STAR Protocols

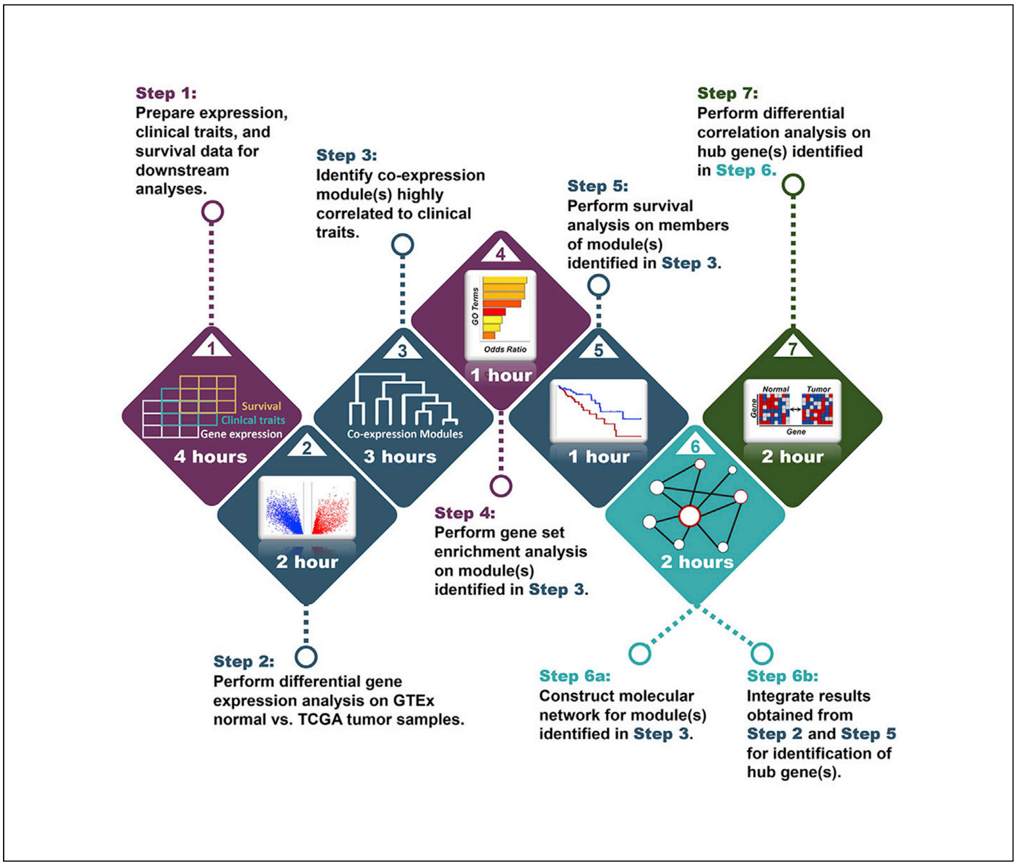
CellPress
OPEN ACCESS

Protocol

Network analysis of TCGA and GTEx gene expression datasets for identification of trait-associated biomarkers in human cancer

Huey-Miin Chen,
Justin A. MacDonald

thmchen@ucalgary.ca
(H.-M.C.)
jmacdo@ucalgary.ca
(J.A.M.)


Step 1:
Prepare expression, clinical traits, and survival data for downstream analyses.
Step 2:
Perform differential gene expression analysis on GTEx normal vs. TCGA tumor samples.
Step 3:
Identify co-expression module(s) highly correlated to clinical traits.
Step 4:
Perform gene set enrichment analysis on module(s) identified in Step 3.
Step 5:
Perform survival analysis on members of module(s) identified in Step 3.
Step 6a:
Construct molecular network for module(s) identified in Step 3.
Step 6b:
Integrate results obtained from Step 2 and Step 5 for identification of hub gene(s).
Step 7:
Perform differential correlation analysis on hub gene(s) identified in Step 6.
1
Survival
Clinical traits
Gene expression
4 hours
2
2 hour
3
Co-expression Modules
3 hours
4
GO Terms
Odds Ratio
1 hour
5
1 hour
6
2 hours
7
Normal
Tumor
Gene
Gene
2 hour

Advances in high-throughput sequencing technologies now yield unprecedented volumes of OMICs data with opportunities to conduct systematic data analyses and derive novel biological insights. Here, we provide protocols to perform differential-expressed gene analysis of TCGA and GTEx RNA-Seq data from human cancers, complete integrative GO and network analyses with focus on clinical and survival data, and identify differential correlation of trait-associated biomarkers.

Highlights

Protocols for the identification of trait-associated molecular correlates in cancer

Differentially-expressed gene (DEG) analysis of TCGA and GTEx transcriptomic data

Protocols for integrative network analysis of RNA-seq, clinical, and survival data

Differential correlation of trait-associated biomarkers for hypothesis testing

Chen & MacDonald, STAR Protocols 3, 101168
March 18, 2022 © 2022 The Author(s).
https://doi.org/10.1016/j.xpro.2022.101168

STAR Protocols

CellPress
OPEN ACCESS

Protocol

Network analysis of TCGA and GTEx gene expression datasets for identification of trait-associated biomarkers in human cancer

Huey-Miin Chen[1,2,3,*] and Justin A. MacDonald[1,*]

[1] Department of Biochemistry & Molecular Biology, Cumming School of Medicine, University of Calgary, Calgary, AB T2N 4Z6, Canada

[2] Technical contact

[3] Lead contact

*Correspondence: thmchen@ucalgary.ca (H.-M.C.), jmacdo@ucalgary.ca (J.A.M.)

https://doi.org/10.1016/j.xpro.2022.101168

SUMMARY

Advances in high-throughput sequencing technologies now yield unprecedented volumes of OMICs data with opportunities to conduct systematic data analyses and derive novel biological insights. Here, we provide protocols to perform differential-expressed gene analysis of TCGA and GTEx RNA-Seq data from human cancers, complete integrative GO and network analyses with focus on clinical and survival data, and identify differential correlation of trait-associated biomarkers. For complete details on the use and execution of this protocol, please refer to Chen and MacDonald (2021).

BEFORE YOU BEGIN

Timing: 0.5–1 h

1. The hardware specifications for the computing platform used to estimate the timing for each step are provided in the key resources table.
2. The R software environment for statistical computing and graphics is required for this protocol. The latest R version (4.1.2), downloaded from https://CRAN.R-project.org/bin/windows/base/, was used to perform the protocol below. This protocol describes the specific steps for network analysis of TCGA colon cancer gene expression. Tissue-specific parameters for the extension of the protocol application to 18 additional human cancers are also contained within this document.
3. The R Studio integrated development environment (IDE), that provides a graphical interface to R, can be downloaded from https://www.rstudio.com/products/rstudio/.
4. R packages utilized in this protocol are listed under the *Software and algorithms* heading of the key resources table. To install the listed R packages, first install BiocManager with commands:

```
> chooseCRANmirror();

> install.packages("BiocManager")
```

5. Then, run the following command to install the R packages listed under the *Software and algorithms* heading of the key resources table.

```
> BiocManager::install("Name_of_Package")
```

6. R is always pointed at a designated directory. Specify a working directory at the start of each R session with the following command:

```
> setwd("Path_to_the_Desired_Folder")
```

7. In addition to R, the Cytoscape (3.9.0) software platform and the stringApp add-on (1.7.0) are required to retrieve molecular networks from the STRING database. Cytoscape 3.9.0 can be downloaded from https://cytoscape.org/download.html. Then, start Cytoscape and go to **App > App Manager** to search for and install stringApp (1.7.0).

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
TcgaTargetGtex_gene_expected_count	Xena Toil data hub	https://toil-xena-hub.s3.us-east-1.amazonaws.com/download/TcgaTargetGtex_gene_expected_count.gz
TcgaTargetGTEX_phenotype	Xena Toil data hub	https://toil-xena-hub.s3.us-east-1.amazonaws.com/download/TcgaTargetGTEX_phenotype.txt.gz
COAD_clinicalMatrix	Xena TCGA data hub	https://tcga-xena-hub.s3.us-east-1.amazonaws.com/download/TCGA.COAD.sampleMap%2FCOAD_clinicalMatrix
TCGA_survival_data	Xena Toil data hub	https://toil-xena-hub.s3.us-east-1.amazonaws.com/download/TCGA_survival_data
Software and algorithms		
Windows OS 10 Home, 64-bit	Microsoft	https://www.microsoft.com/
R (4.1.2)	The R Project	https://CRAN.R-project.org/bin/windows/base/
RStudio (2021.09.1 Build 372)	RStudio Team	https://www.rstudio.com/products/rstudio/
BiocManager (1.30.16)	Morgan (2021)	https://CRAN.R-project.org/package=BiocManager
UCSCXenaTools (1.4.7)	Wang and Liu (2019)	https://CRAN.R-project.org/package=UCSCXenaTools
data.table (1.14.2)	Dowle and Srinivasan (2021)	https://CRAN.R-project.org/package=data.table
R.utils (2.11.0)	Bengtsson (2021)	https://CRAN.R-project.org/package=R.utils
dplyr (1.0.7)	Wickham et al. (2021)	https://CRAN.R-project.org/package=dplyr
limma (3.48.3)	Ritchie et al. (2015)	https://bioinf.wehi.edu.au/limma/
edgeR (3.34.1)	McCarthy et al. (2012) and Robinson et al. (2010)	https://bioinf.wehi.edu.au/edgeR/
topGO (2.44.0)	Alexa and Rahnenführer (2021)	https://bioconductor.org/packages/topGO/
grex (1.9)	Xiao et al. (2019)	https://CRAN.R-project.org/package=grex
biomaRt (2.48.3)	Durinck et al. (2005, 2009)	https://bioconductor.org/packages/biomaRt/
ggplot2 (3.3.5)	Wickham et al. (2016)	https://ggplot2.tidyverse.org/
RegParallel (1.10.0)	Blighe and Lasky-Su (2021)	https://github.com/kevinblighe/RegParallel
survminer (0.4.9)	Kassambara et al. (2021)	https://CRAN.R-project.org/package=survminer
Cytoscape (3.9.0)	Shannon et al. (2003)	https://cytoscape.org/
stringApp (1.7.0)	Doncheva et al. (2019)	https://apps.cytoscape.org/apps/stringapp
DGCA (1.0.2)	McKenzie et al. (2016)	https://CRAN.R-project.org/package=DGCA
org.Hs.eg.db (3.13.0)	Carlson (2021)	https://bioconductor.org/packages/org.Hs.eg.db/
GOstats (2.58.0)	Falcon and Gentleman (2007)	https://bioconductor.org/packages/GOstats/
HGNChelper (0.8.1)	Oh et al. (2020)	https://CRAN.R-project.org/package=HGNChelper
plotrix (3.8-2)	Lemon (2006)	https://CRAN.R-project.org/package=plotrix

(*Continued on next page*)

Continued

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Other		
ID/gene mapping	GENCODE project	https://toil-xena-hub.s3.us-east-1.amazonaws.com/download/probeMap%2Fgencode.v23.annotation.gene.probemap
Genes.xlsx	Piovesan et al. (2019)	https://osf.io/edjzv/
Computing Platform (e.g., Alienware Aurora R12 desktop; 11th Gen Intel® Core™ i7-11700F @ 2.50GHz processor with 32 GB, 2×16GB, 3200 MHz, VMR memory)	Dell Technologies	https://www.dell.com/

Note: The R packages listed under the *Software and algorithms* heading are current as of December 20, 2021.

