9. Morgan HJM, Obenchain V, Pagès H: **SummarizedExperiment: SummarizedExperiment container.** R package version 1.1.0.
[Reference Source](#)
10. Ceccarelli M, Barthel FP, Tathiane M, 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](#)
11. 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](#)
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. 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](#)
15. Samur MK: **Rtcgatoobox: A new Tool for exporting TCGA Firehose data.** *PLoS One.* 2014; 9(9): e106397.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
16. 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](#)
17. Morganella S, Pagnotta SM, Ceccarelli M: **Gaia: Genomic analysis of important aberrations.**
[Reference Source](#)
18. Gu Z: **Complexheatmap: Making complex heatmaps.** R package version 1.7.1.
[Reference Source](#)
19. 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](#)
20. 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](#)
21. 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](#)
22. 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](#)
23. 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](#)
24. 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](#)
25. 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](#)
26. 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](#)
27. 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](#)
28. 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](#)
29. Phillips T: **The role of methylation in gene expression.** *Nature Education.* 2008; 1(1): 116.
[Reference Source](#)
30. 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](#)
31. Robertson KD: **DNA methylation and human disease.** *Nat Rev Genet.* 2005; 6(8): 597–610.
[PubMed Abstract](#) | [Publisher Full Text](#)
32. 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](#)
33. Droit A, Gottardo R, Robertson G, et al.: **rgadem: De novo motif discovery.** 2015.
[Reference Source](#)
34. 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](#)
35. 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](#)
36. Mercier E, Gottardo R: **Motiv: Motif identification and validation.** 2014.
[Reference Source](#)
37. 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](#)
38. 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](#)
39. 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](#)
40. 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](#)
41. Rhodes DR, Chinnaiyan AM: **Integrative analysis of the cancer transcriptome.** *Nat Genet.* 2005; 37(Suppl): S31–S37.
[PubMed Abstract](#) | [Publisher Full Text](#)
42. 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](#)
43. 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](#)

44. Tenenbaum D, Morgan M, Carlson M, *et al.*: **Annotationhub: Client to access annotationhub resources.** r package version 2.2.2. [Reference Source](#)
45. 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](#)
46. 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](#)
47. 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](#)
48. Antoine HF, 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](#)
49. 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](#)
50. 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](#)
51. 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](#)
52. 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](#)
53. Pekowska A, Anders S: **ChIP-seq analysis basics.** 2015. [Reference Source](#)
54. Roadmap Epigenomics Consortium, Kundaje A, Meuleman W, *et al.*: **Integrative analysis of 111 reference human epigenomes.** *Nature.* 2015; **518**(7539): 317–330. [PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
55. Yu G, Wang LG, He QY, *et al.*: **ChIPseeker: an R/Bioconductor package for ChIP peak annotation, comparison and visualization.** *Bioinformatics.* 2015; **31**(14): 2382–2383. [PubMed Abstract](#) | [Publisher Full Text](#)
56. 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 leukodystrophy (add).** *Hum Mol Genet.* 2015; page ddv065.
57. Gröschel S, Sanders MA, Hoogenboezem R, *et al.*: **A single oncogenic enhancer rearrangement causes concomitant *EV1* and *GATA2* deregulation in leukemia.** *Cell.* 2014; **157**(2): 369–381. [PubMed Abstract](#) | [Publisher Full Text](#)
58. 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](#)
59. 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](#)
60. Yao L, Shen H, Laird PW, *et al.*: **Inferring regulatory element landscapes and transcription factor networks from cancer methylomes.** *Genome Biol.* 2015; **16**(1): 105–105. [PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)

Open Peer Review

Current Peer Review Status:



Version 1

Reviewer Report 21 September 2016

<https://doi.org/10.5256/f1000research.9601.r15858>

© 2016 Papaleo E. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



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 too many adjectives. Few lines below "we provide a workflow 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).

Competing Interests: No competing interests were disclosed.

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

Author Response 28 Dec 2016

Tiago Chedraoui Silva, Ribeirao Preto Medical School, University of Sao Paulo, Ribeirao Preto, 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/affyQCReport.pdf>. 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.

Reviewer Report 18 July 2016

<https://doi.org/10.5256/f1000research.9601.r14695>

© 2016 Ellrott K. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



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 were 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.

Competing Interests: No competing interests were disclosed.

I confirm that I have read this submission and 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.

Author Response 28 Dec 2016

Tiago Chedraoui Silva, Ribeirao Preto Medical School, University of Sao Paulo, Ribeirao Preto, 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.

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com

F1000Research