Reprocessing of RNA-sequencing samples from publicly-available data to yield normalized and comparable expression results.

William C. Wright
St. Jude Children's Research Hospital https://orcid.org/0000-0002-2274-9966
Taosheng Chen (taosheng.chen@stjude.org)
St. Jude Children's Research Hospital https://orcid.org/0000-0001-6420-3809

Method Article

Keywords: RNA-Seq, TCGA, Expression, Cancer, Data Analysis, Normalization, CYP3A5, PXR, NR1I2

Posted Date: October 16th, 2019

DOI: https://doi.org/10.21203/rs.2.16081/v1

Abstract

Here we obtained RNA-sequencing data from the publicly-available Pan-Cancer analysis project performed by The Cancer Genome Atlas (TCGA). Data within this project were processed the same experimentally, and analyzed downstream by the UCSC Toil recompute project. We reprocessed the resulting gene count files in batch to obtain normalized expression, which is a step critical for proper and comparable interpretation. We describe the linear modeling and normalization protocol, and provide an example of plotting the results using a gene of interest. We perform the entire protocol using freely available packages within the R framework.

Introduction

RNA-sequencing (RNA-Seq) data is useful for answering a plethora of questions in the field of gene expression. Collections of these datasets provide mountainous quantities of data but can rarely be directly comparable[1]. This is due to various discrepancies related to batch effects, different sequencing technology, different starting materials, different processing tools, etc. However, The Cancer Genome Atlas (TCGA) has sequenced a large (>10,000) collection of tumor samples using similar methods in their Pan-Cancer (PanCan) analysis initiative[2]. These samples are non-redundant and give researchers a wealth of knowledge which can be used to study patterns across cancer types. The results have been further processed downstream by the UCSC Toil recompute initiative[3], which used the exact same analysis pipeline for all samples to produce gene counts.

Use of gene counts alone is insufficient to make comparisons between samples; the counts are affected by various factors such as total number of reads and transcript length[4]. Instead, the data need to be normalized to make useful interpretations. While several methods exist, trimmed mean of M values (TMM) normalization is among the most appropriate ways to handle samples derived from the same source[5]. Furthermore, the results need to be processed into normalized units of expression such as Fragments per Kilobase Million (FPKM), Transcripts per Million (TPM), or Counts per Million (CPM).

In this protocol we obtain processed gene counts from all cohorts of the TCGA PanCan project. We simultaneously normalize the entire dataset, and provide expression results across all cohorts and genes as Log2(Normalized CPM+1). We make use of the popular limma-voom method for estimating mean-variance relationships of log counts, generating observations weights, and performing empirical Bayes smoothening[6]. We provide step-wise instructions for how to extract expression for a single gene of interest, and examples of how to plot results. Our protocol is performed using freely-available software and packages. Furthermore, it is amenable for use with similar (non-TCGA) datasets and can be scaled up or down according to input size and computational resources.

Reagents

Starting Dataset:

1. A gene or transcript counts file of samples which have been sequenced using the same methodology. Here we used RNA-Seq data from TCGA PanCan, processed by UCSC Toil recompute to get gene counts. Files are hosted at UCSC Xena. The dataset used in this protocol is attached as the supplementary file tcga_RSEM_Hugo_norm_count.gz

2. Metadata file corresponding to gene counts, which maps the sample IDs to cancer cohort (or other descriptor). This information can also be obtained from the Xena hub in our example and formatted separately to work with downstream processing. We provide metadata specific to this protocol (TCGA PanCan) as the supplementary file metadata.csv

Software and Packages:

1. R/Rstudio (Version 3.5.2 or higher)

2. edgeR package

3. limma package

4. forcats package

5. ggplot2 package

Equipment

1. Computer with requisite software and packages

(**Note**: we recommend performing the protocol using an Rstudio server or computer with large memory capability if processing data with magnitudes comparable to that of the PanCan dataset we use here.

