Exploiting the DepMap cancer dependency data using the depmap R package

Theo Killian[1,2] and Laurent Gatto[1]

[1]**Computational Biology and Bioinformatics Unit, de Duve Institute, UCLouvain, Brussels, Belgium.**
[2]**Current address: VIB-KULeuven Center for Cancer Biology, Leuven, Belgium.**

laurent.gatto@uclouvain.be

Abstract

The depmap package facilitates access in the R environment to the data from the DepMap project, a multi-year collaborative effort by the Broad Institute and Wellcome Sanger Institute, mapping genetic and chemical dependencies and other molecular biological measurements of over 1700 cancer cell lines. The depmap package formats this data for use of popular R data analysis and visualizing tools such as dplyr and ggplot2. In addition, the depmap package utilizes ExperimentHub, storing versions of the DepMap data accessible from the Cloud, which may be selectively downloaded, providing a reproducible research framework to support exploiting this data. This paper describes a workflow demonstrating how to access and visualize the DepMap data in R using this package.

Keywords

cancer, cancer dependency, DepMap, ExperimentHub, data mining, reproducible research, Bioconductor

Introduction

The consequences of genomic alterations of cancer cells on the molecular biological landscape of the cell may result in differential vulnerabilities, or “dependencies” compared to those of healthy cells. An example of genetic dependency is a gene not necessary for the survival in healthy cells, but essential for the vitality of particular cancer cell line. However, the exact biological nature of many genetic dependencies of cancer are not completely understood [1]. A map that illustrates the relationships between the genetic features of cancer and those of cancer dependencies is therefore desirable. The Cancer Dependency Map or “DepMap”, a collaborative initiative between the Broad Institute and the Wellcome Sanger Institute, aims to map genetic dependencies in a broad range of cancer cell lines. Hundreds of cancer cell lines have been selected to be tested in this effort, intended to mirror the distribution of various cancer diseases in the general population. The stated aim of the DepMap Project is developing a better understanding of the molecular biology of cancer and the exploiting this knowledge to develop new therapies in precision cancer medicine [2].

The DepMap initative is, as of the date of this publication, an ongoing project, with new data releases of select datasets every 90 days. As of the most current 20Q4 DepMap release, 1812 human cancer cell lines have been mapped for dependencies [2]. The DepMap project utilizes CRISPR gene knockout as the primary method to map genomic dependencies in cancer cell lines [2, 3, 4, 5]. The resulting genetic dependency score displayed in the DepMap data is calculated from the observed log fold change in the amount of shRNA detected in pooled cancer cell lines after gene knockout [6, 7]. To correct for potential off-target effects of gene knockout in overestimating dependency with CRISPR, the DepMap iniative utilized the CERES algorithm to moderate the final dependency score estimation [3]. It should be noted that due to advancements in the CERES algorithm to account for CRISPR seed effects, the RNAi dependency measurements have been rendered redundant, and further data releases for RNAi dependency measurement have been discontinued as of the 19Q3 release [2, 4]. In addition to genomic dependency measurements of cancer cell lines, chemical dependencies were also measured by the DepMap PRISM viability screens that as of the 20Q4 release, tested 4,518 compounds against 578 cancer cell lines [8, 2]. A new protemic dataset was added with the 20Q2 release, that provides normalized quantitative profiling of proteins of 375 cancer cell lines by mass spectrometry [9]. The DepMap project has also compiled additional datasets detailing molecular biological characterization of cancer cell lines, such as genomic copy number, Reverse Phase Protein Array (RPPA) data, TPM gene expression data for protein coding genes and genomic mutation data. These datasets are updated quarterly on a release schedule and are publically available under CC BY 4.0 licence [2].

A table of the most recent datasets available for the depmap package (as of 20Q4 release) is displayed in table 1.

The depmap Bioconductor package was created in order to maximally exploit these rich datasets and to promote reproducible research, facilitated by importing the data into the R environment. The value added by the depmap Bioconductor package includes cleaning and converting all datasets to long format tibbles [10], as well as adding the unique key depmap_id for all data tables. The addition of the the unique key depmap_id aides the comparison of molecular features and the use of common R packages such as dplyr [11] and ggplot2 [12].

As new DepMap datasets are released on a quarterly basis, it is not feasible to include all dataset files in binary directly within the directory of the depmap R package. To keep the package lightweight, the depmap package utilizes and fully depends on the ExperimentHub package [13] to store and retrieve all versions of the DepMap data (starting from 19Q1 through 20Q4) in the Cloud using AWS. The depmap package contains accessor functions to directly download and cache the most current datasets from the Cloud into the local R environment. Specific datasets (such as datasets from older releases), which can be downloaded seperately, if desired. The depmap package was designed to enhance reproducible research by ensuring datasets from all releases will remain available in this manner. The depmap R package is available as part of Bioconductor at: https://bioconductor.org/packages/depmap.

Table 1. Datasets available the depmap package. The 'Release' column indicates the most recent available release.

Dataset	Description	EH_Number	Dimensions	Coverage	Release
rnai	(DEMETER2) Batch and off-target corrected RNAi gene knockdown dependency data	EH3080	17309 genes, 712 cancer cell lines	31 primary diseases and 31 lineages	Aug 7 2019
drug	Drug sensitivity data for cancer cell lines derived from logfold change values relative to DMSO	EH3087	4686 compounds, 578 cell lines	23 primary diseases and 25 lineages	Aug 7 2019
proteomic	Normalized quantitative profiling of proteins by mass spectrometry	EH3459	12399 proteins, 375 cancer cell lines	24 primary diseases and 27 lineages	May 20 2020
crispr	(CERES) Batch and off-target corrected CRISPR-Cas9 gene knockdout dependency data	EH3960	18119 genes, 808 cell lines	31 primary diseases and 29 lineages	Nov 20 2020
copyNumber	WES log copy number data	EH3961	27562 genes, 1753 cell lines	35 primary diseases and 38 lineages	Nov 20 2020
TPM	CCLE TPM RNAseq gene expression data for protein coding genes	EH3962	19182 genes, 1376 cancer cell lines	33 primary diseases and 37 lineages	Nov 20 2020
mutationCalls	Merged mutation calls (for coding region, germline filtered)	EH3963	18789 genes, 1749 cell lines	35 primary diseases and 38 lineages	Nov 20 2020
metadata	Metadata for cell lines in the 20Q4 DepMap release	EH3964	1812 cell lines	35 primary diseases and 39 lineages	Nov 20 2020

