
bioexpress Documentation

Release v-5.0

Ned Cauley

Oct 08, 2021

CONTENTS

| | | |
|----------|---|----------|
| 1 | Description | 3 |
| 2 | Running the Pipeline | 5 |
| 3 | Pipeline Overview | 7 |
| 3.1 | BioXpress Downloader Step | 7 |
| 3.2 | BioXpress Annotation Step | 10 |
| 3.3 | BioXpress DESeq step | 13 |
| 3.4 | BioXpress Publisher Step | 16 |
| 3.5 | Major Changes from v-4.0 | 17 |
| 3.6 | Post-processing for OncoMX and Glygen | 19 |

Last Updated August 2021 by Ned Cauley

DESCRIPTION

The BioXpress pipeline takes raw count data from TCGA studies for both Primary Tumor and Normal Tissue and performs differential expression.

The TCGA studies included in the BioXpress pipeline are (by **tissue**):

- **Bladder** - TCGA-BLCA (Bladder urothelial carcinoma)
- **Breast** - TCGA-BRCA (Breast invasive carcinoma)
- **Colorectal** - TCGA-COAD (Colon adenocarcinoma) - TCGA-READ (Rectum adenocarcinoma)
- **Esophageal** - TCGA-ESCA (Esophageal carcinoma)
- **Head and Neck** - TCGA-HNSC (Head and Neck squamous cell carcinoma)
- **Kidney** - TCGA-KICH (Kidney Chromophobe) - TCGA-KIRP (Kidney renal papillary cell carcinoma) - TCGA-KIRC (Kidney renal clear cell carcinoma)
- **Liver** - TCGA-LIHC (Liver hepatocellular carcinoma)
- **Lung** - TCGA-LUAD (Lung adenocarcinoma) - TCGA-LUSC (Lung squamous cell carcinoma)
- **Prostate** - TCGA-PRAD (Prostate adenocarcinoma)
- **Stomach** - TCGA-STAD (Stomach adenocarcinoma)
- **Thyroid** - TCGA-THCA (Thyroid carcinoma)
- **Uterine** - TCGA-UCEC (Uterine Corpus Endometrial Carcinoma)

RUNNING THE PIPELINE

To run the BioXpress pipeline, you need to download the scripts from the HIVE Lab github repo: [GW HIVE Backend Code Repository](#) If running Bioxpress on a HIVE Lab server (such as glygen-vm-dev), place scripts in your user folder `/home/$yourusername/`. Data and other output from the scripts is stored in `/data/projects/bioxpress/`.

PIPELINE OVERVIEW

Step 1: Downloader

The downloader step will use sample sheets obtained from [GDC Data Portal](<https://portal.gdc.cancer.gov/repository>) to download raw counts from RNA-Seq for Primary Tumor and Normal Tissue in all available TCGA Studies.

Index for downloader:

3.1 BioXpress Downloader Step

Step 1 of the BioXpress pipeline

The downloader step will use sample sheets obtained from [GDC Data Portal](#) to download raw counts from RNA-Seq for Primary Tumor and Normal Tissue in all available TCGA Studies.

3.1.1 General Flow of Scripts

```
get_data_all_samples.sh -> get_hits_into_dir.py -> merge_files_tumor_and_normal.sh
```

3.1.2 Procedure

Downloader Step 1 : Get sample list files from the GDC Data Portal

Summary

Sample sheets are downloaded from the GDC data portal and used for the downstream scripts to obtain read count files.

Method

Go to the [GDC Repository](#)

Click on the button labeled **Advanced Search** on the upper right of the repository home page.

- All of the filters can also be selected manually using the search tree on the left side of the page at the link above.
To select a files filter or a cases filter, that tab must be selected on the search bar

To get the Primary Tumor samples, enter the following into the query box

```
files.analysis.workflow_type in ["HTSeq - Counts"] and files.data_type in ["Gene_
↳Expression Quantification"]
and cases.samples.sample_type in ["Primary Tumor"] and cases.project.program.name in [
↳"TCGA"]
```

Click Submit Query

The search results screen will now appear. On this screen, click Add All Files To Cart Then select the Cart on the upper right of the page.

Click Sample Sheet from the Cart page to download the Sample Sheet for the Primary Tumor samples.

- You will need to change the name for the sample sheet, otherwise when we download the sample sheet for Normal tissues it will have the same file name and overwrite the previous file. Add tumor or normal to the file names when downloaded.