Procedure

*Note that all code within this protocol is *italicized* and has been written so it can be directly pasted into RStudio

Load necessary packages:

library(edgeR)

library(limma)

library(forcats)

library(ggplot2)

Import the files:

1. Set the RStudio working directory to the location of the gene counts and metadata files.

2. Import the gene counts file. **Note:** In the case of large datasets like the one used here, we recommend importing the gzipped (.gz) file. The software can unzip the file as it is imported, saving memory. Avoid viewing after opening the file and instead use the following to import :

library(readr)

tcga_RSEM_Hugo_norm_count <- read_delim("tcga_RSEM_Hugo_norm_count.gz",

"\t", escape_double = FALSE, trim_ws = TRUE)

3. Import the metadata file:

library(readr)

metadata <- read_csv("metadata.csv")

View(metadata)

Prepare counts dataframe:

1. Make a dataframe of the counts:

df <- as.data.frame(tcga_RSEM_Hugo_norm_count)

2. Set the rownames equal to the first column, then delete the first column:

row.names(df) <- df$sample

df$sample<-NULL

3. Back-transform the data. Gene counts are in units of Log2(counts+1) and we need just 'counts' as input for the downstream normalization:

df2 <- 2^df

df3 <- df2-1

df <- df3

remove(df2)

remove(df3)

4. Check to ensure that the number of variables in the counts dataframe is equal to the number of observations in the metadata. It is critical these numbers are the same (here that number is 10,535 for each).

Perform TMM normalization and Voom transformation:

dge <- DGEList(counts=df, group=metadata$Cohort)

keep <- rowSums(cpm(dge)>1) >= 3

dge <- dge[keep, keep.lib.sizes=FALSE]

GD <- factor(metadata$Cohort)

design <- model.matrix(~0+GD)

dge <- calcNormFactors(dge)

v <- voom(dge,design,plot=TRUE)

fit <- lmFit(v, design)

*Note: these commands will generally have long (~5 minute) run times for a dataset of this magnitude. The steps will run more quickly if working with smaller data (a single cohort for example).

Generate results as Log2(Normalized CPM+1):

1. Generate the normalized counts per million (CPM):

TCGA_PanCan_CPM <- cpm(dge)

2. Add pseudocount of +1:

cpm_plus1 <- TCGA_PanCan_CPM+1

3. Transform to Log2 scale:

Log2CPM_plus1_TCGA_PanCan <- log2(cpm_plus1)

Log2CPM_plus1_TCGA_PanCan <- as.data.frame(Log2CPM_plus1_TCGA_PanCan)

Export the results:

write.csv(Log2CPM_plus1_TCGA_PanCan, "Log2CPM_plus1_TCGA_PanCan.csv")

*Note: This will export a file approximately 5GB in size. To simply obtain the expression of a single gene of interest this step can be skipped and the protocol can be resumed at the next section.

Obtain normalized expression of a single gene of interest across all cohorts:

*Note: Ensure the row names are the same identifier type as gene of interest's input. (Gene symbol, Ensembl identifier, etc.) Here we use the gene CYP3A5 as an example.

Log2CPM1_CYP3A5 <- Log2CPM_plus1_TCGA_PanCan["CYP3A5",]

Transpose and reformat to obtain a final dataframe having Sample ID, Expression, and Cohort:

Log2CPM1_CYP3A5 <- t(Log2CPM1_CYP3A5)

ID <- row.names(Log2CPM1_CYP3A5)

Log2CPM1_CYP3A5 <- cbind(Log2CPM1_CYP3A5,ID)

colnames(Log2CPM1_CYP3A5)[colnames(Log2CPM1_CYP3A5)=="CYP3A5"] <- "Log2CPM1_CYP3A5"

merged <- as.data.frame(merge(Log2CPM1_CYP3A5, metadata, by.x="ID", by.y="ID", all.x=TRUE))

Change the expression values to numeric instead of a factor (to work correctly with plotting results):

values_numeric <- as.numeric(levels(merged$Log2CPM1_CYP3A5))[merged$Log2CPM1_CYP3A5]

merged <- as.data.frame(cbind(merged, values_numeric))

colnames(merged)[colnames(merged)=="values_numeric"]<-"Log2CPM1_CYP3A5"

merged$Log2CPM1_CYP3A5<-NULL

merged_CYP3A5 <- merged

Export the expression results of GeneX across all cohorts:

write.csv(merged_CYP3A5, "merged_CYP3A5.csv")

Plot the results ranked by median expression:

ggplot(merged_CYP3A5, aes(x = fct_reorder(Cohort, Log2CPM1_CYP3A5, .fun = median, .desc =FALSE,na.rm=TRUE), y = Log2CPM1_CYP3A5)) + geom_boxplot(color="steelblue4", fill="steelblue4",outlier.size=0.8, alpha=0.1, lwd=0.6, fatten=1) + theme(axis.text.x=element_text(angle=0, hjust=1, face="bold", size=10), axis.text.y=element_text(face="bold", size=10)) + geom_point(color="steelblue4", size=0.8, alpha=0.1)+ theme_bw()+ coord_flip() + xlab("TCGA Cohort")

*Note: Plots can be easily exported as high-quality SVG images from the RStudio interface.