Use cases

Dependency scores are the features of primary interest in the DepMap Project datasets. These measurements can be found in datasets `crispr` and `rnai`, which contain information on genetic dependency, as well as the dataset `drug_sensitivity`, which contains information pertaining to chemical dependency. The genetic dependency can be interpreted as an expression of how vital a particular gene for a given cancer cell line. For example, a highly negative dependency score is derived from a large negative log fold change in the population of cancer cells after gene knockout or knockdown, implying that a given cell line is highly dependent on that knocked out gene. Genes that possess highly negative dependency scores may be interesting targets for research in cancer medicine. In this workflow, we will describe exploring and visualizing several DepMap datasets, including those that contain information on genetic dependency.

Below, we start by loading the packages need to run this workflow.

```r
library("depmap")
library("ExperimentHub")
library("dplyr")
library("ggplot2")
library("stringr")
```

The `depmap` datasets are too large to be included into a typical package, therefore these data are stored in the Cloud. There are two ways to access the `depmap` datasets. The first such way calls on dedicated accessor functions that download, cache and load the latest available dataset into the R workspace. Examples for all available data are shown below:

```r
rnai <- depmap_rnai()
crispr <- depmap_crispr()
copyNumber <- depmap_copyNumber()
TPM <- depmap_RPPA()
RPPA <- depmap_TPM()
metadata <- depmap_metadata()
mutationCalls <- depmap_mutationCalls()
drug_sensitivity <- depmap_drug_sensitivity()
proteomic <- depmap_proteomic()
```

Alternatively, specific dataset (from any available release) can be accessed through Bioconductor's `ExperimentHub`. The `ExperimentHub()` function creates an `ExperimentHub` object, which can be queried for specific terms. The list of datasets available that correspond to the query, `depmap` are shown below:

```r
## create ExperimentHub query object
eh <- ExperimentHub()
query(eh, "depmap")
```

```
## ExperimentHub with 48 records
## # snapshotDate(): 2020-10-27
## # $dataprovider: Broad Institute
## # $species: Homo sapiens
## # $rdataclass: tibble
## # additional mcols(): taxonomyid, genome, description,
## #   coordinate_1_based, maintainer, rdatadateadded, preparerclass, tags,
## #   rdatapath, sourceurl, sourcetype
## # retrieve records with, e.g., 'object[["EH2260"]]'
##
##            title
##   EH2260 | rnai_19Q1
##   EH2261 | crispr_19Q1
##   EH2262 | copyNumber_19Q1
##   EH2263 | RPPA_19Q1
##   EH2264 | TPM_19Q1
##   ...      ...
##   EH5358 | crispr_21Q1
##   EH5359 | copyNumber_21Q1
##   EH5360 | TPM_21Q1
##   EH5361 | mutationCalls_21Q1
##   EH5362 | metadata_21Q1
```

Specific datasets are downloaded, cached and loaded into the workspace as tibbles by selecting each dataset by their unique EH numbers. Shown below, datasets from the 20_Q3 release are downloaded in this way.

```r
## download and cache required datasets
crispr <- eh[["EH3797"]]
copyNumber <- eh[["EH3798"]]
TPM <- eh[["EH3799"]]
mutationCalls <- eh[["EH3800"]]
metadata <- eh[["EH3801"]]
proteomic <- eh[["EH3459"]]
```

By importing the depmap data into the R environment, the data can be mined more effectively utilzing R data manipulation tools. For example, if one was interested in researching soft tissue sarcomas and wanted to search all such cancer cell lines for the gene with the greatest dependency, one could accomplish this task by using functions from the dplyr package. Below, the crispr dataset is selected for cell lines with *"SOFT_TISSUE"* in the CCLE name, and displaying a list of the highest dependency scores.

```r
## list of dependency scores
crispr %>%
  dplyr::select(cell_line, gene_name, dependency) %>%
  dplyr::filter(stringr::str_detect(cell_line, "SOFT_TISSUE")) %>%
  dplyr::arrange(dependency)
```

```
## # A tibble: 815,355 x 3
##    cell_line          gene_name dependency
##    <chr>              <chr>          <dbl>
##  1 RH18DM_SOFT_TISSUE RAN            -4.36
##  2 RH18DM_SOFT_TISSUE PSMB6          -3.82
##  3 RH18DM_SOFT_TISSUE C1orf109       -3.67
##  4 RH30_SOFT_TISSUE   RAN            -3.20
##  5 RH18DM_SOFT_TISSUE SNU13          -3.07
##  6 RH18DM_SOFT_TISSUE SPATA5L1       -3.04
##  7 RH18DM_SOFT_TISSUE HSPE1          -3.03
##  8 RH18DM_SOFT_TISSUE POLR1C         -2.96
##  9 RH18DM_SOFT_TISSUE CDC16          -2.84
## 10 RH30_SOFT_TISSUE   BUB3           -2.83
## # ... with 815,345 more rows
```