MATERIALS AND EQUIPMENT

The analyses outlined in this protocol utilize publicly available RNA-Seq, clinical, and survival datasets and several well-established CRAN and BioConductor R packages. All human subjects have been de-identified. The ethical principles set forth by the International Ethical Guidelines for Health-related Research Involving Humans (CIOMS, 2016) should preside over any activities that fall under health-related research with humans, such as biobanking and epidemiological studies. Researchers who study these data should also comply with TCGA and GTEx policies such as maintaining participants' privacy, accessing data securely, and following TCGA and GTEx publication guidelines. To cite R packages, run the following command in R.

```
> citation("Package_Name")
```

STEP-BY-STEP METHOD DETAILS

Data import, cleaning, and preprocessing

Timing: 1–4 h

Note: The Methods S1 file "R Markdown Code Script for UCSCXenaTool" contains the exact script used to generate the expected outcomes for this section.

1. Download `Genes.xlsx` (Piovesan et al., 2019) from https://osf.io/edjzv/ and save the downloaded file as `zz_gene.protein.coding.csv` to your working directory.
2. Download the `ID/gene mapping` for the `TcgaTargetGtex_gene_expected_count` dataset of the `TCGA TARGET GTEx` cohort (see key resources table for link) and save the downloaded file as `zz_gencode.v23.annotation.csv` to your working directory.
3. Load the following R packages: 1) `UCSCXenaTools`, 2) `data.table`, 3) `R.utils`, and 4) `dplyr`.
4. Generate a record of datasets hosted on UCSC Xena Data Hubs via the UCSCXenaTools R package (Wang and Liu, 2019).

   Note: Datasets from UCSC Xena Data Hubs can be filtered by their XenaHostNames, XenaCohorts, and/or XenaDatasets. The object `XenaData` provides information on the contents of each dataset and can be saved to the working directory to assist in the selection of target datasets.

5. Retrieve gene expression, clinical, survival, and phenotype data from the UCSC Xena platform via the UCSCXenaTools R package (Wang and Liu, 2019). For network analysis of TCGA colon cancer gene expression, we selected and downloaded:

a. The `TcgaTargetGtex_gene_expected_count` dataset for the `TCGA TARGET GTEx` cohort from host `toilHub`.
b. The `COAD_clinicalMatrix` dataset for the `TCGA Colon Cancer` cohort from host `tcgaHub` by setting the filters `paraCohort = "TCGA Colon Cancer"` and `paraDatasets = "TCGA.COAD.sampleMap/COAD_clinicalMatrix"`.
c. The `TCGA_survival_data` dataset for the `TCGA TARGET GTEx` cohort from host `toilHub`.
d. The `TcgaTargetGTEX_phenotype` dataset for the `TCGA TARGET GTEx` cohort from host `toilHub`.

Note: Within the `TcgaTargetGtex_gene_expected_count` dataset, GTEx normal tissue data complimentary to TCGA tumor data are available for 16 primary sites (see Table 1). This protocol can be used (with minor modifications) to conduct paired comparison on these 16 primary sites. Table 2 can be used to set values for the arguments `paraCohort` and `paraDatasets` (defined during step 5.b) to retrieve dataset(s) that contain clinical information of the desired cancer type.

6. Subset the gene expression matrix to include only observations of desired tissue type(s).
   a. The Genotype-Tissue Expression (GTEx) project provides gene expression data from healthy, cancer-free individuals. For differential gene expression analysis, we selected GTEx normal colon tissue samples by setting the filters `paraStudy = "GTEX"`, `paraPrimarySiteGTEx = "Colon"` and `paraPrimaryTissueGTEx = "^Colon"` to the `TcgaTargetGTEX_phenotype` dataset. Table 3 can be used to set values for the arguments `paraPrimarySiteGTEx` and `paraPrimaryTissueGTEx`.
   b. The Cancer Genome Atlas (TCGA) program provides gene expression data from primary tumors. For differential gene expression analysis, we selected TCGA colon cancer primary tumor samples by setting the filters `paraSampleType = "Primary Tumor"`, `paraPrimarySiteTCGA ="Colon"`, and `paraHistologicalType = "Colon Adenocarcinoma"`. Table 4 can be used to set values for the arguments `paraPrimarySiteTCGA` and `paraHistologicalType`.
   c. The `TcgaTargetGtex_gene_expected_count` dataset from the `toilHub` data hub combines RNA-Seq data from TCGA and GTEx by uniformly realigning reads to the hg38 genome and re-calling expressions using RSEM and Kallisto methods (Vivian et al., 2017). To compare gene expression between GTEx normal and TCGA tumor for network analyses, subset the

Table 1. List of primary sites where complimentary GTEx normal tissue samples can be found for TCGA tumor samples. Numbers represent count of sample IDs

Primary site	GTEX normal tissue	TCGA primary tumor
Adrenal gland	126	77
Bladder	9	404
Brain	1148	660
Breast	178	1090
Colon	307	282
Esophagus	652	181
Kidney	28	884
Liver	110	369
Lung	288	1011
Ovary	88	419
Pancreas	167	177
Prostate	100	494
Skin	555	102
Stomach	174	410
Testis	165	132
Uterus	78	57

Table 2. List of values that can be used for arguments "`paraCohort`" and "`paraDatasets`" (defined during step 5.b) to retrieve dataset(s) containing the desired cancer type

Primary site	paraCohort	paraDatasets
Adrenal gland	TCGA Adrenocortical Cancer	TCGA.ACC.sampleMap/ACC_clinicalMatrix
Bladder	TCGA Bladder Cancer	TCGA.BLCA.sampleMap/BLCA_clinicalMatrix
Brain	TCGA Glioblastoma	TCGA.GBM.sampleMap/GBM_clinicalMatrix
	TCGA Lower Grade Glioma	TCGA.LGG.sampleMap/LGG_clinicalMatrix
Breast	TCGA Breast Cancer	TCGA.BRCA.sampleMap/BRCA_clinicalMatrix
Colon	TCGA Colon Cancer	TCGA.COAD.sampleMap/COAD_clinicalMatrix
Esophagus	TCGA Esophageal Cancer	TCGA.ESCA.sampleMap/ESCA_clinicalMatrix
Kidney	TCGA Kidney Chromophobe	TCGA.KICH.sampleMap/KICH_clinicalMatrix
	TCGA Kidney Clear Cell Carcinoma	TCGA.KIRC.sampleMap/KIRC_clinicalMatrix
	TCGA Kidney Papillary Cell Carcinoma	TCGA.KIRP.sampleMap/KIRP_clinicalMatrix
Liver	TCGA Liver Cancer	TCGA.LIHC.sampleMap/LIHC_clinicalMatrix
Lung	TCGA Lung Cancer	TCGA.LUNG.sampleMap/LUNG_clinicalMatrix
Ovary	TCGA Ovarian Cancer	TCGA.OV.sampleMap/OV_clinicalMatrix
Pancreas	TCGA Pancreatic Cancer	TCGA.PAAD.sampleMap/PAAD_clinicalMatrix
Prostate	TCGA Prostate Cancer	TCGA.PRAD.sampleMap/PRAD_clinicalMatrix
Skin	TCGA Melanoma	TCGA.SKCM.sampleMap/SKCM_clinicalMatrix
Stomach	TCGA Stomach Cancer	TCGA.STAD.sampleMap/STAD_clinicalMatrix
Testis	TCGA Testicular Cancer	TCGA.TGCT.sampleMap/TGCT_clinicalMatrix
Uterus	TCGA Uterine Carcinosarcoma	TCGA.UCS.sampleMap/UCS_clinicalMatrix

`TcgaTargetGtex_gene_expected_count` dataset via lists generated during step 6.a and step 6.b.

Note: When generating the GTEx and TCGA sample lists to subset the gene expression matrix `TcgaTargetGtex_gene_expected_count`, the **`subset`** function may not compute, or may return zero observations. Please refer to troubleshooting, problem 1 (step 6) for potential solution.

7. Subset the gene expression matrix to include only protein-coding genes. This can be achieved by utilizing the `zz_gene.protein.coding.csv` saved during **step 1.**
8. The gene expression matrix can now be saved for downstream analyses.
9. The `COAD_clinicalMatrix` dataset contains 133 administrative and phenotypic annotations (see the Methods S1 file "R Markdown Code Script for UCSCXenaTool" for a full list), ranging from `sample IDs` to `pathologic stage` to `KRAS mutation codon`. Keep only the variable(s) of interest. For example, to identify potential biomarkers for lymphatic invasion during network analysis of TCGA colon cancer gene expression, we retained the following variables:

```
> varClinKeep = c("sampleID", "lymphaticinvasion")
```

Note: Phenotype variables included in clinical matrices differ among the 19 cancer types laid out in Table 2. For examples of how one can view, select, and re-code other phenotype variable(s) into an annotation matrix that is appropriate for downstream analyses, please refer to troubleshooting, problem 2 (step 9).