Troubleshooting

*In this section, possible problems that may arise are broken down by protocol step, and troubleshooting tips follow.

Load necessary packages:

Usually errors along the lines of "Error : (package) was built under R version x.x.x" are okay. If an 'object not found' error is presented when installing/calling a specific package, check to ensure your R installation is version 3.5.2 or higher.

Import the files:

Sometimes when importing zipped count files of this size, R will seem to stall. It often takes several minutes for a progress bar to show up. If after 15 minutes there is no visible progress status, it is possible the machine lacks memory. You can then try breaking the file down separately (from all cohorts into groups of 5 for example).

Prepare counts dataframe:

The combination of operations in this step can yield RAM errors in R when executed all at once (transposing, back-transforming). If this occurs, try pasting in the lines one-by-one, taking care not to perform any tasks in R while the "stop" icon is present.

Perform TMM normalization and Voom transformation:

The first thing to check if presented with an error during this step is the counts dataframe matching the metadata. It is critical that the number of variables of the dataframe matches the number of observations in the metadata. For example, if the counts dataframe displays in the Environment tab as "58581 obs. of 10535 variables" , and the metadata as "10535 obs. of 2 variables" this is a match. If this number (here 10535) is different, it will interfere with processing. If this occurs, it is possible to use your preferred method to ensure they are the same (for example, using a VLOOKUP command in Excel to ensure that all IDs in the metadata have a corresponding expression value in the counts file).

This is also one of the longest steps of the protocol, and for us took 17 minutes. Allow for time to process, especially if pasting all code in this section into the interface at once.

Generate results as Log2(Normalized CPM+1):

If any error occurs here, make sure you've correctly followed the steps in the section "Prepare counts dataframe". If pseudocounts were not added, the command will attempt to take the Log2 value of 0 in cases where the expression value is 0, rendering an error.

Export the results:

Exporting the results as the entire dataset is the longest step of the protocol and produces a file of ~5GB in our example. It is possible that a memory error ("cannot allocate memory") will occur. If this occurs on an RStudio server, contact the server administrator and ask to see if it is feasible to be granted more memory. Another approach is to clear out files located in the working directory to free up memory.

If memory errors occur on a local RStudio installation, try repeating the protocol starting from a smaller dataset. It is possible to utilize this protocol on cohorts of interest rather than all cohorts.

*Note that this step is only for exporting the tabular results, and is unnecessary if the goal is just to obtain a plot of a single gene of interest.

Obtain normalized expression of a single gene of interest across all cohorts:

The most common reason for an error is that the gene of interest identifier does not match that of the results. To troubleshoot this, check that the gene identifier is in the same format (gene symbol, Ensembl ID etc). If you try to pull out expression results for GAPDH but the expression file lists this gene identifier as "ENSG00000111640" an error will occur. Query the correct identifier (or change all identifiers to match using a conversion tool such as Ensembl martview).

*Note: the query must exactly match the results file in order to pull out your gene of interest. This can be particularly problematic with Ensembl gene identifiers having 'versions' appended to them. GADPH, ENSG00000111640, and ENSG00000111640.15 all represent the same gene but do not match up in value and will cause an error unless identifier type is changed.

Plot the results ranked by median expression:

It is important that the expression column of the resulting dataframe is classified as numeric. Otherwise it will interfere with plotting. If the section "Change the expression values to numeric instead of a factor" was correctly followed, the column "ID" should be a factor, "Cohort" should be a Character, and "Log2CPM1_CYP3A5" should be numeric (See Figure 3).

Time Taken

*Note that benchmarking of run times was performed on a RStudio server with large RAM. In cases where only standard equipment is available the run times will be longer, although still reasonable.