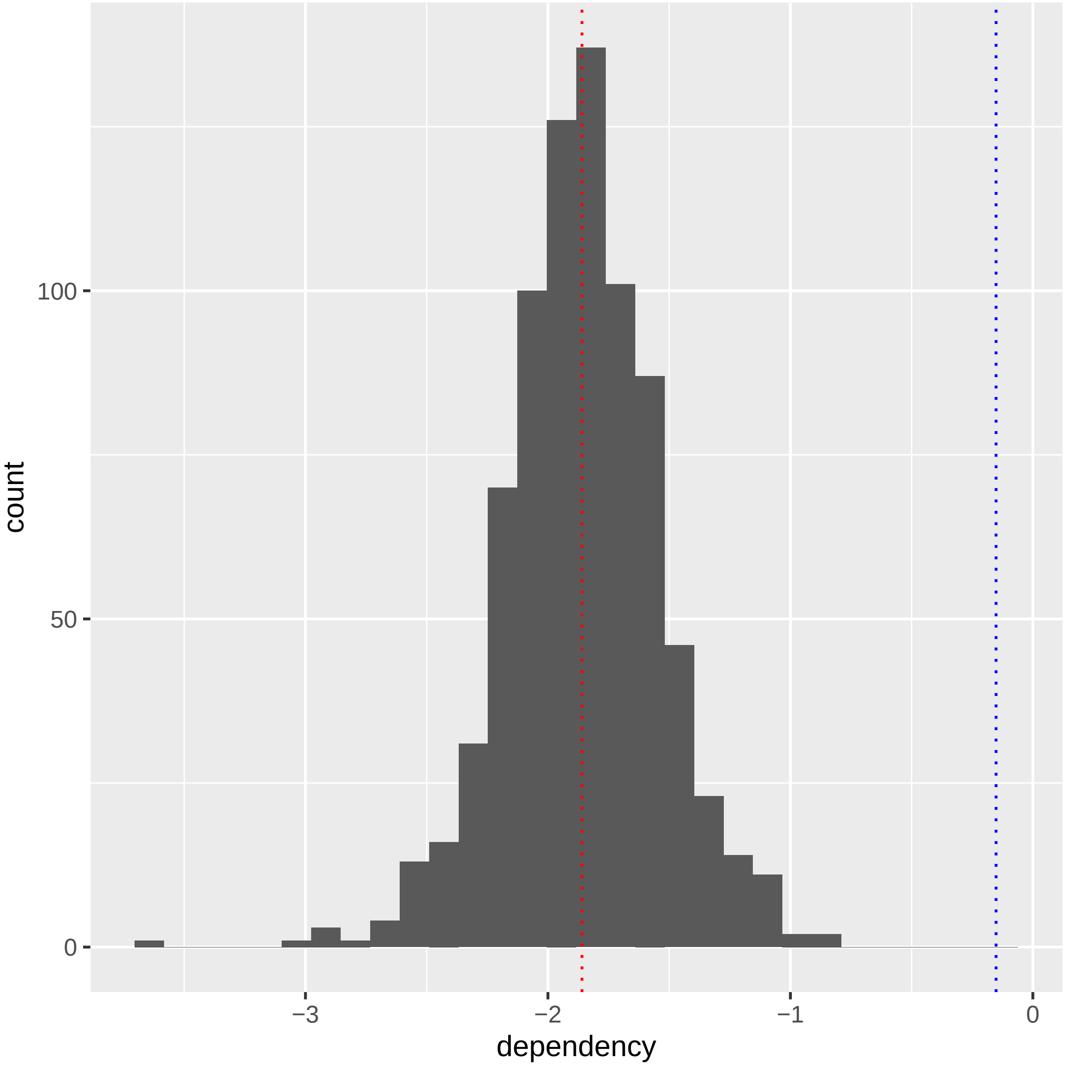

Figure 1. Histogram of CRISPR dependency scores for gene C1orf109.

The gene `C1orf109` appears in the selected list of top dependencies scores for soft tissue cancer cell lines. This gene, also known by the alias Chromosome 1 Open Reading Frame 109, codes for a poorly characterized protein which is theorized to promote cancer cell proliferation by controlling the G1 to S phase transition [14]. This protein may present as an interesting candidate target to explore and visualize the `depmap` data. Figure 1 displays the `crispr` data as a histogram showing the distribution of dependency scores for gene `C1orf109`. The red dotted line signifies the mean dependency score for that gene, while the blue dotted line signifies the global mean dependency score for all `crispr` measurements.

```
mean_crispr_dep <- crispr %>%
  dplyr::select(gene_name, dependency) %>%
  dplyr::filter(gene_name == "C1orf109")

crispr %>%
  dplyr::select(gene, gene_name, dependency) %>%
  dplyr::filter(gene_name == "C1orf109") %>%
  ggplot(aes(x = dependency)) + geom_histogram() +
  geom_vline(xintercept = mean(mean_crispr_dep$dependency, na.rm = TRUE),
             linetype = "dotted", color = "red") +
  geom_vline(xintercept = mean(crispr$dependency, na.rm = TRUE),
             linetype = "dotted", color = "blue")
```

A more complex plot of the `crispr` data, is shown below 2. Visualizing this data involves plotting the distribution of dependency scores for gene `C1orf109` for each major type of cancer, while highlighting the qualitative nature of mutations of this gene in such cancer cell lines (e.g. if such mutations are damaging or conserved, etc.). Notice that the plot above 1 reflects the same overall distribution in two dimensions.