Remove all samples from the cart, then repeat Step 1 for the Normal Tissue samples.

In the Advanced Search query box add enter the following for Normal Tissue samples:

```
files.analysis.workflow_type in ["HTSeq - Counts"] and files.data_type in ["Gene_
↳Expression Quantification"]
and cases.samples.sample_type in ["Solid Tissue Normal"] and cases.project.program.name_
↳in ["TCGA"]
```

Once both Sample Sheets are downloaded, Primary Tumor and Normal Tissue, move both sample sheets to the server on which the pipeline will be ran, to the path /data/projects/bioexpress/\$version/downloads/ where \$version one increment higher then the latest version of BioXpress i.e. latest version is v-4.0 so new run will be v-5.0.

Downloader Step 2: Run the script get_data_all_samples.sh

Summary

The shell script get_data_all_samples.sh provides arguments to the python script get_data_all_samples.py. It generates a log file that is used to create directories and filter out TCGA studies with low sample numbers.

Method

Edit the hard-coded paths in the script get_data_all_samples.sh

- The shell script will call the python script once for the tumor samples and once for the normal sample, so for both tumor and normal you will need to specify the path to the appropriate sample sheet and the path to the log file

Edit a hard-coded path in the script get_data_all_samples.py

- Edit the line (~line 44) path0 = "/data/projects/bioexpress/\$version/downloads" with the version for your current run of bioexpress.

Run the shell script sh get_data_all_samples.sh

Output

After the script has completed, you will have a folder for each TCGA study with all read count files compressed into a file `results.tar.gz`. You will also have three log files, one each for Tumor and Normal as well as a third log file that is the two combined `get_data_all_samples.log`

Downloader Step 3: `get_hits_into_dir.py`

Summary

The python script `get_hits_into_dir.py` decompresses all read count files and uses the log file generated in the previous script to filter out all TCGA studies that have less than 10 Normal Tissue samples. Count files are generated and labeled as *intermediate* because they will be further manipulated in later Steps

Method

Edit the hard-coded paths in `get_hits_into_dir.py`

- Edit the line (line ~12) with `open("/data/projects/bioxpress/$version/downloads/get_data_all_samples.log", 'r')` as `fil:` with the version for your current run of BioXpress - Ensure that the log file is the joined log file from the previous script, it should contain information for both Primary Tumor and Solid Tissue normal
- Edit the line (line ~44) `topDir = "/data/projects/bioxpress/$version/downloads/"` with the version for your current run of BioXpress

Run the python script `python get_hits_into_dir.py`

Output

For each TCGA study there will be a folder named `$study_$sampletype_intermediate` that contains a read count file for each sample within that study.

Downloader Step 4: Run the script `merge_files_tumor_and_normal.sh`

Summary

The shell script `merge_files_tumor_and_normal.sh` provides arguments to the python script `merge_files_tumor_and_normal.py`. All read count files for Tumor and Normal from the intermediate folders are merged so that there is one read count file per study (All samples as fields and one row per gene) and one category file per study (defines whether a sample ID corresponds to Primary-Tumor or Solid Tissue Normal).

Method

Edit the hard-coded paths in `merge_files_tumor_and_normal.sh`

- Specify the paths for the variables `in_dir` and `out_dir`

Run the shell script `sh merge_files_tumor_and_normal.sh`

Output

The `out_dir` specified in `merge_files_tumor_and_normal.sh` contains two files per study, one for counts and one for categories. The counts file contains all read counts for that study for each gene and provide sample IDs as the fields. The categories file contains information on each sample ID as either Primary Tumor or Solid Tissue Normal.

For checking sample names and numbers lists from v-5.0, all lists and the sample log have been moved to the folder `downloads/v-5.0/sample_lists`.

Step 2: Annotation

The annotation step maps transcripts to gene symbols and creates the organized read count and category files used in the DESeq step.

Index for annotation:

3.2 BioXpress Annotation Step

Step 2 of the BioXpress pipeline

3.2.1 General Flow of Scripts

`merge_per_study.sh` -> `merge_per_tissue.py` -> `split_per_case.py`

3.2.2 Procedure

Annotation Step 1 : Run the script `merge_per_study.sh`

Summary

The shell script `merge_per_study.sh` provides arguments to the python script `merge_per_study.py`. This step maps all ENSG IDs to gene symbols based on a set of mapping files. It will also filter out microRNA genes. The steps for creating the mapping files are described in the annotation README.