***Total time is approximately 50 minutes**

Load necessary packages:

<1 minute

Import the files:

3 minutes, 20 seconds

Prepare counts dataframe:

1 minute

Perform TMM normalization and Voom transformation:

17 minutes

Generate results as Log2(Normalized CPM+1):

1 minute

Export the results:

20 minutes

Obtain normalized expression of a single gene of interest across all cohorts:

<1 minute

Transpose and reformat to obtain a final dataframe having Sample ID, Expression, and Cohort:

<1 minute

Change the expression values to numeric instead of a factor (to work correctly with plotting results):

<1 minute

Export the expression results of GeneX across all cohorts:

<1 minute

Plot the results ranked by median expression:

<1 minute

Anticipated Results

During the protocol, the mean-variance trend plot (produced by voom) will be shown. If interested in interpretation of results, please see the voom manuscript.

Upon protocol completion, the normalized expression results and their corresponding sample IDs and cohorts should be exported in tabular format. An example of the results for a gene of interest (CYP3A5) is shown in Figure 1. To follow the protocol example and check results, the entire CYP3A5 results are available in the Supplementary File ExpressionResults_CYP3A5.csv. Note that these files (especially for multiple genes) will be large. If this data has been successfully plotted, it should look like Figure 2. If the data cannot be plotted, the first thing to check is that all column types are correct (shown in Figure 3, discussed in Troubleshooting). It is also possible to query the expression of several genes of interest one-by-one (Figure 4). This is easily achieved by simply changing the gene name (here "CYP3A5") in all appropriate commands and re-executing them.

Results can be plotted according to preference. Orientation, colors, transparency, dot size, etc., can all be easily manipulated by changing parameters within the plotting code (Figure 5).

References

1. Wang Q, Armenia J, Zhang C, et al. Unifying cancer and normal RNA sequencing data from different sources. *Sci Data*. 2018;5:180061. Published 2018 Apr 17. doi:10.1038/sdata.2018.61

2. Weinstein JN, Collisson EA, et al. The Cancer Genome Atlas Pan-Cancer analysis project. *Nat Genet*. 2013;45(10):1113–1120. doi:10.1038/ng.2764

3. Vivian J, Rao AA, Nothaft FA, et al. Toil enables reproducible, open source, big biomedical data analyses. *Nat Biotechnol*. 2017;35(4):314–316. doi:10.1038/nbt.3772

4. Conesa A, Madrigal P, Tarazona S, et al. A survey of best practices for RNA-seq data analysis [published correction appears in Genome Biol. 2016;17(1):181]. *Genome Biol*. 2016;17:13. Published 2016 Jan 26. doi:10.1186/s13059-016-0881-8

5. Robinson MD, Oshlack A. A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biol*. 2010;11(3):R25. doi:10.1186/gb-2010-11-3-r25

6. Law CW, Chen Y, Shi W, Smyth GK. voom: Precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biol*. 2014;15(2):R29. Published 2014 Feb 3. doi:10.1186/gb-2014-15-2-r29

Acknowledgements

We thank Drs. Gang Wu and Ti-Cheng Chang (Center for Applied Bioinformatics, St. Jude Children's Research Hospital) for assistance with our analysis. We also thank other members of the Chen laboratory for valuable discussions and help.

Figures

Sample ID	Log2(NormalizedCPM+1)_CYP3A5	Cohort
TCGA-02-0047-01	1.495876745	GBM
TCGA-02-0055-01	0.110222131	GBM
TCGA-02-2483-01	0.282744682	GBM
TCGA-02-2485-01	1.446231801	GBM
TCGA-04-1331-01	0.59351725	OV
TCGA-04-1332-01	1.37806812	OV
TCGA-04-1337-01	0.878662271	OV
TCGA-04-1338-01	5.008136155	OV
TCGA-04-1341-01	0.834002285	OV
TCGA-04-1343-01	0.411467054	OV
TCGA-04-1347-01	0.154993667	OV
TCGA-04-1348-01	0.269029363	OV
TCGA-04-1350-01	0.260239418	OV
TCGA-04-1356-01	1.54353813	OV
TCGA-04-1357-01	3.560376271	OV
TCGA-04-1361-01	0.444549232	OV
TCGA-04-1362-01	2.269351514	OV
TCGA-04-1364-01	0.198246844	OV
TCGA-04-1365-01	0.970333785	OV

Figure 1

Normalized expression results for a single gene of interest. Sample ID, Expression, and Cohort all corresponding to CYP3A5.