10. The phenotype annotation matrix can now be saved for downstream analyses.

⚠ CRITICAL: Tissue samples from TCGA are classified by cancer types, as well as by sample types (e.g., primary tumor, solid tissue normal). Solid tissue normal samples (referred to as NAT, normal adjacent to tumor) are collected from histologically normal tissues adjacent

Table 3. List of values that can be used for arguments "`paraPrimarySiteGTEx`" and "`paraPrimaryTissueGTEx`" (defined during step 6.a) to retrieve IDs for GTEx normal samples of desired tissue type(s)

paraPrimarySiteGTEx	paraPrimaryTissueGTEx	Sample size
Adrenal Gland	Adrenal Gland	126
Bladder	Bladder	9
Brain	Brain - Amygdala	69
	Brain - Anterior Cingulate Cortex \\(Ba24\\)	83
	Brain - Caudate \\(Basal Ganglia\\)	108
	Brain - Cerebellar Hemisphere	97
	Brain - Cerebellum	117
	Brain - Cortex	105
	Brain - Frontal Cortex \\(Ba9\\)	101
	Brain - Hippocampus	84
	Brain - Hypothalamus	82
	Brain - Nucleus Accumbens \\(Basal Ganglia\\)	104
	Brain - Putamen \\(Basal Ganglia\\)	81
	Brain - Spinal Cord \\(Cervical C-1\\)	60
	Brain - Substantia Nigra	57
Breast	Breast - Mammary Tissue	178
Colon	Colon - Sigmoid	141
	Colon - Transverse	166
Esophagus	Esophagus - Gastroesophageal Junction	136
	Esophagus - Mucosa	271
	Esophagus - Muscularis	245
Kidney	Kidney - Cortex	28
Liver	Liver	110
Lung	Lung	288
Ovary	Ovary	88
Pancreas	Pancreas	167
Prostate	Prostate	100
Skin	Skin - Not Sun Exposed \\(Suprapubic\\)	232
	Skin - Sun Exposed \\(Lower Leg\\)	323
Stomach	Stomach	174
Testis	Testis	165
Uterus	Uterus	78

to tumor margins and are often utilized as healthy controls for comparison with tumor samples. However, analyses of expression profiles from GTEx healthy, TCGA NAT, and TCGA tumor tissues indicate that NAT constitutes an intermediate state between healthy and tumor (Aran et al., 2017). That is, the expression profiles obtained from healthy, NAT, and tumor tissues segregate into distinct clusters, with closer resemblance of NAT to tumor in some tissues (e.g., prostate and colon) and greater similarity of NAT to healthy in other tissues (e.g., uterus and breast). Over half of the differentially expressed genes (DEGs) in the healthy to tumor comparison were not identified in the NAT to tumor comparison, while ~40% of DEGs in the NAT to tumor comparison were insignificant in the healthy to tumor comparison (Aran et al., 2017). Given the middling nature of solid tissue normal (i.e., NAT), these samples should be removed as they may distort subsequent analyses. Furthermore, solid tissue normal samples often have matched primary tumor samples. As such, inclusion of these solid tissue normal samples will introduce duplicate patient IDs.

⚠ CRITICAL: Tissue samples from TCGA are also classified by histological type. Histological heterogeneity can confound gene module-trait correlation if specific clinical traits are overrepresented in certain histological type. For example, colon mucinous adenocarcinoma has

Table 4. List of values that can be used for arguments "`paraPrimarySiteTCGA`" and "`paraHistologicalType`" (defined during step 6.b) to retrieve IDs for TCGA primary tumor samples of desired histological type(s)

paraPrimarySiteTCGA	paraDatasets	paraHistologicalType	Sample size
Adrenal gland	TCGA.ACC.sampleMap/ACC_clinicalMatrix	Adrenocortical Carcinoma- Myxoid Type	1
		Adrenocortical Carcinoma- Oncocytic Type	3
		Adrenocortical carcinoma- Usual Type	73
Bladder	TCGA.BLCA.sampleMap/BLCA_clinicalMatrix	Muscle invasive urothelial carcinoma	404
Brain	TCGA.GBM.sampleMap/GBM_clinicalMatrix	Glioblastoma Multiforme	1
		Treated primary GBM	1
		Untreated primary \\(de novo\\) GBM	150
	TCGA.LGG.sampleMap/LGG_clinicalMatrix	Astrocytoma	193
		Oligoastrocytoma	126
		Oligodendroglioma	189
Breast	TCGA.BRCA.sampleMap/BRCA_clinicalMatrix	Infiltrating Carcinoma NOS	1
		Infiltrating Ductal Carcinoma	780
		Infiltrating Lobular Carcinoma	203
		Medullary Carcinoma	6
		Metaplastic Carcinoma	9
		Mixed Histology	29
		Mucinous Carcinoma	17
		Other	45
Colon	TCGA.COAD.sampleMap/COAD_clinicalMatrix	Colon Adenocarcinoma	244
		Colon Mucinous Adenocarcinoma	38
Esophagus	TCGA.ESCA.sampleMap/ESCA_clinicalMatrix	Esophagus Adenocarcinoma, NOS	89
		Esophagus Squamous Cell Carcinoma	92
Kidney	TCGA.KICH.sampleMap/KICH_clinicalMatrix	Kidney Chromophobe	66
	TCGA.KIRC.sampleMap/KIRC_clinicalMatrix	Kidney Clear Cell Renal Carcinoma	530
	TCGA.KIRP.sampleMap/KIRP_clinicalMatrix	Kidney Papillary Renal Cell Carcinoma	288
Liver	TCGA.LIHC.sampleMap/LIHC_clinicalMatrix	Fibrolamellar Carcinoma	3
		Hepatocellular Carcinoma	359
		Hepatocholangiocarcinoma \\(Mixed\\)	7
Lung	TCGA.LUNG.sampleMap/LUNG_clinicalMatrix	Lung Acinar Adenocarcinoma	18
		Lung Adenocarcinoma Mixed Subtype	105
		Lung Adenocarcinoma- Not Otherwise Specified	320
		Lung Basaloid Squamous Cell Carcinoma	14
		Lung Bronchioloalveolar Carcinoma Mucinous	5
		Lung Bronchioloalveolar Carcinoma Nonmucinous	19
		Lung Clear Cell Adenocarcinoma	2
		Lung Micropapillary Adenocarcinoma	3
		Lung Mucinous Adenocarcinoma	2
		Lung Papillary Adenocarcinoma	23
		Lung Papillary Squamous Cell Carcinoma	6
		Lung Signet Ring Adenocarcinoma	1
		Lung Small Cell Squamous Cell Carcinoma	1
		Lung Solid Pattern Predominant Adenocarcinoma	5
		Lung Squamous Cell Carcinoma- Not Otherwise Specified	477
		Mucinous \\(Colloid\\) Carcinoma	10
Ovary	TCGA.OV.sampleMap/OV_clinicalMatrix	Serous Cystadenocarcinoma	419
Pancreas	TCGA.PAAD.sampleMap/PAAD_clinicalMatrix	Pancreas-Adenocarcinoma Ductal Type	147
		Pancreas-Adenocarcinoma-Other Subtype	25
		Pancreas-Colloid \\(mucinous non-cystic\\) Carcinoma	4
		Pancreas-Undifferentiated Carcinoma	1
Prostate	TCGA.PRAD.sampleMap/PRAD_clinicalMatrix	Prostate Adenocarcinoma Acinar Type	479
		Prostate Adenocarcinoma, Other Subtype	15

(*Continued on next page*)

Table 4. *Continued*

paraPrimarySiteTCGA	paraDatasets	paraHistologicalType	Sample size
Skin	TCGA.SKCM.sampleMap/SKCM_clinicalMatrix	Not Available	102
Stomach	TCGA.STAD.sampleMap/STAD_clinicalMatrix	Stomach Adenocarcinoma, Signet Ring Type	12
		Stomach, Adenocarcinoma, Diffuse Type	68
		Stomach, Adenocarcinoma, Not Otherwise Specified	155
		Stomach, Intestinal Adenocarcinoma, Mucinous Type	19
		Intestinal Adenocarcinoma, Not Otherwise Specified*	73
		Stomach, Intestinal Adenocarcinoma, Papillary Type	7
		Stomach, Intestinal Adenocarcinoma, Tubular Type	76
Testis	TCGA.TGCT.sampleMap/TGCT_clinicalMatrix	^Non-Seminoma; Choriocarcinoma	1
		^Non-Seminoma; Embryonal Carcinoma	32
		^Non-Seminoma; Teratoma \\(Immature\\)	5
		^Non-Seminoma; Teratoma \\(Mature\\)	16
		^Non-Seminoma; Yolk Sac Tumor	8
		^Seminoma; NOS	70
Uterus	TCGA.UCS.sampleMap/UCS_clinicalMatrix	Uterine Carcinosarcoma/ Malignant Mixed Mullerian Tumor	24
		Uterine Carcinosarcoma/ MMMT: Heterologous Type	20
		Uterine Carcinosarcoma/MMMT: Homologous Type	13

distinct molecular aberrations (e.g., overexpression of the MUC2 protein) and higher ratio of lymphatic invasion as compared to its non-mucinous counterpart (Luo et al., 2019). Out of the 282 primary tumor samples from the `TCGA Colon Cancer` cohort, 38 samples (13%) were annotated with histological type `colon mucinous adenocarcinoma`. Pursuing weighted gene co-expression network analysis (WGCNA) without first resolving histological type may lead to detection of gene-module trait correlations that are associated with histological type in addition to the clinical trait of interest. Table 4 can be used to set values for `paraHistologicalType`, defined during step 6.b.

Note: For complete details on the use and execution of the UCSCXenaTools R package, please refer to (Wang and Liu, 2019).

Differential gene expression analysis with limma-voom

Timing: 0.5–2 h

Differential gene expression analysis is routinely used to investigate the biological differences between healthy and diseased states (McDermaid et al., 2019). Identification of DEGs can be valuable for uncovering potential biomarkers, therapeutic targets, and gene signatures for diagnostics. In this section, we utilize the limma workflow (Law et al., 2016) to detect DEGs across TCGA primary tumor and GTEx normal colon tissue samples.

Note: The Methods S2 file "R Markdown Code Script for LIMMA_ColonCancer" contains the exact script used to generate the expected outcomes for this section.

11. Load the following R packages: 1) `dplyr`, 2) `limma`, and 3) `edgeR`.
12. The `TcgaTargetGtex_gene_expected_count` dataset for the `TCGA TARGET GTEx` cohort from host `toilHub` was previously $\log_2(x+1)$ transformed. As such, it is necessary to back-transform the gene expression matrix that was saved to the working directory during **step 8** into RSEM gene-level expected count before passing the dataset to limma.
13. Convert the back-transformed gene expression matrix into a DGEList-object using the **`DGEList`** function.

Note: In *limma-voom*, all samples are assumed to have a similar range and distribution of log-CPM values (Law et al., 2016). Samples that have significantly different range and/or distribution of log-CPM values should be removed prior to the generation of the DGEList-object. To resolve the potential issue of sample outliers, please refer to troubleshooting, problem 3 (**step 13**). It should also be noted that if there are any sample outliers, and these outliers were not removed prior to the generation of the DGEList-object, then warning messages will start appearing when undertaking **step 17**, up until **step 22**, when the **`eBayes`** function will fail on execution.