Method

The mapping files are available in the folder `/annotation/mapping_files/` and moved to a similar path in the version of your run of Bioxpress

- `mart_export.txt`
- `mart_export_remap_retired.txt`
- `new_mappings.txt`

Edit the hard-coded paths in `merge_per_study.sh`

- Specify the `in_dir` as the folder containing the final output of the Downloader step, count and category files per study.
- Specify the `out_dir` so that it is now in the top folder `generated/annotation` not `downloads`
- Specify the location of the mapping files downloaded in the previous sub-step

Validate the file `studies.dat` contains all studies that you wish to process

Run the shell script `sh merge_per_study.sh`

Output

All ENSG IDs in the counts files have been replaced by gene symbols in new count files located in the `out_dir`. Transcripts have also been merged per gene and microRNA genes filtered out. The categories files remain the same but are copied over to the annotation folder.

Annotation Step 2 : Run the script `merge_per_tissue.py`

Summary

The python script `merge_per_tissue.py` takes all files created by the script `merge_per_study.sh` and merges these files based on the file `tissues.csv`, which assigns TCGA studies to specific tissues terms.

Method

Download the files `tissues.csv` from the previous version of BioXpress at `/data/projects/bioxpress/$version/generated/misc/tissues.csv` and place in a similar folder in the version of your run of BioXpress

Edit the hard-coded paths in `merge_per_tissue.py`

- Edit the line (line ~23) `in_file = "/data/projects/bioxpress/v$version/generated/misc/tissues.csv"` with the version for your current run of BioXpress
- Edit the line (line ~36) `out_file_one = "/data/projects/bioxpress/v-5.0/generated/annotation/per_tissue/%s.htseq.counts" % (tissue_id)` with the version for your current run of BioXpress
- Edit the line (line ~37) `out_file_two = "/data/projects/bioxpress/v-5.0/generated/annotation/per_tissue/%s.categories" % (tissue_id)` with the version for your current run of BioXpress
- Edit the line (line ~45) `in_file = "/data/projects/bioxpress/v-5.0/generated/annotation/per_study/%s.categories" % (study_id)` with the version for your current run of BioXpress

- Edit the line (line ~52) `in_file = "/data/projects/bioexpress/v-5.0/generated/annotation/per_study/%s.htseq.counts" % (study_id)` with the version for your current run of BioXpress

Run the python script `python merge_per_tissue.py`

Output

Read count and category files are generated for each tissue specified in the `tissues.csv` file.

Annotation Step 3 : Run the script `split_per_case.py`

Summary

The python script `split_per_case.py` takes case and sample IDs from the sample sheets downloaded from the GDC data portal and splits annotation data so that there is one folder per case with only that case's annotation data.

Method

Edit the hard-coded paths in `split_per_case.py`

- Edit the line (line ~29) `in_file = "/data/projects/bioexpress/v-5.0/generated/misc/studies.csv"` with the version for your current run of BioXpress
- Edit the line (line ~38) `in_file = "/data/projects/bioexpress/v-5.0/downloads/sample_list_from_gdc/gdc_sample_sheet.primary_tumor.tsv"` with the version for your current run of BioXpress as well as the same of the sample sheet for tumor samples downloaded from the GDC data portal
- Edit the line (line ~57) `in_file = "/data/projects/bioexpress/v-5.0/downloads/sample_list_from_gdc/gdc_sample_sheet.solid_tissue_normal.tsv"` with the version for your current run of BioXpress as well as the same of the sample sheet for normal samples downloaded from the GDC data portal
- Edit the line (line ~81) `out_file_one = "/data/projects/bioexpress/v-5.0/generated/annotation/per_case/%s.%s.htseq.counts" % (study_id,case_id)` with the version for your current run of BioXpress
- Edit the line (line ~82) `out_file_two = "/data/projects/bioexpress/v-5.0/generated/annotation/per_case/%s.%s.categories" % (study_id,case_id)` with the version for your current run of BioXpress
- Edit the line (line ~85) `in_file = "/data/projects/bioexpress/v-5.0/generated/annotation/per_study/%s.htseq.counts" % (study_id)` with the version for your current run of BioXpress

Run the python script `python split_per_case.py`

Output

A folder is generated for each case ID that has a tumor sample and a normal tissue sample. Two files are generated per case: read counts and categories. These files are needed to run DESeq per case.