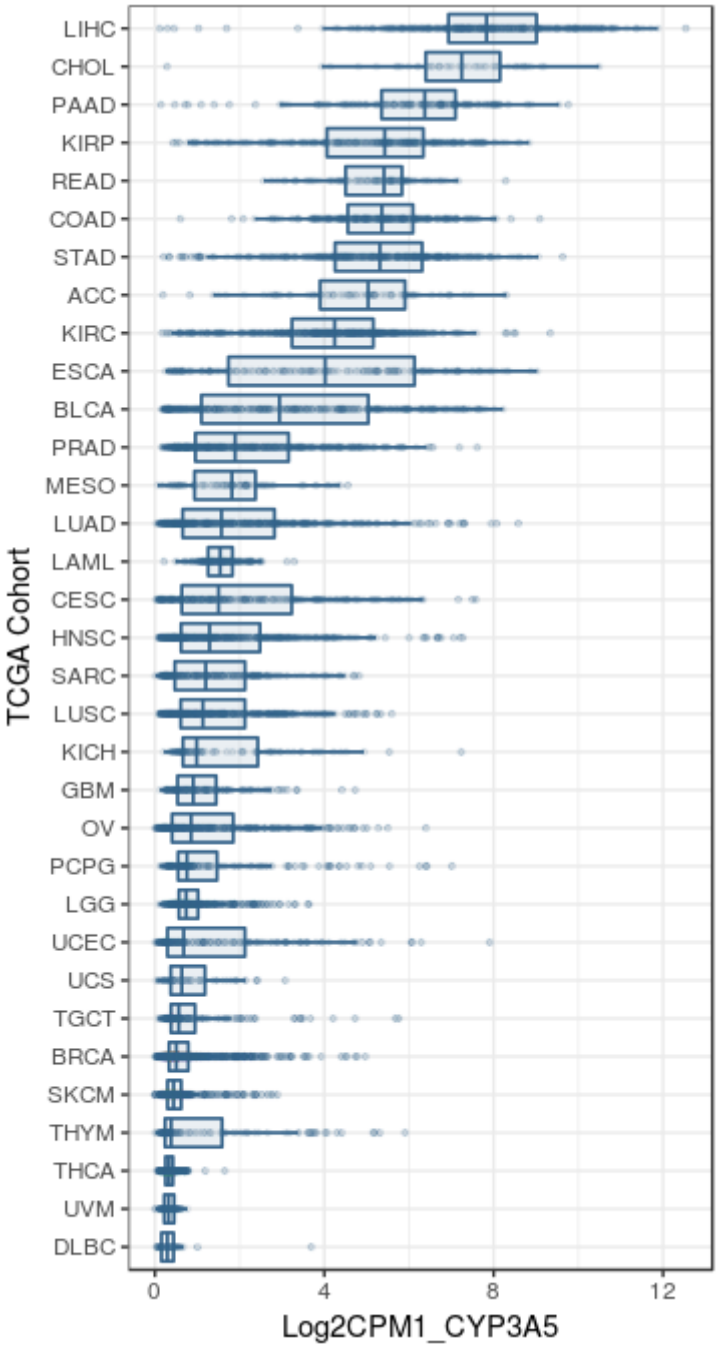