14. Group samples by condition (i.e., TCGA tumor or GTEx normal).
15. Convert expected counts to counts per million (CPM) and $\log_2$-counts per million using the function **`cpm`**.
16. Remove genes that are lowly expressed using the function **`filterByExpr`**.
17. Perform normalization on gene expression using the function **`calcNormFactors`**.
18. Generate a design matrix using the function **`model.matrix`**.
19. Set up contrast for comparison using the function **`makeContrasts`**.
20. Transform gene expression data for linear modeling using the function **`voom`**.
21. Perform linear modeling using the function **`lmFit`**, then the function **`contrasts.fit`**.
22. Compute empirical Bayes statistics for differential expression using the function **`eBayes`**.
23. Inspect the number of significantly up- and down-regulated genes.
24. Save the list of DEGs for subsequent analyses.
25. Save the *voom* transformed gene expression matrix for subsequent analyses.

Note: For complete details on the use and execution of the limma R package, please refer to Ritchie et al. (2015) and Law et al. (2016).

Identification of gene set(s) highly correlated to specific traits of human cancer with WGCNA

Timing: 1–3 h

In addition to the identification of DEGs, the analysis of correlated gene expression (i.e., co-expression) can provide a framework for describing changes in expression of gene sets within the confine of human cancer traits. WGCNA is a bioinformatic algorithm, developed by Langfelder and Horvath (2008), that can be used to find clusters (modules) of correlated genes and to associate the identified modules with specific sample traits. Importantly, the gene expression data input for WGCNA is not pre-filtered by differential expression. As such, gene sets that may be highly correlated to specific traits of human cancer, but do not pass the differential expression threshold, may still be revealed via WGCNA. In this section, the *voom* normalized expression data, generated from the *limma-voom* workflow presented in the previous section, is subjected to WGCNA to identify gene set(s) highly correlated to lymphatic invasion in human colon cancer.

Note: The Methods S3 file "R Markdown Code Script for WGCNA" contains the exact script used to generate the expected outcomes for this section.

26. Load the following R packages: 1) `WGCNA`, and 2) `dplyr`.
27. Subset the *voom* transformed gene expression matrix, saved during **step 25**, to include only TCGA gene expression data.
28. Remove genes that are lowly-expressed and/or genes with low variation between samples.

Note: Genes that are lowly-expressed and/or display low variation between samples have a tendency to generate noise in WGCNA (Langfelder and Horvath, 2008). Lowly-expressed genes can be removed by defining a mean expression cutoff value. Filtering for low variance can be achieved by setting a variance cutoff value. The method and threshold-value chosen for gene-filtering may vary according to study-specific characteristics. Users should examine expression values across the entire dataset for features that would distinguish uninformative variables. For WGCNA on TCGA gene expression data, we removed genes that had normalized expression value < 0 in any of the 244 colon tissue samples.

29. Identify outlier samples by clustering samples using the function **`hclust`**. Then, remove the outlier samples by setting a value for the `cutHeight` argument in the function **`cutreeStatic`**.

Note: Users should refer to the sample clustering dendrogram generated via the function **`plot`** to discern the most reasonable cut height for their dataset. In general, each dataset will require a different cut height. A histogram of height (the distance between samples and/or clusters) can also assist in the setting of a value for the `cutHeight` argument.

30. Convert the phenotype annotation matrix, saved during **step 10**, into a trait data frame that is analogous to the gene expression data frame.

Note: Sample traits data can be plotted alongside the sample clustering dendrogram, generated during **step 29**, to investigate whether samples with distinct trait values have globally distinct gene expression patterns. If so, then there is likely a high correlation among large groups of genes, which invalidates the scale-free topology assumption and can interfere with the appropriate selection of the soft-thresholding power β (**step 31**). If the lack of scale-free topology is caused by the sample trait of interest, then the recommended power β is six for unsigned or signed hybrid networks, and 12 for signed networks (for sample size listed in Table 1). If the lack of scale-free topology is caused by traits that are uninteresting for the study, then the WGCNA consensus network analysis (not described in this protocol) should be employed. WGCNA consensus network analysis allows for detection of common co-expression patterns across multiple conditions (e.g., the driver of sample differences that is not being studied). Tutorials for the identification of consensus modules that cluster genes with dense connectivity in multiple conditions can be found at https://horvath.genetics.ucla.edu/html/CoexpressionNetwork/Rpackages/WGCNA/Tutorials/, under the heading *II. Consensus analysis of female and male liver expression data*. (Langfelder and Horvath, 2008)

31. Use the function **`pickSoftThreshold`** to analyze scale-free topology for a set of candidate β-values.
32. Plot, and then inspect the results returned by the function **`pickSoftThreshold`**. Select the smallest β-value that satisfies the scale-free topology assumption (i.e., $R^2 > 0.8$).

Note: A reasonable β-value is < 15 for unsigned or signed hybrid networks, and < 30 for signed networks. If the scale-free topology assumption cannot be satisfied with a reasonable β-value, then the input data should be investigated for heterogeneity drivers that instigated the globally distinct expression pattern (see **Note** under **step 30**).

33. Calculate adjacencies with the selected β-value via function **`adjacency`**. The adjacency matrix is a square, symmetrical matrix with values ranging between 0 and 1 that correspond to the connection strengths between each pair of genes.
34. Calculate the topological overlap matrix (TOM) from the adjacency matrix using the function **`TOMsimilarity`**. Then calculate the dissimilarity matrix via command `dissTOM = 1 – TOM`.

Note: When executing functions **`pickSoftThreshold`**, **`adjacency`** and **`TOMsimilarity`**, the network type needs to be defined via arguments *networkType*, *type*, and *networkType*, respectively. In "signed" networks, a pair of genes with positive correlation is considered connected whereas a pair of genes with negative correlation is considered unconnected. In "unsigned" networks, positive and negative correlations are both regarded as connected (i.e., mixed). Although the default network type in WGCNA is "unsigned", the use of "signed" networks is recommended for analysis of gene expression data (see https://peterlangfelder.com/2018/11/25/signed-or-unsigned-which-network-type-is-preferable/).

35. Detect modules within the network by clustering genes based on `dissTOM` using the function **`hclust`**. Module assignments can be plotted under the gene dendrogram using the function **`plotDendroAndColors`**.

    Note: WGCNA may return modules that capture a large fraction of the input data (i.e., containing 10, 20, or 50% of the total input data), which may not provide adequate resolution for the identification of core mechanisms associated with the sample trait of interest. To resolve the potential issue of large modules, please refer to troubleshooting, problem 4 (step 35).

36. Calculate the weighted average value (eigengene) for each module using the function **`moduleEigengenes`**.
37. Calculate the correlation between trait and module eigengenes (MEs) using the function **`cor`** and **`corPvalueStudent`** to uncover module(s) with significant association to the sample trait of interest. A graphical view of module-trait relationship can be generated with the function **`labeledHeatmap`**.
38. Calculate the correlation between trait and gene expression levels with the function **`cor`** and **`corPvalueStudent`** to define gene significance (GS).
39. Calculate the correlation between MEs and gene expression levels using the function **`cor`** and **`corPvalueStudent`** to define module membership (MM).
40. Plot a scatterplot of variables GS vs. MM to examine if genes that are highly associated with the trait of interest are also highly associated with their assigned module.
41. Annotate results from WGCNA with Ensembl IDs. This can be done by utilizing the `zz_gencode.v23.annotation.csv` downloaded and saved during **step 2** of this protocol.
42. Save the annotated WGCNA result for subsequent analyses.

    Note: For complete details on the use and execution of the WGCNA R package, please refer to Langfelder and Horvath (2008).

Gene set enrichment analysis with topGO

Timing: 0.5–1 h

Having identified clusters (modules) of genes that are highly correlated to specific traits of human cancer, the next step is to infer underlying molecular mechanisms based on the biological attributes of these genes. Gene ontology (GO) is a set of structured and controlled vocabulary, which describes gene characteristics in terms of their function and localization. In the hierarchical tree structure of GO, each child node is a more specific term than its parents. At the highest level, GO terms are classified into three major categories: cellular components (where the gene product is localized), molecular functions (function of the gene product), and biological processes (the activity with which the gene product is involved). In this section, GO enrichment analysis is completed with the topGO R package (Alexa and Rahnenführer, 2021) for the category "biological processes". topGO was chosen to owe to its '*elim*' method, that considers GO hierarchy when calculating enrichment. In brief, the algorithm accounts for the 'inheritance problem' (i.e., where root terms inherit annotations from descendent terms; a situation which can generate false positives) in GO enrichment analysis by disregarding genes that had already been annotated with significantly enriched descendant GO terms (Alexa et al., 2006).

Note: The Methods S4 file "R Markdown Code Script for topGO" contains the exact script used to generate the expected outcomes for this section.

43. Load the following R packages: 1) `data.table`, 2) `grex`, 3) `biomaRt`, 4) `topGO`, 5) `dplyr`, and 6) `ggplot2`.
44. Annotate the WGCNA result, saved during **step 42**, with Entrez IDs, via function **`grex`**.
45. Connect to the `ENSEMBL_MART_ENSEMBL` BioMart database to query GO IDs for the list of Entrez IDs returned by the function **`grex`**. This constitutes the 'gene universe' that we will compare our list of genes of interest to.
46. Define the WGCNA module of interest then set up named factors for genes located within and outside of the module of interest.
47. Build a topGO data object with the base function **`new`**, then run GO analysis.
48. Test significance of GO terms using the function **`runTest`**.
49. Generate a summary table of results obtained from topGO enrichment analysis with the function **`GenTable`**.

Note: If users decide to re-run GO term enrichment analysis after some time (e.g., six months), the outcome of this later analysis may not always agree with those obtained from analysis conducted at an earlier time. For explanation and potential solution, please refer to troubleshooting, problem 5 (step 49, step 61, and step 71).

50. Calculate odds ratios via the command:

```
> all_res$OR = log2((all_res$Significant/tot_candidate)/(all_res$Annotated/
tot_background))
```

51. Generate a summary figure of topGO results via the ggplot2 R package (Wickham et al., 2016).

Note: For complete details on the use and execution of the topGO R package, please refer to Alexa and Rahnenführer (2021) and Alexa et al. (2006).

Survival analysis with RegParallel

Timing: 0.5–1 h

Potential biomarkers for specific human cancer traits can be detected by cross-referencing genes that reside within the WGCNA module(s) of interest with the list of DEGs derived with *limma-voom*. To further restrict this list of potential biomarkers, the prognostic ability of gene expression on survival may be evaluated. In this section, the RegParallel R package (Blighe and Lasky-Su, 2021) was utilized to examine how the expression of genes within the WGCNA module of interest influenced the rate of overall survival.

Note: The Methods S5 file "R Markdown Code Script for Survival" contains the exact script used to generate the expected outcomes for this section.

52. Load the following R packages: 1) `dplyr`, 2) `data.table`, 3) `RegParallel`, and 4) `survminer`.
53. Subset the *voom* transformed gene expression matrix, saved during **step 25**, to include only TCGA sample IDs and only genes within the module of interest.
54. Transform the gene expression data to Z-score with the function **`scale`**.