Step 3: DESeq

DESeq is used to calculate differential expression and determine statistical significance.

Index for DESeq:

3.3 BioXpress DESeq step

Step 3 of the BioXpress pipeline.

3.3.1 General Flow of Scripts

run_per_study.py -> run_per_tissue.py -> run_per_case.py

3.3.2 Procedure

DESeq step 1: Run the script run_per_study.sh

Summary

The python script run_per_study.py provides arguments to the R script deseq.R. The count and category files generated from the Annotation step are used to calculate differential expression and statistical significance. The result is a series of files per tissue including the normalized reads (DESeq normalization method), the DE results and significance, and QC files such as the PCA plot.

- Note: this step is time consuming (~2-3 hours of run time)

Method

Edit the hard-coded paths in the script run_per_tissue.py

- Specify the `in_dir` to be the folder containing the final output files of the Annotation steps for per study
- Specify the `out_dir`
- Ensure that the file `list_files/studies.csv` contains all of the tissues you wish to process - Note: the studies can be run separately (in the event that 2-3 hours cannot be dedicated to run the all studies at once) by creating separate dat files with specific tissues to run

Run the shell script `sh run_per_study.sh`

- Note: the R libraries specified in deseq.R will need to be installed if running on a new server or system, as these installations are not included in the scripts

Output

A set of files:

- log file
- deSeq_reads_normalized.csv - Normalized read counts (DESeq normalization method applied)
- results_significance.csv - log2fc differential expression results and statistical significance (t-test)
- dispersion.png
- distance_heatmap.png
- pca.png - Principal component analysis plot, important for observing how well the Primary Tumor and Solid Tissue Normal group together

DESeq Step 2 : Run the script run_per_tissue.sh

Summary

The python script run_per_tissue.py provides arguments to the R script deseq.R. The count and category files generated from the Annotation step are used to calculate differential expression and statistical significance. The result is a series of files per study including the normalized reads (DESeq normalization method), the DE results and significance, and QC files such as the PCA plot.

- Note: this step is time consuming (~2-3 hours of run time)

Method

Edit the hard-coded paths in the script run_per_tissue.py

- Specify the in_dir to be the folder containing the final output files of the Annotation steps for per tissue
- Specify the out_dir
- Ensure that the file list_files/tissue.dat contains all of the tissues you wish to process - Note: the tissues can be run separately (in the event that 2-3 hours cannot be dedicated to run the all tissues at once) by creating separate dat files with specific tissues to run

Run the shell script `sh run_per_tissue.sh`

Output

A set of files:

- log file
- deSeq_reads_normalized.csv - Normalized read counts (DESeq normalization method applied)
- results_significance.csv - log2fc differential expression results and statistical significance (t-test)
- dispersion.png
- distance_heatmap.png
- pca.png - Principal component analysis plot, important for observing how well the Primary Tumor and Solid Tissue Normal group together

DESeq Step 3 : Run the script run_per_case.sh

Summary

The python script run_per_case.py provides arguments to the R script deseq.R. The count and category files generated from the Annotation step are used to calculate differential expression and statistical significance. The result is a series of files per case including the normalized reads (DESeq normalization method), the DE results and significance, and QC files such as the PCA plot.

- Note: this step is time consuming (~2-3 hours of run time)

Method

Edit the hard-coded paths in the script run_per_case.py

- Specify the `in_dir` to be the folder containing the final output files of the Annotation step for per_case
- Specify the `out_dir`
- Ensure that the file `list_files/cases.csv` contains all of the cases you wish to process - Note: the cases can be run separately (in the event that 2-3 hours cannot be dedicated to run the all tissues at once) by creating separate dat files with specific cases to run

Run the shell script `sh run_per_tissue.sh`

Output

A set of files:

- log file
- `deSeq_reads_normalized.csv` - Normalized read counts (DESeq normalization method applied)
- `results_significance.csv` - log2fc differential expression results and statistical significance (t-test)
- `dispersion.png`
- `distance_heatmap.png`
- `pca.png` - Principal component analysis plot, important for observing how well the Primary Tumor and Solid Tissue Normal group together

Step 4: Publisher

Differential expression results for each tissue are combined into one master dataset.

Index for publisher:

3.4 BioXpress Publisher Step

Step 4 of the BioXpress pipeline.