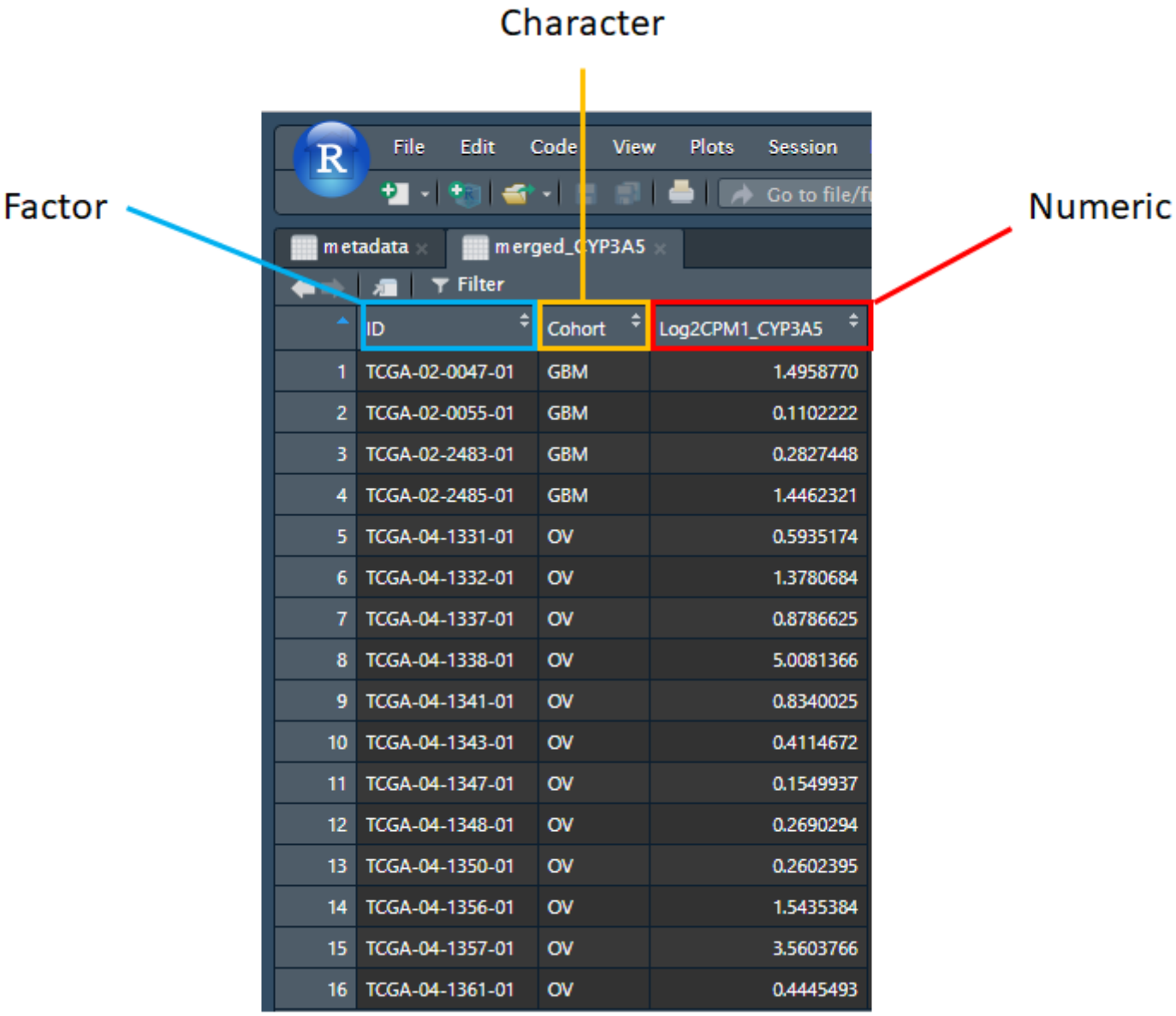