55. Merge the transformed gene expression matrix with the file TCGA_survival_data that was downloaded and saved during **step 5.c.**

Note: For analysis of how gene expression within the WGCNA module of interest influences the rate of overall survival for members of the TCGA colon cancer cohort, we subset the TCGA_survival_data to include only data ≤3,650 days (i.e., 10 years). The subset criterion of 10-year was set based on data that were available within the TCGA_survival_data dataset for the TCGA colon cancer cohort. A histogram summarizing the range and distribution of available survival data can be generated to assist in the setting of the subset criterion. Five-year survival is commonly used, but long-term survival outcomes (e.g., 10, 15, or 20 years) could also be interrogated when data are available (Miller et al., 2019; Myers and Ries, 1989).

56. Run the function **RegParallel** to fit Cox proportional hazards regression model to gene expression to independently test the association between survival time and each gene within the WGCNA module of interest.

Note: For network analysis of TCGA colon cancer gene expression, we examined how gene expression influences the rate of overall survival. However, other events of interest can be found within the file TCGA_survival_data (e.g., disease-specific survival, disease-free interval, and progression-free interval).

57. Load the summary of *limma-voom* differential expression analysis, saved during **step 24.**
58. Merge the *limma-voom* differential expression data with *RegParallel* output, then save the composite data table for Cytoscape network visualization.
59. Generate a short-list of genes that satisfy the conditions:
    a. Is associated with the trait of interest (see **step 53**).
    b. Is differentially expressed (adj.P.Val < 0.05).
    c. Statistically significant prognostic separation between high and low gene expression (LogRank < 0.05).
60. Generate Kaplan-Meier (KM) plots for shortlisted genes using the function **ggsurvplot**.

Note: For complete details on the use and execution of the RegParallel and survminer R package, please refer to Blighe and Lasky-Su (2021) and Kassambara et al. (2021), respectively.

Molecular network visualization with Cytoscape

Timing: 1–2 h

The effective visual display of data architecture is critical for information dissemination and broader contextual understanding. This section describes the use of stringApp for Cytoscape (Doncheva et al., 2019; Shannon et al., 2003) to retrieve molecular networks from the search tool for the retrieval of interacting genes/proteins (STRING) database (Szklarczyk et al., 2021) given a list of proteins of interest. Moreover, the means to import external data (e.g., results from differential expression and survival analysis) for mapping additional information onto the STRING network is provided.

61. Construct the protein-protein interaction (PPI) network with the STRING database.
    a. Start Cytoscape and go to File > Import > Network from Public Database.
    b. Select STRING: protein query as Data Source.
    c. Copy/paste the *variable* column of data table saved during step 58 into the box titled: Enter protein names or identifiers.
    d. Press Import.

Note: The default *confidence (score) cutoff* value is 0.40. Lowering the cutoff will increase the sensitivity but will also increase the likelihood of false positives. The setting of the cutoff value is arbitrary and is often based on the number of interactions required for the analysis. For example, to visualize PPIs for genes within our WGCNA module of interest, the cutoff was set to 0.15. At this cutoff-value, a single molecular network incorporating 145 out of 151 genes in our module of interest was constructed via the STRING database.

Note: If users decide to re-construct their PPI network after some time (e.g., six months), the newly constructed network may not be the same as what was returned via earlier import of network from the STRING database. For explanation and potential solution, please refer to troubleshooting, problem 5 (**step 49**, **step 61**, and **step 71**).

62. Import the composite data table, saved during **step 58**, to the constructed STRING network to visualize differential expression and survival analysis overlap.
    a. Go to `File > Import > Table from File`.
    b. Select the composite data table, saved during **step 58.** Then press **`Open`**.
    c. Select **`To a Network Collection`** for field **`Where to Import Table Data:`**.
    d. Select **`Node Table Columns`** for field **`Import Data as:`**.
    e. Select **`shared name`** for field **`Key Column for Network`**.
    f. Under the **`Preview`** window, click on the **`Variable`** column then change the **`Meaning:`** to **`Key`**. Then press **`OK`**.
63. Compute a set of topological parameters for the STRING network by selecting **`Tool > Analyze Network`**. Then press **`OK`**.
64. Adjust *Layout* and *Styling* as needed. For our network analysis of TCGA colon cancer gene expression, we emphasized the degree of interaction by mapping node size to the network parameter *degree.* We also highlighted genes that were differentially expressed via colored nodes, and genes that were associated with overall survival with colored node border.
65. Export topology data from Cytoscape by selecting **`File > Export > Table to File`**.... Then change the **`Select a table to export:`** to **`String Network default node.`** Then press **`OK`**.

Note: The exported table can be used to prioritize hub gene(s) by cross-referencing DEGs that significantly impact overall survival probability with their *node degree.* Node degree represents the number of interactions linked to a given gene.

Note: For complete details on the use and execution of the Cytoscape (3.9.0) software platform, please refer to Cytoscape User Manual (http://manual.cytoscape.org/en/stable/index.html) and Shannon et al. (2003).

Differential (co-expression) correlation analysis with DGCA

Timing: 1–2 h

Over the past decade, increased efficiencies in global gene expression profiling have made common the study of differential gene expression. However, this approach overlooks the fact that biological processes are defined by complex interactions among molecules. Thus, methods such as WGCNA (Langfelder and Horvath, 2008) emerged to facilitate the exploration of relationships between gene sets and sample traits. Complementary to differential expression and co-expression studies, differential co-expression (correlation) studies analyze the rewiring of molecular interactions associated with perturbations such as disease or oncogene activation (Savino et al., 2020). In this section, we perform differential correlation analysis with the R package DGCA (McKenzie et al., 2016) on our hub gene (identified during **step 65**) within the gene universe, and between normal and primary tumors. The identification of differentially-correlated genes (DCGs) and the comparison of enrichment of GO terms derived from DCGs can provide insight into the functional relevance of a hub gene in different cellular contexts (e.g., normal vs. tumor). These linkages can further promote the development of an effective hypothesis for integration with experimental validation.

> *Note:* The Methods S6 file "R Markdown Code Script for DCGA" contains the exact script used to generate the expected outcomes for this section.

66. Load the following R packages: 1) `dplyr`, and 2) `DGCA`, 3) `org.Hs.eg.db`, 4) `GOstats`, 5) `HGNChelper`, and 6) `plotrix`.
67. Import the *voom* transformed gene expression matrix, saved during **step 25.**
68. Extract column names from the gene expression matrix to construct a design matrix for DGCA.
69. Filter the gene expression matrix using the function **`filterGenes`**:
    a. First filter genes by central tendency (e.g., `filterCentralPercentile = 0.25`; this removes genes in the bottom 25th percentile of median expression).
    b. Then filter genes by dispersion (e.g., `filterDispersionPercentile = 0.25`; this removes genes in the bottom 25th percentile of the dispersion index of expression).
70. Perform differential correlation analysis on the hub gene between normal and primary tumor using the function **`ddcorAll`**.

> *Optional:* Visualize the correlation in each condition (i.e., normal, and primary tumor) between the hub gene and its top DCGs via the following command:

```
>plotCors(inputMat = dataExpr, design = dataMatrix, compare = c("dataGTEx", "dataTCGA"), ge-
neA = "hub_gene" ,geneB = "top_DCG")
```

71. Perform GO enrichment on genes with significant gain of correlation to the hub gene in primary tumor and on genes with significant loss of correlation to the hub gene in primary tumor via the DGCA wrapper function **`ddcorGO`**.

> *Note:* If users decide to re-run GO term enrichment analysis after some time (e.g., 6 months), the outcome of this later analysis may not always agree with those obtained from analysis conducted at an earlier time. For explanation and potential solution, please refer to troubleshooting, problem 5 (step 49, step 61, and step 71).

72. Visualize the odd-ratios from enrichment of GO terms derived from DCGs via the wrapper function **`plotGOTwoGroups`**.

> *Note:* For complete details on the use and execution of the DGCA R package, please refer to McKenzie et al. (2016).

EXPECTED OUTCOMES

The **`filterByExpr`** function removes genes that are either unexpressed or lowly expressed while keeping as many genes as possible with worthwhile counts (i.e., keeping genes that have CPM above *k* in *n* samples, where *n* is determined by the minimum group sample size and *k* is determined by minimum count (default: 10) in a minimum proportion (default: 70% of the smallest group size) of samples). The density plot of log-CPM values for raw vs. filtered expected count should show a sizable proportion of lowly-expressed genes (Figure 1A) that is removed after application of **`filterByExpr`** (Figure 1B). For network analysis of TCGA colon cancer gene expression, 15,164/18,205 genes were retained for differential expression analysis after application of the **`filterByExpr`** function. The definition of "lowly-expressed" is subjective. In addition to modifying the values of

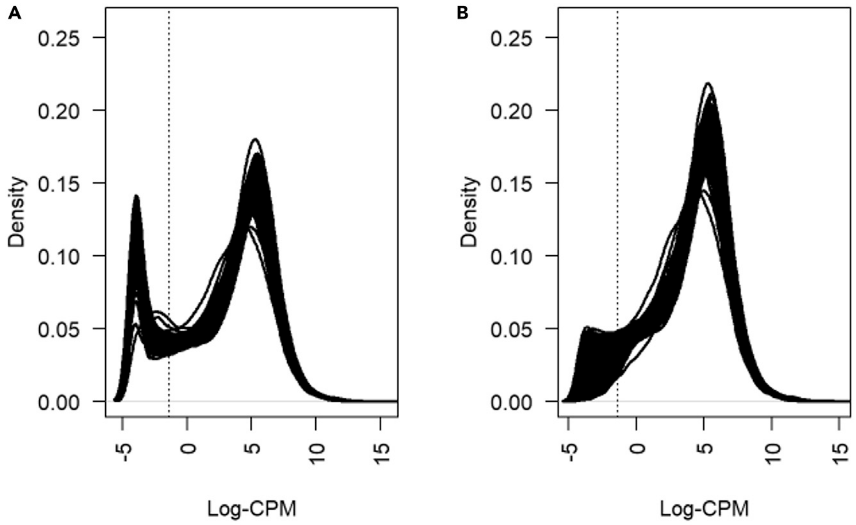


Figure 1. Impact of removing lowly-expressed genes on the distribution of expression values
(A and B) Density plot of log-CPM values before (A) and after (B) removal of genes that are lowly-expressed in TCGA primary tumor and GTEx normal colon tissue samples.

arguments for the **filterByExpr** function, the filtering of lowly expressed genes can also be achieved by functions external to **edgeR::filterByExpr** (e.g., the genefilter R package (Gentleman et al., 2021)) to maximize the number of DEGs. However, it should be noted that inadequate removal of lowly expressed genes will negatively impact linear modeling in *limma-voom*, which is carried out on log-CPM values which are assumed to be normally distributed. If the filtering of lowly-expressed genes is insufficient for linear modeling in *limma-voom*, then the mean-variance trend plot, generated as part of the **voom** function, will show a drop in variance levels at the low end of the expression scale. Figure 2 shows the expected mean-variance trend for network analysis of TCGA colon cancer gene expression. Please refer to materials (i.e., Figure 1) in Law et al. (2014) for acceptable trends of mean-variance relationship that can be applied to linear modeling with *limma-voom*.