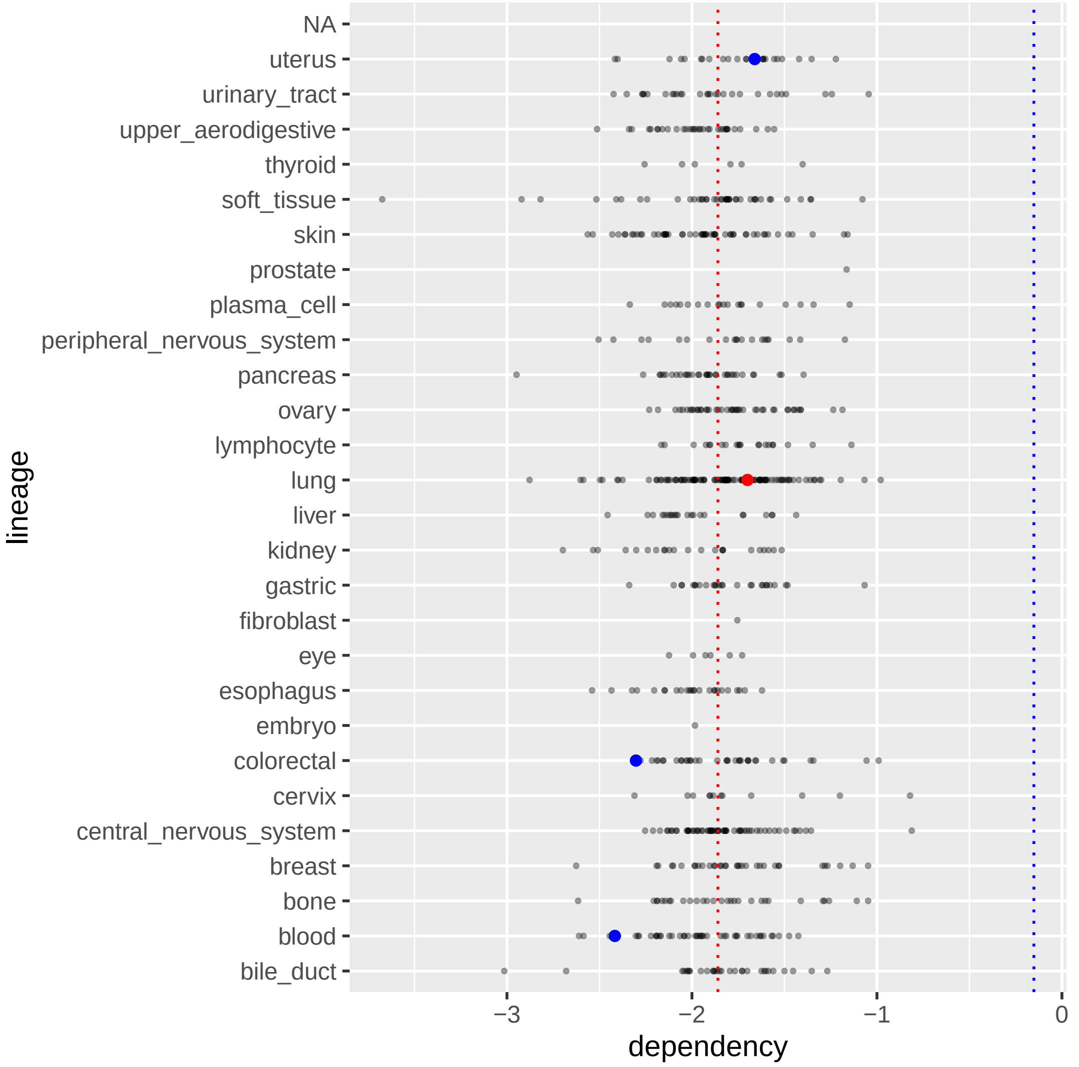

Figure 2. Plot of CRISPR dependency scores for gene C1orf109 by lineage.

```
meta_crispr <- metadata %>%
  dplyr::select(depmap_id, lineage) %>%
  dplyr::full_join(crispr, by = "depmap_id") %>%
  dplyr::filter(gene_name == "C1orf109") %>%
  dplyr::full_join((mutationCalls %>%
                      dplyr::select(depmap_id, entrez_id,
                                    is_cosmic_hotspot,
                                    var_annotation)),
                   by = c("depmap_id", "entrez_id"))

meta_crispr %>%
    ggplot(aes(x = dependency, y = lineage)) +
    geom_point(alpha = 0.4, size = 0.5) +
    geom_point(data = subset(meta_crispr,
                             var_annotation == "damaging"),
               color = "red") +
    geom_point(data = subset(meta_crispr,
                             var_annotation == "other non-conserving"),
               color = "blue") +
    geom_vline(xintercept = mean(meta_crispr$dependency, na.rm = TRUE),
               linetype = "dotted", color = "red") +
    geom_vline(xintercept = mean(crispr$dependency, na.rm = TRUE),
               linetype = "dotted", color = "blue")
```

Many cancer phenotypes are the result of changes in gene expression [15, 16, 17]. The extensive coverage