Figure 2

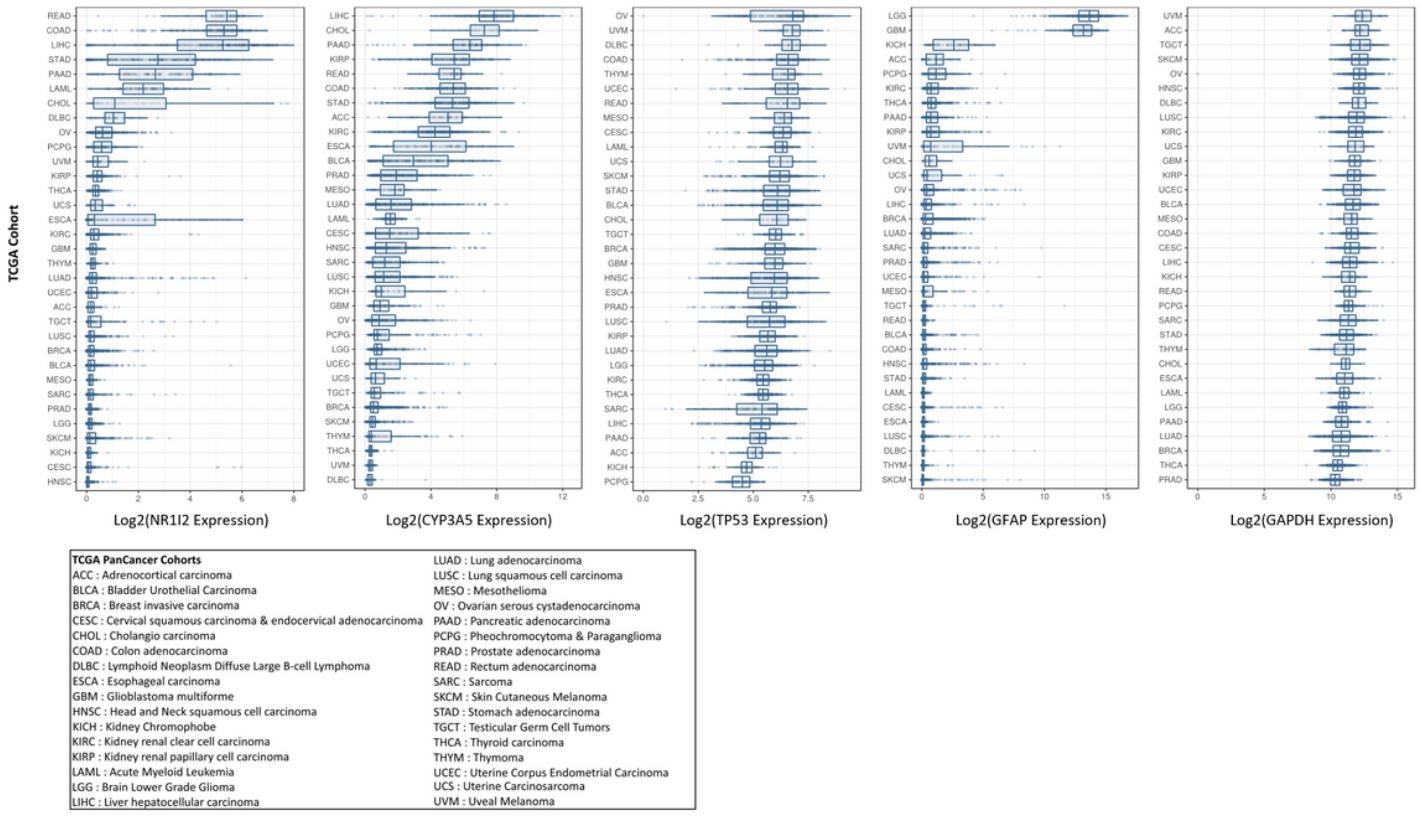
Plot of normalized CYP3A5 expression across all cohorts.



	ID	Cohort	Log2CPM1_CYP3A5
1	TCGA-02-0047-01	GBM	1.4958770
2	TCGA-02-0055-01	GBM	0.1102222
3	TCGA-02-2483-01	GBM	0.2827448
4	TCGA-02-2485-01	GBM	1.4462321
5	TCGA-04-1331-01	OV	0.5935174
6	TCGA-04-1332-01	OV	1.3780684
7	TCGA-04-1337-01	OV	0.8786625
8	TCGA-04-1338-01	OV	5.0081366
9	TCGA-04-1341-01	OV	0.8340025
10	TCGA-04-1343-01	OV	0.4114672
11	TCGA-04-1347-01	OV	0.1549937
12	TCGA-04-1348-01	OV	0.2690294
13	TCGA-04-1350-01	OV	0.2602395
14	TCGA-04-1356-01	OV	1.5435384
15	TCGA-04-1357-01	OV	3.5603766
16	TCGA-04-1361-01	OV	0.4445493

Figure 3

Column types if protocol has been properly executed. Column types can be checked within the RStudio environment by hovering over with cursor.



TCGA PanCancer Cohorts	
ACC : Adrenocortical carcinoma	LUAD : Lung adenocarcinoma
BLCA : Bladder Urothelial Carcinoma	LUSC : Lung squamous cell carcinoma
BRCA : Breast invasive carcinoma	MESO : Mesothelioma
CESC : Cervical squamous carcinoma & endocervical adenocarcinoma	OV : Ovarian serous cystadenocarcinoma
CHOL : Cholangio carcinoma	PAAD : Pancreatic adenocarcinoma
COAD : Colon adenocarcinoma	PCPG : Pheochromocytoma & Paraganglioma
DLBC : Lymphoid Neoplasm Diffuse Large B-cell Lymphoma	PRAD : Prostate adenocarcinoma
ESCA : Esophageal carcinoma	READ : Rectum adenocarcinoma
GBM : Glioblastoma multiforme	SARC : Sarcoma
HNSC : Head and Neck squamous cell carcinoma	SKCM : Skin Cutaneous Melanoma
KICH : Kidney Chromophobe	STAD : Stomach adenocarcinoma
KIRC : Kidney renal clear cell carcinoma	TGCT : Testicular Germ Cell Tumors
KIRP : Kidney renal papillary cell carcinoma	THCA : Thyroid carcinoma
LAML : Acute Myeloid Leukemia	THYM : Thymoma
LGG : Brain Lower Grade Glioma	UCEC : Uterine Corpus Endometrial Carcinoma
LIHC : Liver hepatocellular carcinoma	UCS : Uterine Carcinosarcoma
	UVM : Uveal Melanoma

