LEEDS OMICS
BUILDING RESEARCH PARTNERSHIPS AT THE UNIVERSITY OF LEEDS

# *Ad-hoc* method to download and prepare TCGA data for DE analysis

By Elton Vasconcelos (Aug/2021)

1. Navigate through https://portal.gdc.cancer.gov/repository in order to find your target cancer and data types. For this particular tutorial, you must select "Gene Expression Quantification" Data Type on your left-hand side panel. Target cancer type can be selected through the "Primary Site" pie chart.
2. Once satisfied with your selections, do the following:
    a. Click on "Add All Files to Chart" and then on the "Manifest" button to download a manifest plain text file that will be further used by the gdc-client tool (step 4.c below). The manifest file should be called something like "gdc_manifest_YYYYMMDD_fewMoreDigits.txt"
    b. Click on the "Cart" button on the top-right corner of the webpage. Once within the cart page, click on the "Sample Sheet" button to download a metadata table. The metadata file should be called something like "gdc_sample_sheet.YYYY-MM-DD.tsv".
3. Download gdc-client tool at https://gdc.cancer.gov/access-data/gdc-data-transfer-tool . In this tutorial, we will be using the Linux-Ubuntu version of that software, which comes already compiled after unzipping, and placed within an automatically generated "GDCdttc" directory.

```
$ unzip gdc-client_v1.6.1_Ubuntu_x64.zip
```

4. Download your gene expression data according to the following:
    a. First create a work directory for such task and go within it:

```
$ mkdir TCGA-yourTargetCancer
$ cd TCGA-yourTargetCancer
```

    b. Then move both manifest and metadata files (downloaded on item 2 above) to your current work directory:

```
$ mv /path/to/your/gdc_manifest_YYYYMMDD_fewMoreDigits.txt .
$ mv /path/to/your/gdc_sample_sheet.YYYY-MM-DD.tsv .
```

    c. Now download all your TCGA target dataset with the gdc-client tool:

```
$ nohup /path/to/your/GDCdttc/gdc-client download -m
gdc_manifest_YYYYMMDD_fewMoreDigits.txt &
```

#NOTE: Keep regularly checking the nohup.out file in order to see whether files are being properly downloaded (suggestion: $ tail -f nohup.out). Once it's all done, nohup.out must show "Successfully downloaded" on its last line.

5. Data prep:

a. While gdc-client does its job, you may already want to start some data prep regarding setting both normal and tumour sample IDs:

```
$ grep -P -o 'TCGA[\w\-]+11A' gdc_sample_sheet.YYYY-MM-DD.tsv | sort -u >normal.ids
$ grep -P -o 'TCGA[\w\-]+01[AB]' gdc_sample_sheet.YYYY-MM-DD.tsv | sort -u >tumour.ids
```

#NOTE-1: In case you are interested on paired samples (tumours and their respective adjacent normal tissue), do the following rather than the last command above:

```
$ sed 's/11A$//g' normal.ids | xargs -i grep -P '{}.*Tumor' gdc_sample_sheet.YYYY-MM-DD.tsv | cut -f 7 | sort -u >tumour-paired.ids
```

#NOTE-2: Here we are dealing with 11A (Solid Tissue Normal) and 01A or 01B (Primary Solid Tumour) TCGA codes, only. In case you are handling other TCGA sample codes, please replace the grep regexp according to what you want to extract from the metadata.

b. Creating *ad-hoc* Bash scripts to associate samples to htseq.counts gene expression files:

```
$ for i in `cat normal.ids`; do grep -P "htseq\.counts.*Gene Expression.*\t$i\t" gdc_sample_sheet.YYYY-MM-DD.tsv  | cut -f 1,2 | sed 's/\t/\//g'  | sed -r "s/^./echo $i >x\\nzcat &/g"| sed -r 's/\/.*$/& \| cut -f 2 >y\ncat x y >&2/g' | sed 's/>\//>/g' | sed 's/gz2$/txt2/g' ; done >genes-normal-counts.bsh

$ for i in `cat tumour.ids`; do grep -P "htseq\.counts.*Gene Expression.*\t$i\t" gdc_sample_sheet.YYYY-MM-DD.tsv  | cut -f 1,2 | sed 's/\t/\//g'  | sed -r "s/^./echo $i >x\\nzcat &/g"| sed -r 's/\/.*$/& \| cut -f 2 >y\ncat x y >&2/g' | sed 's/>\//>/g' | sed 's/gz2$/txt2/g' ; done >genes-tumour-counts.bsh
```

c. Running both Bash scripts (gdc-client download from step 4.c above must be finished for the proper execution of the *ad-hoc* scripts). But first create some subdirectories for organisational purposes:

```
$ mkdir DE && mkdir DE/normal DE/tumour
$ bash genes-normal-counts.bsh
$ mv *txt2 DE/normal/
$ bash genes-tumour-counts.bsh
$ mv *txt2 DE/tumour/
```

d. Preparing a readCounts table to be used as input on any DE tool of choice:

# d.1. Enter DE dir:

```
$ cd DE/
```

# d.2. Take a look at the two first files from the genes-normal-counts script:

```
$ head -2 ../genes-normal-counts.bsh
```

# d.3. Copy the htseq file name displayed on the second line from the command above and paste it in the command below as advised:

```
 $ zcat ../dir_code_from_the_htseq_file_retrieved_by_head_-
2_above/file_code_from_htseq.counts.gz | cut -f 1 >IDs-col.txt
```

# d.4. Use nano to edit IDs-col.txt by adding a "geneID" header on the top of the genes' list

```
$ nano IDs-col.txt
```

# d.5. Count the number of lines of all files that will be combined into a single table.

# ATTENTION: They must all have the same number of lines (or number of genes)

```
$ wc -l IDs-col.txt normal/* tumour/*
```

# d.6. Combining geneIDs, normal-samples, and tumour-samples readCounts

```
$ paste IDs-col.txt normal/* tumour/* >geneCounts-input4DE.tsv
```

e. Use "geneCounts-input4DE.tsv" as input to your favourite DE stats tool, and don't forget to correct for batch effects (ComBat-seq from the SVA Bioconductor package is recommended).

## Retrieving normalized expression values (FPKM) rather than raw read counts

In case one does not wish to run a downstream DE analysis on his/her own, and would rather take a look at normalized expression values for a small set of genes, the pipeline above can be fully adapted for such purpose:

➔ Slightly edit step 5.b commands like the following:

```
$ for i in `cat normal.ids`; do grep -P "FPKM.*Gene
Expression.*\t$i\t" gdc_sample_sheet.YYYY-MM-DD.tsv  | cut -f 1,2 |
sed 's/\t/\//g'  | sed -r "s/^./echo $i >x\\nzcat &/g"| sed -r
's/\/.*$/& \| cut -f 2 >y\ncat x y >&2/g' | sed 's/>\//>/g' | sed
's/gz2$/txt2/g' ; done >genes-normal-FPKM.bsh

$ for i in `cat tumour.ids`; do grep -P "FPKM.*Gene
Expression.*\t$i\t" gdc_sample_sheet.YYYY-MM-DD.tsv  | cut -f 1,2 |
sed 's/\t/\//g'  | sed -r "s/^./echo $i >x\\nzcat &/g"| sed -r
```

```
's/\/.*$/& \| cut -f 2 >y\ncat x y >&2/g' | sed 's/>\//>/g' | sed
's/gz2$/txt2/g' ; done >genes-tumour-FPKM.bsh
```

➔ Then continue downstream in the pipeline replacing the word “counts” by “FPKM”.