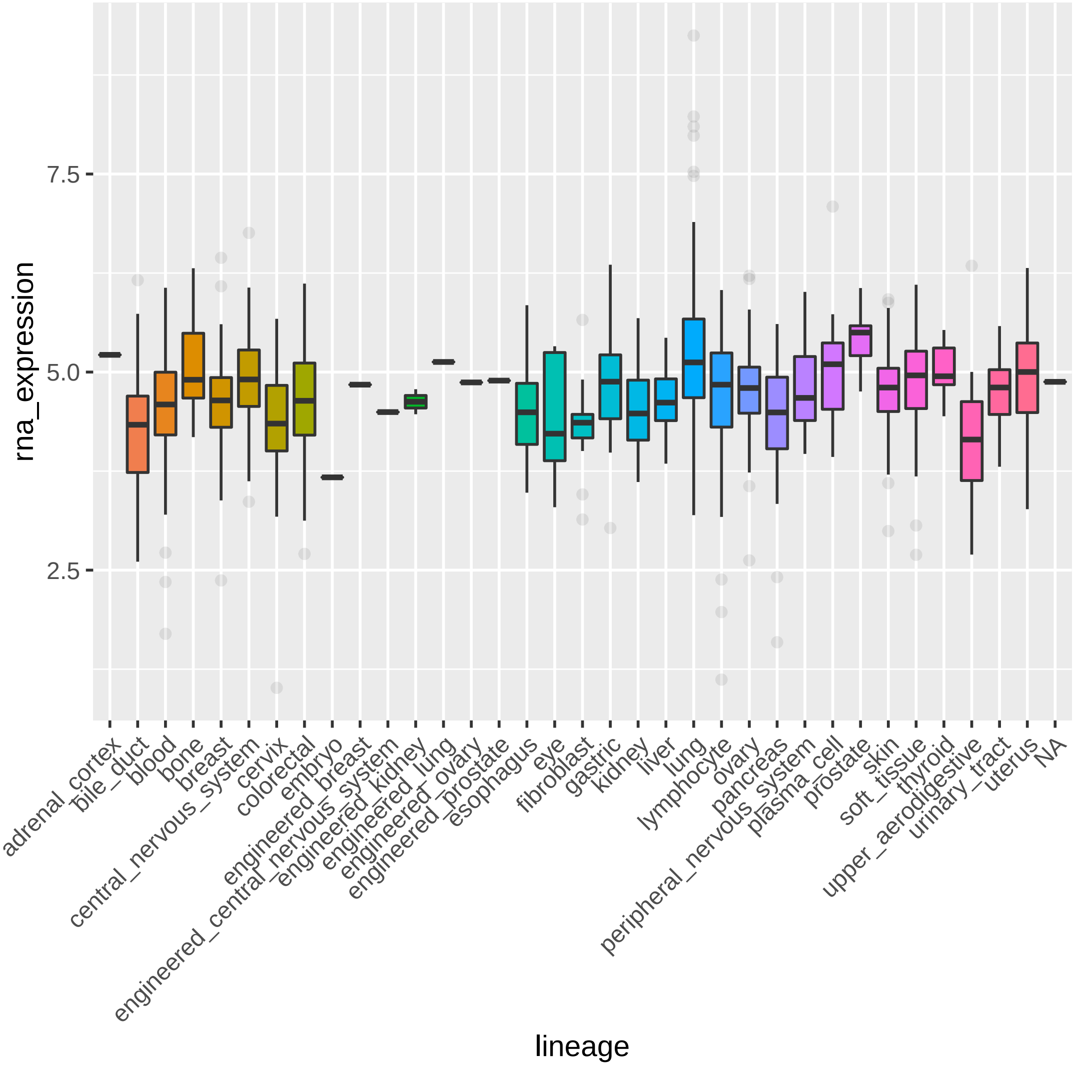

Figure 3. Boxplot of TPM expression values for gene C1orf109 by lineage.

of the `depmap` data affords visualization of genetic expression patterns across many major types of cancer. Elevated expression of gene `C1orf109` in lung cancer tissue has been reported in literature [14]. Figure 3 below shows a boxplot illustrating expression values for gene `C1orf109` by lineage:

```
metadata %>%
  dplyr::select(depmap_id, lineage) %>%
  dplyr::full_join(TPM, by = "depmap_id") %>%
  dplyr::filter(gene_name == "C1orf109") %>%
  ggplot(aes(x = lineage, y = rna_expression, fill = lineage)) +
  geom_boxplot(outlier.alpha = 0.1) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  theme(legend.position = "none")
```

Elevated gene expression and genetic dependency in cancer cell lines have been reported in literature [1, 7]. Therefore, genes with elevated gene expression and high genetic dependency may present especially interesting research targets. Figure 4 shows a plot of expression versus CRISPR gene dependency for Rhabdomyosarcoma. The red vertical line represents the average gene expression for this form of cancer, while the horizontal line represents the average dependency for this cancer type.

```
sarcoma <- metadata %>%
  dplyr::select(depmap_id, cell_line, primary_disease, subtype_disease) %>%
  dplyr::filter(primary_disease == "Sarcoma",
                subtype_disease == "Rhabdomyosarcoma")

crispr_sub <- crispr %>%
  dplyr::select(depmap_id, gene, gene_name, dependency)

tpm_sub <- TPM %>%
  dplyr::select(depmap_id, gene, gene_name, rna_expression)

sarcoma_dep <- sarcoma %>%
  dplyr::left_join(crispr_sub, by = "depmap_id") %>%
  dplyr::select(-cell_line, -primary_disease,
                -subtype_disease, -gene_name)

sarcoma_exp <- sarcoma %>%
  dplyr::left_join(tpm_sub, by = "depmap_id")

sarcoma_dat_exp <- dplyr::full_join(sarcoma_dep, sarcoma_exp,
                                    by = c("depmap_id", "gene")) %>%
                   dplyr::filter(!is.na(rna_expression))
```

```
sarcoma_dat_exp %>%
    ggplot(aes(x = dependency, y = rna_expression)) +
    geom_point(alpha = 0.4, size = 0.5) +
    geom_vline(xintercept = mean(sarcoma_dat_exp$dependency, na.rm = TRUE),
               linetype = "dotted", color = "red") +
    geom_hline(yintercept = mean(sarcoma_dat_exp$rna_expression, na.rm = TRUE),
               linetype = "dotted", color = "red") +
    theme(axis.text.x = element_text(angle = 45))
```

Genes with the highest depenency scores and highest TPM gene expression are found in the upper left section of the plot above 4. Almost all of the genes with the highest depenency scores display increased expression.

```
sarcoma_dat_exp %>%
  dplyr::select(cell_line, gene_name, dependency, rna_expression) %>%
  dplyr::arrange(dependency, rna_expression)
```

```
## # A tibble: 95,720 x 4
##    cell_line            gene_name dependency rna_expression
##    <chr>                <chr>          <dbl>          <dbl>
##  1 JR_SOFT_TISSUE       RAN            -2.49           9.51
##  2 SCMCRM2_SOFT_TISSUE RAN             -2.43           9.89
##  3 SCMCRM2_SOFT_TISSUE SNRPD1          -2.31           7.99
##  4 JR_SOFT_TISSUE       C1orf109       -2.28           4.56
##  5 SCMCRM2_SOFT_TISSUE ATP6V1B2        -2.23           5.44
##  6 SCMCRM2_SOFT_TISSUE POLR2L          -2.21           6.09
##  7 SCMCRM2_SOFT_TISSUE PSMA3           -2.20           7.58
##  8 JR_SOFT_TISSUE       TXNL4A         -2.19           5.53
##  9 SCMCRM2_SOFT_TISSUE POLR2I          -2.19           6.51
## 10 JR_SOFT_TISSUE       SNRPD1         -2.19           8.28
## # ... with 95,710 more rows
```