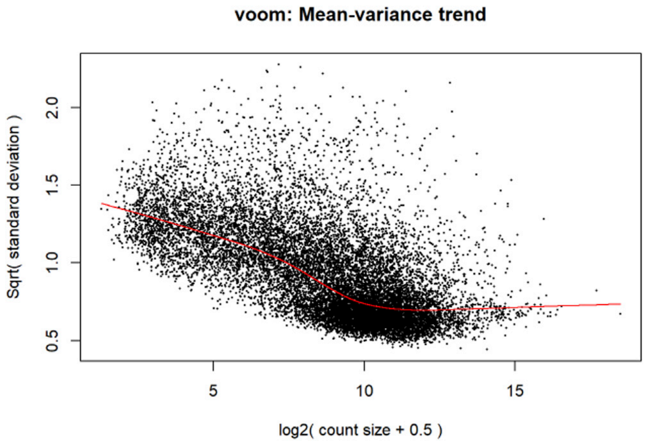


Figure 2. The mean-variance relationship of the input gene expression data
Mean-variance relationship of log-CPM values for the input dataset (TCGA primary tumor and GTEx normal colon tissue gene expression data) is appropriate for subsequent linear modeling with *limma-voom* since a drop in variance levels at the low end of the expression scale was not observed.

The number of DEGs detected across TCGA primary tumor and GTEx normal colon tissue samples accounts for 92% of total gene input (see the Methods S2 file "R Markdown Code Script for LIMMA_ColonCancer"). To focus on genes that have significant differential expressions relative to a non-zero-fold change threshold (and thus are more likely to be biologically meaningful), the **`treat`** function can be used to trim large DEG lists (McCarthy and Smyth, 2009). Upon application of an arbitrary threshold of $\log_2$FC = 0.58, the number of DEGs were reduced to account for 56% of total gene input (see the Methods S2 file "R Markdown Code Script for LIMMA_ColonCancer"). The higher the chosen threshold, the stronger the evidence needed for any particular gene to be defined as a DEG. As such, the decision to apply the **`treat`** function and the selection of threshold value are dependent on the magnitude of contrast between conditions and the size of the gene input. Another illustration of the **`treat`** function is presented in the Methods S7 file "R Markdown Code Script for LIMMA_OvarianCancer").

The results of differential expression analyses can be visualized as volcano plots, a standard in DEG visualization. A volcano plot is a type of scatter plot that illustrates the statistical significance ($-\log_{10}$(adjusted p value)) and magnitude of change ($\log_2$(fold-change)) associated with the conditions under comparison. The volcano plot seen in Figure 5B of Chen and MacDonald (2021) is the expected outcome of differential gene expression analysis on TCGA tumor vs. GTEx normal tissue. The Methods S8 file "R Markdown Code Script for Enhanced Volcano" contains the exact script used to construct a volcano plot with the refined list (i.e., have undergone **`treat`**) of DEGs saved during **step 24** of step-by-step method details using the EnhancedVolcano R package (Blighe and Lasky-Su, 2021). However, any software that can create scatter plots may be used to create volcano plots.

Quantification and statistical analyses can be performed as optional functions within specific R packages. For differential gene expression analysis with *limma-voom*, the *treat* method with a nonparametric empirical Bayes approach for the analysis of factorial data provided a paired t-test for every gene within the limma R environment. To calculate the correlation between clinical trait and *WGCNA* module eigengenes, Pearson coefficients (r) and the corresponding p-values were used to determine statistically significant linear relationships. For gene set enrichment analysis with *topGO*, the *elim* method was applied alongside Fisher's exact test to assess significance of GO term enrichment. For survival analyses with *RegParallel*, a univariate Cox survival model was used to compute hazard ratios, and outputs from log-rank testing was used to describe overall significance of the model. For DGCA, the pipeline provided Pearson coefficients (r) and the corresponding p values for each pair of genes across samples.

LIMITATIONS

The presented protocol utilizes TCGA and GTEx RNA-Seq datasets. These datasets excel at having sufficient observations for statistically sound correlation studies. However, should additional datasets that offer a juxtaposition of tumor vs. normal samples become available, these resources should be utilized for the validation of results obtained from the application of the presented protocol. The presented protocol employs gene expression profiles of tumor bulk tissues. Given that genes may demonstrate diverse functions across different cell types, gene sets identified from averaged datasets need to be reexamined in a cell type-specific manner for the identification of susceptible cell types and converged pathways among different cells. Deconvolution methods may be utilized to reconstruct cell type-specific gene expression from tumor bulk tissues (Aran et al., 2015). Finally, the presented protocol examines gene-to-gene correlations, which do not indicate causal relationships.

TROUBLESHOOTING

Problem 1

The **`subset`** function does not compute, or the **`subset`** function returns zero observations (step 6).

Potential solution

In R, unique names (i.e., identifiers) given to variables must not start with an underscore (_). However, many variables found in the `TcgaTargetGTEX_phenotype` and `clinicalMatrix` datasets contain a starting underscore (e.g., `_primary_site`).

One can utilize the **`gsub`** function to remove or replace underscores in identifiers. For example:

```
> gsub("\\_", "" ,names(YourDataset))
```

The R language is generally case sensitive. However, TCGA and GTEx data stored with the UCSC-Xena platform carry capitalization inconsistencies, which can interfere with subsetting in R. For example, in `TcgaTargetGTEX_phenotype`, the `_primary_site` for `_sample_type` "Normal Tissue" under `_study` "GTEX" is "Adrenal **G**land", whereas the `_primary_site` for `_sample_type` "Primary Tumor" under `_study` "TCGA" is "Adrenal **g**land".

One can use **`subset`** with **`grepl`** to ignore case. For example:

```
> subset(YourDataset, grepl("adrenal gland", primarysite, ignore.case = TRUE) & study ==
"GTEX")
```

Samples stored with the skin cancer clinical matrix do not have assigned histological type. When retrieving IDs for skin cancer primary tumor samples. One can use the following script:

```
> filterTCGA02 = subset(filterTCGA01, sampletype == paraSampleType & primarysite ==
paraPrimarySiteTCGA)
```

In R, certain characters hold special meaning to certain functions (i.e., metacharacters). For example, when subsetting with **`grepl`**, the first argument is a pattern, a regular expression. If we set the value for this first argument as `"^Colon"`, then **`grepl`** gives back the rows (observations) that begins with the pattern "Colon"; If we set the value for this first argument as `"Colon*"`, then **`grepl`** gives back the rows (observations) that contains the pattern "Colon". When utilizing **`grepl`**, parentheses are also metacharacters. To match literal parentheses, we need to escape the metacharacters by placing backslashes (`\\`) in front of parentheses to suppress their special meaning. When setting the values for `paraPrimaryTissueGTEx` (**step 6.a**) or `paraHistologicalType` (**step 6.b**), please refer to Tables 3 or 4, respectively, to obtain non-zero returns.

Problem 2

In general, phenotypic information stored in TCGA clinical matrices can be classified into three categories: discrete numeric (e.g., age), binomial (e.g., additional therapy?), or ordinal (e.g., clinical stage) variables. While discrete numeric variables can be used directly (i.e., without modification) to estimate the relationship of gene expression to phenotype, binomial and ordinal variables need to be re-coded for downstream analyses. Given that 1) the variables included in clinical matrices differ among the 19 cancer types listed in Table 2, and 2) the importance of research question in the context of a study design, we provide here a more generalized solution to the preparation of binomial and ordinal variables for downstream analyses (step 9).

Potential solution

The clinical matrix `TCGA.OV.sampleMap/OV_clinicalMatrix` for ovarian cancer is used to exemplify the generalized solution for re-coding nominal and ordinal variable for downstream analyses. Please refer to the Methods S9 file "R Markdown Code Script for troubleshooting problem 2" for the exact script used to generate the expected outcome.

To view a list of variables included in the clinical matrix (downloaded during **step 5.b**), one can use the following script:

```
> names(filterTCGA02)
```

To keep variable(s) of interest (e.g., "Additional Radiation Therapy" and "Clinical Stage"), one can use the following script:

```
> varClinKeep = c("sampleID", "additionalradiationtherapy", "clinicalstage")
```

To re-code the binomial variable "Additional Radiation Therapy" from YES/NO to 1/0, one can use the following script:

```
> clinFinal$additionalradiationtherapy = if_else(clinFinal$additionalradiationtherapy ==
"YES", 1, 0, missing = NULL)
```

To re-code the ordinal variable "Clinical Stage" from Stages I/II/III/IV to 1/2/3/4, one can use the following script:

```
> clinFinal$clinicalstage[grepl("^Stage IV", clinFinal$clinicalstage)] <- 4;

> clinFinal$clinicalstage[grepl("^Stage III", clinFinal$clinicalstage)] <- 3;

> clinFinal$clinicalstage[grepl("^Stage II", clinFinal$clinicalstage)] <- 2;

> clinFinal$clinicalstage[grepl("^Stage I", clinFinal$clinicalstage)] <- 1
```

Problem 3

In *limma-voom*, all samples are assumed to have a similar range and distribution of log-CPM values (Law et al., 2016). If a subset of samples appears as outliers on the density plot for filtered expected count (Figure 3), these sample outliers should be removed prior to running *limma-voom* on the DGEList-object (step 13).

Potential solution

The script used as a solution to this problem is provided in the Methods S10 file "R Markdown Code Script for troubleshooting problem 3 – Colon Cancer". In which, a `lcpm.cutoff` of $\log_2(10/M + 2/L)$, where M is the median library size in millions and L is the mean library size in millions, was applied to the `lcpm` data matrix generated during **step 15** of step-by-step method details to identify lowly-expressed genes. Next, the proportion of genes below `lcpm.cutoff` by sample was calculated then summarized with a histogram to identify sample outlier(s). For example, in determination of DEGs across TCGA primary tumor and GTEx normal colon tissue samples, we filtered out four GTEx samples that had >80% of genes below the defined `lcpm.cutoff` (Figure 3C). Another illustration of this solution is presented in the code script included in the Methods S11 file "R Markdown Code Script for troubleshooting problem 3 – Ovarian Cancer". In which, one TCGA sample was identified as outlier and was removed prior to the generation of the DGEList-object in the code script provided in the Methods S7 file "R Markdown Code Script for LIMMA_OvarianCancer" to facilitate clean execution of functions called during step 17–22.