Figure 4

Plot of normalized results using several examples. GFAP is a known brain-specific gene; GADPH is a known evenly-expressed gene.

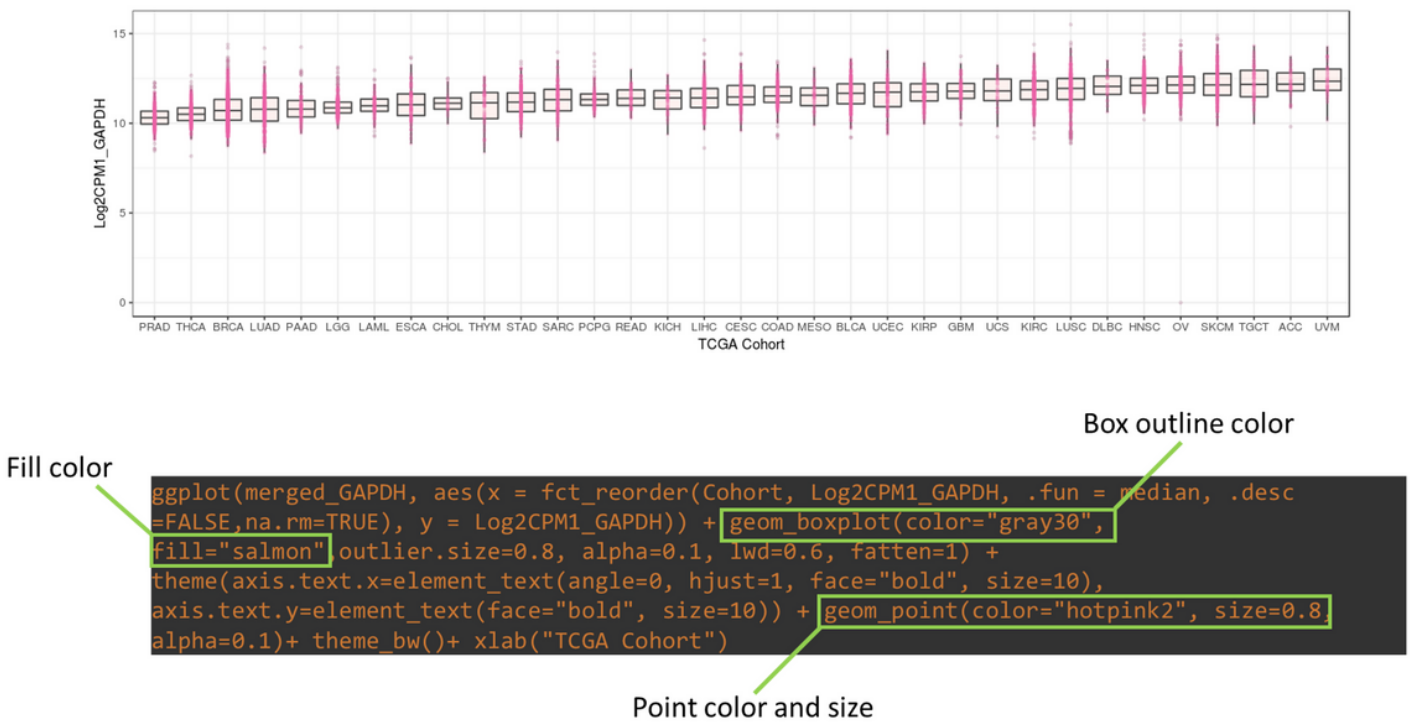


Figure 5

Results can be plotted in various ways by changing orientation, color, dot size, etc.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- ExpressionResultsCYP3A5.csv
- tcgaRSEMHugonormcount.gz
- metadata.csv