Changes in genomic copy number may also play a role in some cancer phenotypes [3, 18, 19]. The `depmap` data allows the display of log genomic copy number for across many cancer lineages. Figure 5 shows such a plot for gene `C1orf109` for each major type of cancer lineage:

```
metadata %>%
  dplyr::select(depmap_id, lineage) %>%
  dplyr::full_join(copyNumber, by = "depmap_id") %>%
  dplyr::filter(gene_name == "C1orf109") %>%
  ggplot(aes(x = lineage, y = log_copy_number, fill = lineage)) +
  geom_boxplot(outlier.alpha = 0.1) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  theme(legend.position = "none")
```

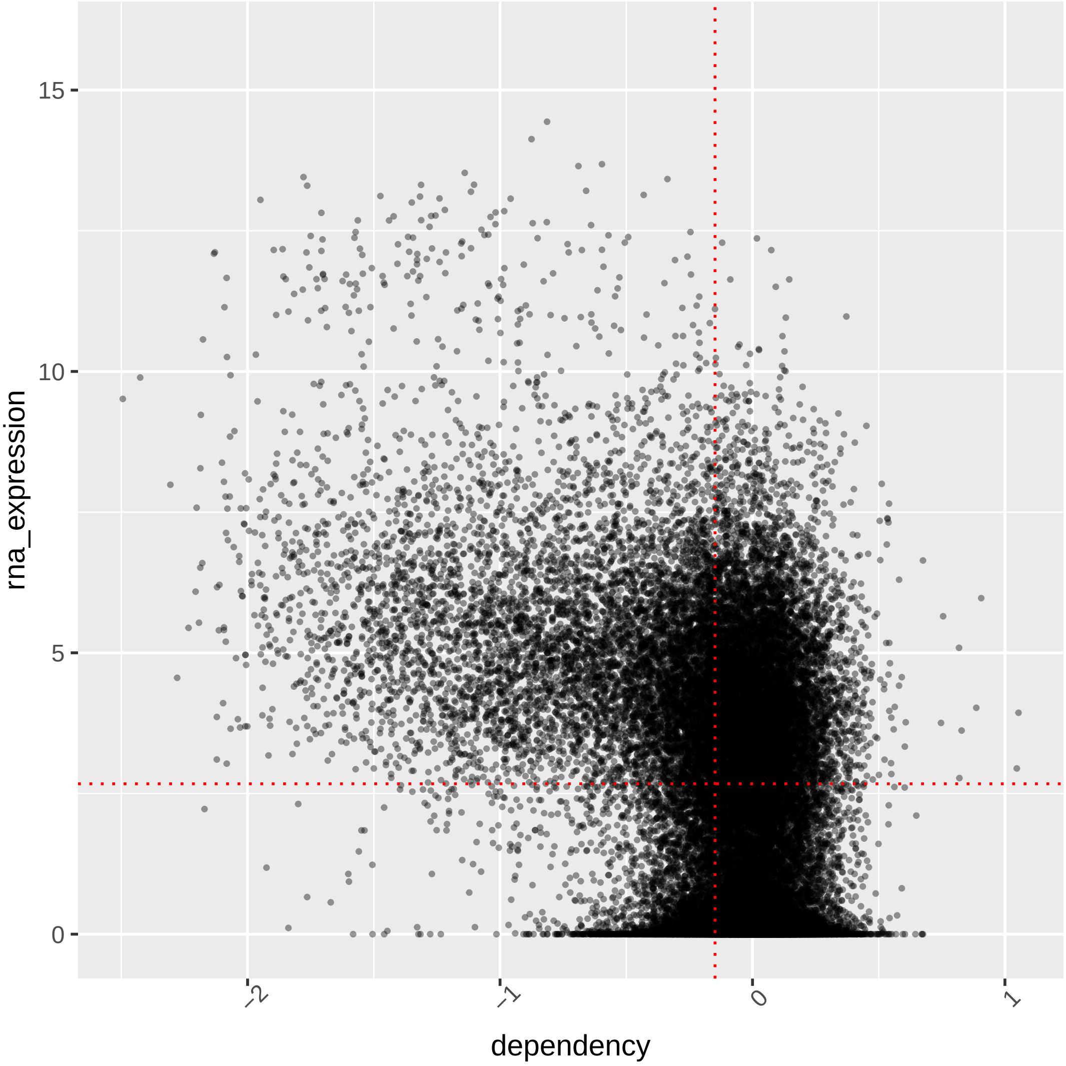

Figure 4. **Expression vs crispr gene dependency for Rhabdomyosarcoma.**

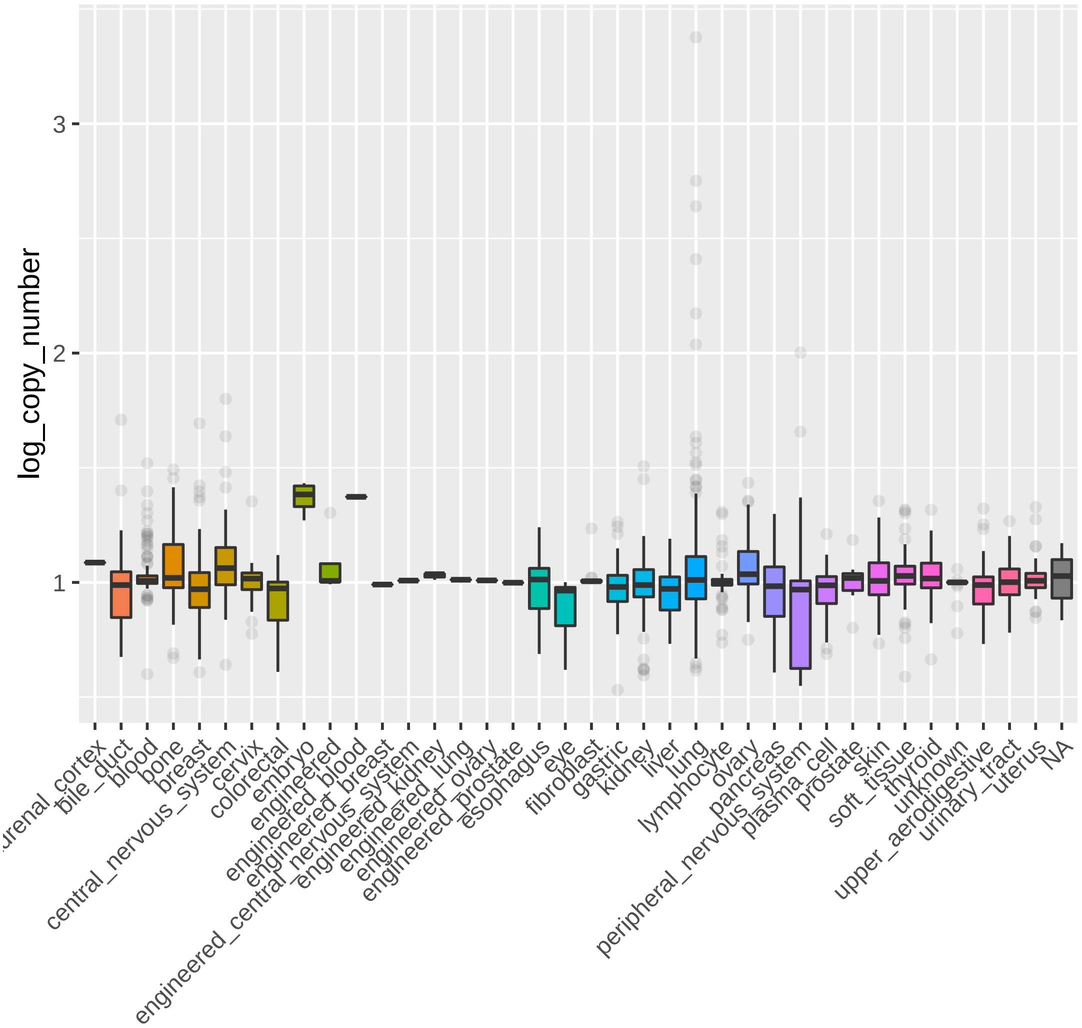

Figure 5. Boxplot of log copy number for gene C1orf109 by lineage

Discussion and outlook

We hope that this package will be used by cancer researchers to dig deeper into the DepMap data and to support their research. Additionally, we highly encourage future `depmap` users to combine `depmap` data with other datasets of interest, such as `TCGA` and `CCLE`.

The depmap R package will continue to be maintained in line with the biannual Bioconductor release, in addition to quarterly releases of DepMap data.

We welcome feedback and questions from the community. We also highly appreciate contributions to the code in the form of pull requests.

Software availability

All packages used in this workflow are available from the Comprehensive R Archive Network (`https://cran.r-project.org`) or Bioconductor (`http://bioconductor.org`). The specific version numbers of R and the packages used are shown below.

Note, the latest `depmap` datasets are only available through `ExperimentHub` by installing and running the latest `devel` version of Bioconductor. To install the `devel` version of Bioconductor:

```
if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
BiocManager::install(version = "devel")
BiocManager::valid()              # checks for out of date packages
```

To install the `depmap` package:

```
BiocManager::install('depmap')
```

```