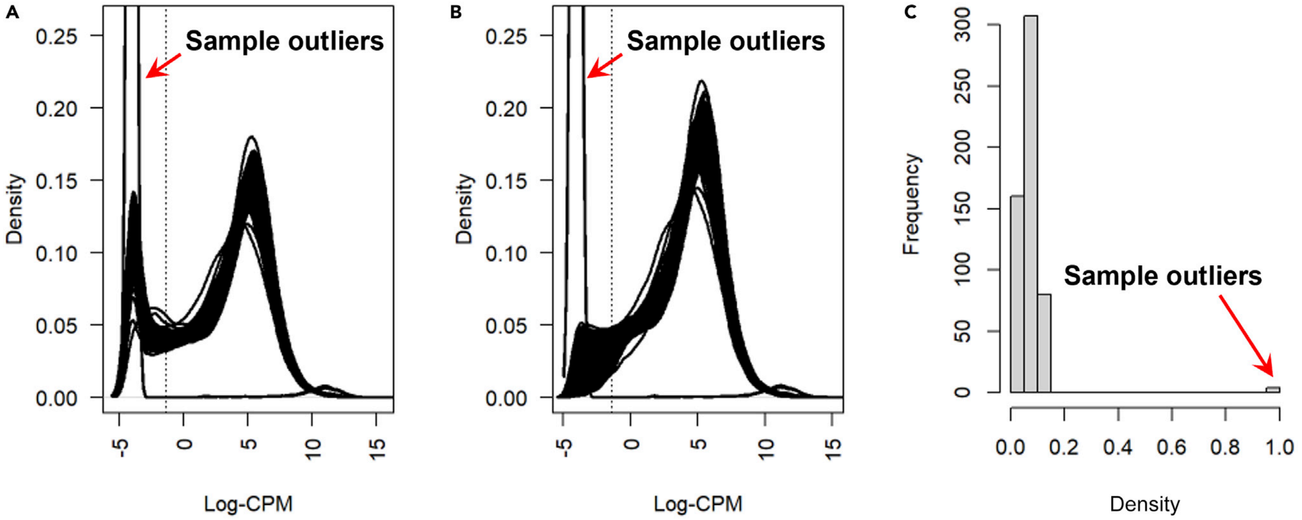


Figure 3. Identification of sample outliers on a density plot for filtered expected count

(A–C) A density plot of log-CPM values for expected count shows distinct distributions of log-CPM values before (A) and after (B) removal of genes using the **`filterByExpr`** function. The proportion of genes below `lcpm.cutoff` (indicated by the vertical dotted lines in A and B) by sample is summarized in a histogram (C), and samples with density (proportion of genes) > 0.8 for log-CPM values < `lcpm.cutoff` were defined as outliers.

Problem 4

WGCNA returns gene clusters (modules) that are labeled with colors. The module labeled turquoise contains the highest number of genes, the module labeled blue contains the second highest number of genes, then brown, green, etc. The module labeled gray contains non-module genes. Depending on the nature of the input data, and/or the argument settings in functions **`adjacency`**, **`TOMsimilarity`**, and **`cutreeDynamic`**, WGCNA may return modules that capture a large fraction of the input data (i.e., containing 10, 20, or 50% of the total input data). From the perspective of WGCNA, there is no "correct" upper-limit on the size of any module or the number of non-module genes. However, given that the typical objective of performing WGCNA is to infer underlying molecular mechanisms based on biological attributes of a given gene set (module), large modules may not provide adequate resolution for the identification of core mechanisms associated with the sample trait of interest (step 35).

Potential solution

A solution to this problem is to conduct multiple runs of WGCNA, and adjust the argument `minClusterSize` (i.e., `minModuleSize`) and/or the argument `deepSplit` under the function **`cutreeDynamic`** between runs. The `deepSplit` argument controls the sensitivity of module splitting. Users can select an integer value between 0 and 4 for `deepSplit`, with 4 being the most sensitive. The `minModuleSize` argument controls the lower limit of module size. For WGCNA on TCGA colon primary tumors, the deepSplit was set at 4. By decreasing the `minModulSize` value, we observed 1) an increased in number of modules detected, 2) a decrease in the number of genes per module, 3) the elimination of the gray module, and 4) the detection of a gene set (module) significantly correlated with the trait of interest *lymphatic invasion* (Figure 4).

Problem 5

Discordant results from GO term enrichment analysis and/or molecular network analysis when rerunning analyses later (**step 49**, **step 61**, and **step 71**).

Potential solution

In this protocol, GO term enrichment was completed by retrieving GO annotations from the Ensembl database (Durinck et al., 2005, 2009) or the org.Hs.e.g.,.db database (Carlson, 2021). And

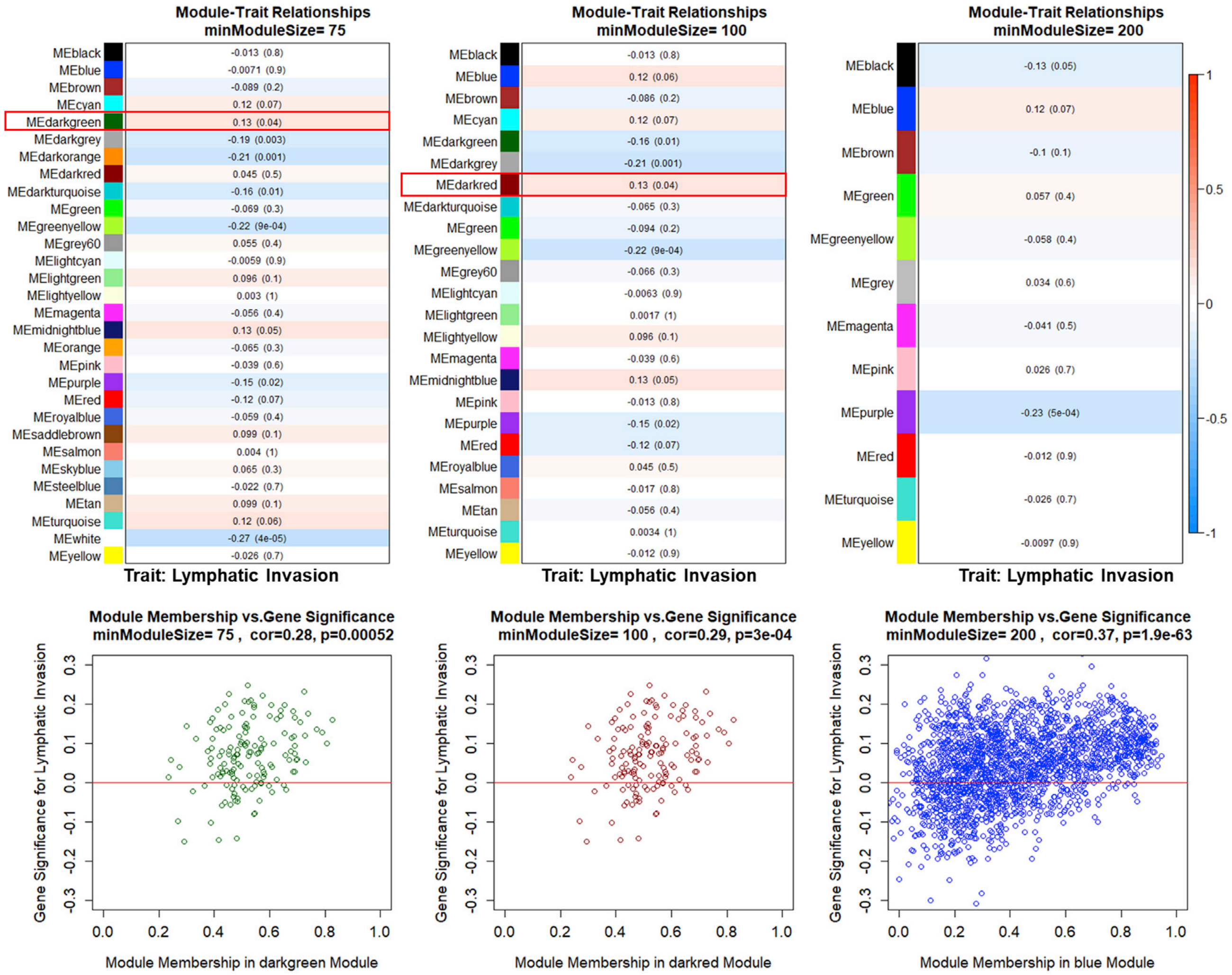


Figure 4. Comparison of module-trait relationship matrices

For the Module-Trait Relationships heat maps, the Pearson correlation value and p value (in brackets) are provided; p values below 0.05 were considered significant. Gene significance (GS) vs. module membership (MM) scatterplots were generated with varied `minModuleSize` values. Red boxes mark modules that were significantly correlated with the clinical trait of interest *lymphatic invasion*.

a PPI network was constructed utilizing the STRING database (Szklarczyk et al., 2021). Databases implement different approaches to manage annotation. Furthermore, databases are updated over time. As such, the outcome of enrichment or network analysis can change depending on which database was employed for annotation and/or when the analysis was conducted (Tomczak et al., 2018). One way to improve the reproducibility of gene set analysis is to increase the input sample size (Maleki et al., 2019). Comparing annotations from multiple databases may also improve reproducibility. Finally, one should keep track of data provenance as the outcome of enrichment or network analysis will differ when R packages are updated (e.g., org.Hs.e.g.,.db for gene annotation by Entrez ID, and biomaRt for annotation by Ensemble ID).

RESOURCE AVAILABILITY

Lead contact

Requests for further information should be directed to and will be fulfilled by the lead contact, Huey-Miin Chen (thmchen@ucalgary.ca).

Materials availability

This protocol did not generate any new materials.

Data and code availability

1. Data

This paper analyzes existing, publicly available RNA-Seq data. The sources for the datasets are listed in the key resources table.

2. Code

This paper does not report original code. All codes were used in this study in alignment with recommendations made by authors of R packages in their respective user's guides.

3. Additional information requests

Any additional information required to reanalyze the data used in this paper is available from the lead contact upon request.

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.xpro.2022.101168.

ACKNOWLEDGMENTS

This work was supported by a research grant from the Canadian Institutes of Health Research (MOP#97931 to J.A.M.). H.C. was recipient of a CIHR Fredrick Banting and Charles Best Canada Graduate Scholarship and an Alberta Graduate Excellence Scholarship.

AUTHOR CONTRIBUTIONS

H.C. completed the protocol outline, validated the materials, and wrote the manuscript. J.A.M. coordinated the study, secured grant funding support, edited the manuscript, and provided intellectual contributions to the project. All authors reviewed and approved the final version of the manuscript.