3.4.1 General Flow of Scripts

de-publish-per-study.py -> de-publish-per-tissue.py

3.4.2 Procedure

Publisher Step 1 : Run the script de-publish-per-study.py

Summary

The python script de-publish-per-study.py takes the output from running DESeq in the previous step for each TCGA study and combines into one master file.

Method

Edit the hard-coded paths in the script de-publish-per-study.py

- Specify the `in_file` for the disease ontology mapping file (line ~26)
- Specify the `in_file` for the uniprot accession id (protein id) mapping file (line ~40)
- Specify the `in_file` for the refseq mapping file (line ~51)
- Specify the `in_file` for the list of TCGA studies to include in the final output (line ~72)
- Specify the `deseq_dir` for the folder containing all deseque output (line ~80)
- Specify the path to write the output (line ~135)

Run the python script `python de-publish-per-study.py`

Output

A csv file with the DESeq output for all TCGA studies, mapped to DO IDs, uniprot accession ids, and refseq ids. The path is specified in the script as one of the hard-coded lines edited during the method.

Publisher Step 2 : Run the script de-publish-per-tissue.py

Summary

The python script de-publish-per-tissue.py takes the output from running DESeq in the previous step for each tissue and combines into one master file.

Method

Edit the hard-coded paths in the script de-publish-per-study.py

- Specify the `in_file` for the disease ontology mapping file (line ~26)
- Specify the `in_file` for the uniprot accession id (protein id) mapping file (line ~40)
- Specify the `in_file` for the refseq mapping file (line ~51)
- Specify the `in_file` for the list of tissues to include in the final output (line ~72)
- Specify the `deseq_dir` for the folder containing all deseq output (line ~80)
- Specify the path to write the output (line ~135)

Output

A csv file with the DEseq output for all tissues, mapped to DO IDs, uniprot accession ids, and refseq ids. The path is specified in the script as one of the hard-coded lines edited during the method.

Other documentation

3.5 Major Changes from v-4.0

Major updates to the BioXpress from the previous version (v-4.0)

3.5.1 Tumor samples added for each tissue

| Tissue | TCGA Studies | New Samples |
|---------------|----------------|----------------|
| Bladder | BLCA | 126 |
| Breast | BRCA | 159 |
| Colorectal | COAD/READ | 159 (141/18) |
| Esophageal | ESCA | 25 |
| Head and Neck | HNSC | 118 |
| Kidney | KICH/KIRP/KIRC | 289(15/82/192) |
| Liver | LIHC | 169 |
| Lung | LUAD/LUSC | 264 (174/90) |
| Prostate | PRAD | 116 |
| Stomach | STAD | 22 |
| Thyroid | THCA | 176 |
| Uterine | UCEC | 216 |

3.5.2 Mapping files updated to reflect most recent mapping of DOIDs to UBERON IDs.

The following is a list of the current cancer tissue (DOID) to healthy tissue (UBERON ID) mapping:

| DO Name (DOID) | UBERON Name (UBERON ID) |
|-----------------------------------|---|
| Stomach Cancer (DOID:10534) | Stomach (UBERON:0000945) |
| Thyroid Cancer (DOID:1781) | Thyroid Gland (UBERON:0002046) |
| Esophageal Cancer (DOID:5041) | Esophagus (UBERON:0001043) |
| Kidney Cancer (DOID:263) | Adult Mammalian Kidney (UBERON:0000082) |
| Lung Cancer (DOID:1324) | Lung (UBERON:0002048) |
| Uterine Cancer (DOID:363) | Uterine Cervix (UBERON:0000002) |
| Bladder Cancer (DOID:11054) | Urinary Bladder (UBERON:0001255) |
| Prostate Cancer (DOID:10283) | Prostate Gland (UBERON:0002367) |
| Colorectal Cancer (DOID:9256) | Colon (UBERON:0001155) Rectum (UBERON:0001052) |
| Liver Cancer (DOID:3571) | Liver (UBERON:0002107) |
| Breast Cancer (DOID:1612) | Thoracic Mammary Gland (UBERON:0005200) |
| Head and Neck Cancer (DOID:11934) | Oral Cavity (UBERON:0000167) |

3.5.3 Automatic alphabetical re-ordering of count matrices for DESeq2

Due to the added samples in v-5.0, the ordering of samples in the count matrices needed for DESeq2 was disrupted and DESeq2 was producing randomized results. Column and row names in count matrices are now re-ordered as part of the *DESeq.R* script, so that samples are aligned correctly. This re-ordering should account for instances of added samples in future versions.