## R version 4.0.3 Patched (2021-01-18 r79847)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Manjaro Linux
##
## Matrix products: default
## BLAS:   /usr/lib/libblas.so.3.9.0
## LAPACK: /usr/lib/liblapack.so.3.9.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] parallel  stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] stringr_1.4.0        ggplot2_3.3.3        ExperimentHub_1.16.0
##  [4] AnnotationHub_2.22.0 BiocFileCache_1.14.0 dbplyr_2.1.1
##  [7] BiocGenerics_0.36.0  depmap_1.4.0         dplyr_1.0.5
## [10] kableExtra_1.3.4
##
## loaded via a namespace (and not attached):
##  [1] Biobase_2.50.0                httr_1.4.2
##  [3] bit64_4.0.5                   viridisLite_0.3.0
##  [5] shiny_1.6.0                   assertthat_0.2.1
##  [7] interactiveDisplayBase_1.28.0 BiocManager_1.30.12
##  [9] stats4_4.0.3                  blob_1.2.1
## [11] yaml_2.2.1                    BiocWorkflowTools_1.16.0
## [13] BiocVersion_3.12.0            pillar_1.5.1
## [15] RSQLite_2.2.5                 glue_1.4.2
## [17] digest_0.6.27                 promises_1.2.0.1
## [19] rvest_1.0.0                   colorspace_2.0-0
## [21] htmltools_0.5.1.1             httpuv_1.5.5
## [23] pkgconfig_2.0.3               bookdown_0.21.6
## [25] purrr_0.3.4                   xtable_1.8-4
## [27] scales_1.1.1                  webshot_0.5.2
## [29] svglite_2.0.0                 later_1.1.0.1
## [31] git2r_0.28.0                  tibble_3.1.0
## [33] farver_2.1.0                  generics_0.1.0
## [35] IRanges_2.24.1                usethis_2.0.1
## [37] ellipsis_0.3.1                cachem_1.0.4
## [39] withr_2.4.1                   cli_2.4.0
## [41] magrittr_2.0.1                crayon_1.4.1
## [43] mime_0.10                     ps_1.6.0
## [45] memoise_2.0.0                 evaluate_0.14
## [47] fs_1.5.0                      fansi_0.4.2
## [49] xml2_1.3.2                    tools_4.0.3
## [51] lifecycle_1.0.0               S4Vectors_0.28.1
## [53] munsell_0.5.0                 AnnotationDbi_1.52.0
## [55] compiler_4.0.3                systemfonts_1.0.1
## [57] rlang_0.4.10                  grid_4.0.3
## [59] rstudioapi_0.13               rappdirs_0.3.3
## [61] labeling_0.4.2                rmarkdown_2.7
## [63] gtable_0.3.0                  DBI_1.1.1
## [65] curl_4.3                      R6_2.5.0
## [67] knitr_1.31.3                  fastmap_1.1.0
## [69] bit_4.0.4                     utf8_1.2.1
## [71] stringi_1.5.3                 Rcpp_1.0.6
## [73] vctrs_0.3.7                   tidyselect_1.1.0
## [75] xfun_0.22
```

Acknowledgements

Competing interests

No competing interests were disclosed.