DECLARATION OF INTERESTS

Justin A. MacDonald is cofounder and a shareholder of Arch Biopartners Inc. Huey-Miin Chen has no competing interests.

INCLUSION AND DIVERSITY

One or more of the authors of this paper self-identifies as an underrepresented ethnic minority in science. While citing references scientifically relevant for this work, we also actively worked to promote gender balance in our reference list. The author list of this paper includes contributors from the location where the research was conducted who participated in the data collection, design, analysis, and/or interpretation of the work.

REFERENCES

Alexa, A., and Rahnenführer, J. (2021). topGO: enrichment analysis for gene ontology (R package version 2.44.0). https://bioconductor.org/packages/topGO/.

Alexa, A., Rahnenführer, J., and Lengauer, T. (2006). Improved scoring of functional groups from gene expression data by decorrelating GO graph structure. Bioinformatics *22*, 1600–1607. https://doi.org/10.1093/bioinformatics/btl140.

Aran, D., Camarda, R., Odegaard, J., Paik, H., Oskotsky, B., Krings, G., Goga, A., Sirota, M., and Butte, A.J. (2017). Comprehensive analysis of normal adjacent to tumor transcriptomes. Nat. Commun. *8*, 1077. https://doi.org/10.1038/S41467-017-01027-Z.

Aran, D., Sirota, M., and Butte, A.J. (2015). Systematic pan-cancer analysis of tumour purity. Nat. Commun. *6*, 1–12. https://doi.org/10.1038/ncomms9971.

Bengtsson, H. (2021). R.utils: various programming utilities (R package version 2.11.0.). https://cran.r-project.org/package=R.utils.

Blighe, K., and Lasky-Su, J. (2021). Standard regression functions in R enabled for parallel processing over large data-frames (R Package Version 1.10.0.). https://github.com/kevinblighe/RegParallel.

Carlson, M. (2021). org.Hs.eg.db: genome wide annotation for human (R package version 3.13.0.). https://bioconductor.org/packages/org.Hs.eg.db/.

Chen, H.-M., and MacDonald, J.A. (2021). Network analysis identifies DAPK3 as a potential biomarker for lymphatic invasion and colon adenocarcinoma prognosis. IScience *24*, 102831. https://doi.org/10.1016/j.isci.2021.102831.

CIOMS (2016). International ethical guidelines for health-related research involving humans (Fourth) (Council for International Organizations of Medical Sciences (CIOMS)).

Doncheva, N., Morris, J.H., Gorodkin, J., and Jensen, L.J. (2019). Cytoscape StringApp: network analysis and visualization of proteomics data. J. Proteome Res. *18*, 623–632. https://doi.org/10.1021/acs.jproteome.8b00702.

Dowle, M., and Srinivasan, A. (2021). data.table: extension of 'data.frame' (R package version 1.14.2.). https://cran.r-project.org/package=data.table.

Durinck, S., Moreau, Y., Kasprzyk, A., Davis, S., de Moor, B., Brazma, A., and Huber, W. (2005). BioMart and bioconductor: a powerful link between biological databases and microarray data analysis. Bioinformatics *21*, 3439–3440. https://doi.org/10.1093/bioinformatics/bti525.

Durinck, S., Spellman, P.T., Birney, E., and Huber, W. (2009). Mapping identifiers for the integration of genomic datasets with the R/Bioconductor package biomaRt. Nat. Protoc. *4*, 1184–1191. https://doi.org/10.1038/nprot.2009.97.

Falcon, S., and Gentleman, R. (2007). Using GOstats to test gene lists for GO term association. Bioinformatics *23*, 257–258. https://doi.org/10.1093/bioinformatics/btl567.

Gentleman, R., Carey, V.J., and Huber, W. (2021). Genefilter: Genefilter: methods for filtering genes from high-throughput experiments (R package version 1.74.1.). https://bioconductor.org/packages/genefilter/.

Kassambara, A., Kosinski, M., and Biecek, P. (2021). Survminer: drawing survival curves using "Ggplot2". (R package version 0.4.9.). https://cran.r-project.org/package=survminer.

Langfelder, P., and Horvath, S. (2008). WGCNA: an R package for weighted correlation network analysis. BMC Bioinformatics *9*, 1–13. https://doi.org/10.1186/1471-2105-9-559.

Law, C.W., Alhamdoosh, M., Su, S., Dong, X., Tian, L., Smyth, G.K., and Ritchie, M.E. (2016). RNA-seq Analysis Is Easy as 1-2-3 with Limma, Glimma and edgeR. F1000Research *5*. https://doi.org/10.12688/f1000research.9005.3.

Law, C.W., Chen, Y., Shi, W., and Smyth, G.K. (2014). voom: precision weights unlock linear model analysis tools for RNA-seq read counts. Genome Biol. *15*, R29. https://doi.org/10.1186/gb-2014-15-2-r29.

Lemon, J. (2006). Plotrix: a package in the red light district of R. (3.8-2). https://cran.r-project.org/package=plotrix.

Luo, C., Cen, S., Ding, G., and Wu, W. (2019). Mucinous colorectal adenocarcinoma: clinical pathology and treatment options. Cancer Commun. *39*, 13. https://doi.org/10.1186/S40880-019-0361-0.

Maleki, F., Ovens, K., McQuillan, I., and Kusalik, A.J. (2019). Size matters: how sample size affects the reproducibility and specificity of gene set analysis. Hum. Genomics *13*, 42. https://doi.org/10.1186/S40246-019-0226-2.

McCarthy, D.J., Chen, Y., and Smyth, G.K. (2012). Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation. Nucleic Acids Res. *40*, 4288–4297. https://doi.org/10.1093/NAR/GKS042.

McCarthy, D.J., and Smyth, G.K. (2009). Testing significance relative to a fold-change threshold is a TREAT. Bioinformatics *25*, 765–771. https://doi.org/10.1093/bioinformatics/btp053.

McDermaid, A., Monier, B., Zhao, J., Liu, B., and Ma, Q. (2019). Interpretation of differential gene expression results of RNA-seq data: review and integration. Brief. Bioinform. *20*, 2044. https://doi.org/10.1093/BIB/BBY067.

McKenzie, A.T., Katsyv, I., Song, W.M., Wang, M., and Zhang, B. (2016). DGCA: a comprehensive R package for differential gene correlation analysis. BMC Syst. Biol. *10*, 1–25. https://doi.org/10.1186/s12918-016-0349-1.

Miller, K.D., Nogueira, L., Mariotto, A.B., Rowland, J.H., Yabroff, K.R., Alfano, C.M., Jemal, A., Kramer, J.L., and Siegel, R.L. (2019). Cancer treatment and survivorship statistics, 2019. CA Cancer J. Clin. *69*, 363–385. https://doi.org/10.3322/caac.21565.

Morgan, M. (2021). BiocManager: access the bioconductor project package repository (R package version 1.30.16). https://cran.r-project.org/package=BiocManager.

Myers, M.H., and Ries, L.A.G. (1989). Cancer patient survival rates: SEER program results for 10 years of follow-up. CA Cancer J. Clin. *39*, 21–32. https://doi.org/10.3322/canjclin.39.1.21.

Oh, S., Abdelnabi, J., Al-Dulaimi, R., Aggarwal, A., Ramos, M., Davis, S., Riester, M., and Waldron, L. (2020). HGNChelper: identification and correction of invalid gene symbols for human and mouse. F1000Research *9*, 1493. https://doi.org/10.12688/f1000research.28033.1.

Piovesan, A., Antonaros, F., Vitale, L., Strippoli, P., Pelleri, M.C., and Caracausi, M. (2019). Human protein-coding genes and gene feature statistics in 2019. BMC Res. Notes *12*, 315. https://doi.org/10.1186/s13104-019-4343-8.

Ritchie, M.E., Phipson, B., Wu, D., Hu, Y., Law, C.W., Shi, W., and Smyth, G.K. (2015). Limma powers differential expression analyses for RNA-sequencing and microarray studies. Nucleic Acids Res. *43*, e47. https://doi.org/10.1093/nar/gkv007.

Robinson, M.D., McCarthy, D.J., and Smyth, G.K. (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics *26*, 139–140. https://doi.org/10.1093/bioinformatics/btp616.

Savino, A., Provero, P., and Poli, V. (2020). Differential Co-expression analyses allow the identification of critical signalling pathways altered during tumour transformation and progression. Int. J. Mol. Sci. *21*, 1–23. https://doi.org/10.3390/IJMS21249461.

Shannon, P., Markiel, A., Ozier, O., Baliga, N., Wang, J., Ramage, D., Amin, N., Schwikowski, B., and Ideker, T. (2003). Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Res. *13*, 2498–2504. https://doi.org/10.1101/gr.1239303.

Szklarczyk, D., Gable, A.L., Nastou, K.C., Lyon, D., Kirsch, R., Pyysalo, S., Doncheva, N.T., Legeay, M., Fang, T., Bork, P., et al. (2021). The STRING database in 2021: customizable protein–protein networks, and functional characterization of user-uploaded gene/measurement sets. Nucleic Acids Res. *49*, D605–D612. https://doi.org/10.1093/nar/gkaa1074.

Tomczak, A., Mortensen, J.M., Winnenburg, R., Liu, C., Alessi, D.T., Swamy, V., Vallania, F., Lofgren, S., Haynes, W., Shah, N.H., et al. (2018). Interpretation of biological experiments changes with evolution of the Gene Ontology and its annotations. Sci. Rep. *8*, 5115. https://doi.org/10.1038/s41598-018-23395-2.

Vivian, J., Rao, A.A., Nothaft, F.A., Ketchum, C., Armstrong, J., Novak, A., Pfeil, J., Narkizian, J., Deran, A.D., Musselman-Brown, A., et al. (2017). Toil enables reproducible, open source, big biomedical data analyses. Nat. Biotechnol. *35*, 314–316. https://doi.org/10.1038/nbt.3772.

Wang, S., and Liu, X. (2019). The UCSCXenaTools R package: a toolkit for accessing genomics data from UCSC Xena platform, from cancer multi-omics to single-cell RNA-seq. J. Open Source Softw. *4*, 1627. https://doi.org/10.21105/JOSS.01627.

Wickham, H., François, R., Henry, L., and Müller, K. (2021). Dplyr: a grammar of data manipulation (R package version 1.0.7.). https://cran.r-project.org/package=dplyr.

Wickham, H., Navarro, D., and Pedersen, T.L. (2016). ggplot2: Elegant Graphics for Data Analysis (Springer). https://ggplot2.tidyverse.org.

Xiao, N., Wang, G., and Sun, L. (2019). Grex: gene ID mapping for genotype-tissue expression (GTEx) data (R package version 1.9.). https://cran.r-project.org/package=grex.