3.5.4 Issue Running DESeq per case

The step for DESeq per case was performed, however the results were not used to calculate subjects up/down/total in the publisher step, as was the case in v-4.0. Also, a final publisher file per case was not generated.

The `run_per_case.py` script performs DESeq analysis using both the tumor and normal count files per case. For most cases, there is only one tumor counts file and one normal counts file. DESeq encounters an error when running analysis with a sample size of 1 per group:

```
Error in checkForExperimentalReplicates(object, modelMatrix):`
```

The design matrix has the same number of samples and coefficients to fit, so estimation of dispersion is not possible. Treating samples as replicates was deprecated in v1.20 and no longer supported since v1.22.

The DESeq2 vignette also mentions DESeq analysis with no replicates in their [FAQ](#):

Can I use DESeq2 to analyze a dataset without replicates? No. This analysis is not possible in DESeq2.

This is likely due to the read count normalization model used by DESeq. DESeq's model contains a variable called the dispersion estimate, which relies on the variance of the one sample's read counts for a gene to the mean read count for that gene across the whole group (condition). If there are no other replicates on the group then there is no comparison to be made and no normalization can occur.

Even for cases that have only 2-3 replicates, the significance of the DE analysis should be heavily scrutinized as such a low replicate number is not a standard statistical practice, because low sample sizes may lead to an increase in false positive and false negatives.

3.6 Post-processing for OncoMX and Glygen

Processing done for integration of BioXpress data into OncoMX and Glygen.

3.6.1 Processing for OncoMX

The final output from BioXpress v-5.0 is available on the OncoMX-tst server at the path: `/software/pipeline/integrator/downloads/bioexpress/v-5.0/``

For OncoMX, the `de_per_tissue.csv` is used to report gene expression per tissue, however `data.oncomx.org` hosts both per tissue and per study datasets. The files are processed with the recipe pipeline. The recipes filter for all genes that are successfully mapped to uniprotkb accession IDs.

Recipes

`human_cancer_mRNA_expression_per_study.json` `human_cancer_mRNA_expression_per_tissue.json`

The output is available on the OncoMX-tst server at the path: `/software/pipeline/integrator/unreviewed`

Final output files

`human_cancer_mRNA_expression_per_study.csv` `human_cancer_mRNA_expression_per_tissue.csv`

3.6.2 Processing for Glygen

The final output from BioXpress v-5.0 was modified to align with the previous input for cancer gene expression, and now includes the following columns:

- `pmid`
- `sample_name`
- same as `DOID` and `name`
- `parent_doid`
- same as `DOID`
- All `DOIDs` in v-5.0 are parent terms
- `parent_doname`
- same as `DOID` and `name`
- All `DOIDs` in v-5.0 are parent terms
- `sample_id`
- Taken from previous version, unclear on the origin of these numbers

The following mapping for the column `sample_id` was recovered from the previous version and mapped to `DOIDs` present in v-5.0

| sample_name | sample_id |
|--|-----------|
| DOID:10283 / Prostate cancer [PCa] | 42 |
| DOID:10534 / Stomach cancer [Stoca] | 19 |
| DOID:11054 / Urinary bladder cancer [UBC] | 34 |
| DOID:11934 / Head and neck cancer [H&NC] | 46 |
| DOID:1612 / Breast cancer [BRCA] | 70 |
| DOID:1781 / Thyroid cancer [Thyca] | 16 |
| DOID:234 / Colon adenocarcinoma | 3 |
| DOID:263 / Kidney cancer [Kidca] & Kidney renal cl ... | 61 |
| DOID:3571 / Liver cancer [Livca] | 60 |
| DOID:3907 / Lung squamous cell carcinoma | 33 |
| DOID:3910 / Lung adenocarcinoma | 53 |
| DOID:4465 / Papillary renal cell carcinoma | 57 |
| DOID:4471 / Chromophobe adenocarcinoma | 23 |
| DOID:5041 / Esophageal cancer [EC] | 32 |

The processed file for Glygen is available on the glygen-vm-dev server at `/software/pipeline/integrator/downloads/bioxpress/August_2021/human_cancer_mRNA_expression_per_tissue_glygen.csv`