Grant information

References

[1] Aviad Tsherniak, Francisca Vazquez, Phil G Montgomery, Barbara A Weir, Gregory Kryukov, Glenn S Cowley, Stanley Gill, William F Harrington, Sasha Pantel, John M Krill-Burger, et al. Defining a cancer dependency map. *Cell*, 170(3): 564–576, 2017.

[2] Depmap Broad. Depmap achilles 20q1 public. *Broad Institute, Cambridge, MA*, 2020.

[3] Robin M Meyers, Jordan G Bryan, James M McFarland, Barbara A Weir, Ann E Sizemore, Han Xu, Neekesh V Dharia, Phillip G Montgomery, Glenn S Cowley, Sasha Pantel, et al. Computational correction of copy number effect improves specificity of crispr–cas9 essentiality screens in cancer cells. *Nature genetics*, 49(12):1779–1784, 2017.

[4] Joshua M Dempster, Jordan Rossen, Mariya Kazachkova, Joshua Pan, Guillaume Kugener, David E Root, and Aviad Tsherniak. Extracting biological insights from the project achilles genome-scale crispr screens in cancer cell lines. *BioRxiv*, page 720243, 2019.

[5] Joshua M Dempster, Clare Pacini, Sasha Pantel, Fiona M Behan, Thomas Green, John Krill-Burger, Charlotte M Beaver, Scott T Younger, Victor Zhivich, Hanna Najgebauer, et al. Agreement between two large pan-cancer crispr-cas9 gene dependency data sets. *Nature Communications*, 10(1):1–14, 2019.

[6] Glenn S Cowley, Barbara A Weir, Francisca Vazquez, Pablo Tamayo, Justine A Scott, Scott Rusin, Alexandra East-Seletsky, Levi D Ali, William FJ Gerath, Sarah E Pantel, et al. Parallel genome-scale loss of function screens in 216 cancer cell lines for the identification of context-specific genetic dependencies. *Scientific data*, 1:140035, 2014.

[7] James M McFarland, Zandra V Ho, Guillaume Kugener, Joshua M Dempster, Phillip G Montgomery, Jordan G Bryan, John M Krill-Burger, Thomas M Green, Francisca Vazquez, Jesse S Boehm, et al. Improved estimation of cancer dependencies from large-scale rnai screens using model-based normalization and data integration. *Nature communications*, 9(1):1–13, 2018.

[8] Steven M Corsello, Rohith T Nagari, Ryan D Spangler, Jordan Rossen, Mustafa Kocak, Jordan G Bryan, Ranad Humeidi, David Peck, Xiaoyun Wu, Andrew A Tang, et al. Non-oncology drugs are a source of previously unappreciated anti-cancer activity. *bioRxiv*, page 730119, 2019.

[9] David P Nusinow, John Szpyt, Mahmoud Ghandi, Christopher M Rose, E Robert McDonald III, Marian Kalocsay, Judit Jané-Valbuena, Ellen Gelfand, Devin K Schweppe, Mark Jedrychowski, et al. Quantitative proteomics of the cancer cell line encyclopedia. *Cell*, 180(2):387–402, 2020.

[10] Kirill Müller and Hadley Wickham. tibble: Simple data frames. r package version 1.3. 3, 2017.

[11] Hadley Wickham and Maintainer Hadley Wickham. Package 'dplyr'. *Retreived from https://cran. rproject. org/web/packages/dplyr/dplyr. pdf*, 2020.

[12] Hadley Wickham. ggplot2. *Wiley Interdisciplinary Reviews: Computational Statistics*, 3(2):180–185, 2011.

[13] Martin Morgan and Lori Shepherd. *ExperimentHub: Client to access ExperimentHub resources*, 2020. R package version 1.14.0.

[14] Shan-shan Liu, Hong-xia Zheng, Hua-dong Jiang, Jie He, Yang Yu, You-peng Qu, Lei Yue, Yao Zhang, and Yu Li. Identification and characterization of a novel gene, c1orf109, encoding a ck2 substrate that is involved in cancer cell proliferation. *Journal of biomedical science*, 19(1):49, 2012.

[15] Xianghua Li, Jasna Lalić, Pablo Baeza-Centurion, Riddhiman Dhar, and Ben Lehner. Changes in gene expression predictably shift and switch genetic interactions. *Nature communications*, 10(1):1–15, 2019.

[16] Enrique Hernández-Lemus, Helena Reyes-Gopar, Jesús Espinal-Enríquez, and Soledad Ochoa. The many faces of gene regulation in cancer: A computational oncogenomics outlook. *Genes*, 10(11):865, 2019.

[17] Sara J Felts, Xiaojia Tang, Benjamin Willett, Virginia P Van Keulen, Michael J Hansen, Krishna R Kalari, and Larry R Pease. Stochastic changes in gene expression promote chaotic dysregulation of homeostasis in clonal breast tumors. *Communications biology*, 2(1):1–7, 2019.

[18] Andrew J Aguirre, Robin M Meyers, Barbara A Weir, Francisca Vazquez, Cheng-Zhong Zhang, Uri Ben-David, April Cook, Gavin Ha, William F Harrington, Mihir B Doshi, et al. Genomic copy number dictates a gene-independent cell response to crispr/cas9 targeting. *Cancer discovery*, 6(8):914–929, 2016.

[19] Xin Shao, Ning Lv, Jie Liao, Jinbo Long, Rui Xue, Ni Ai, Donghang Xu, and Xiaohui Fan. Copy number variation is highly correlated with differential gene expression: a pan-cancer study. *BMC medical genetics*, 20(1):175, 2019.