[ ]:
```
%load_ext autoreload
%autoreload 2
```

Descarga de Ficheros

En este paso se descargan los ficheros de imágenes histológicas y de expresión genénica a partir de la información obtenida en el apartado anterior.

Packages

[61]:
```
from gdc.download import api_download_iterative
from gdc.utils import gunzip
```

[62]:
```
import requests
import yaml
import json
import os
import re
import pandas as pd
```

Config

[63]:
```
with open('conf/user_conf.yaml', 'r') as f:
    conf = yaml.load(f)
```

Paths

[64]:
```
slides_metadata_file = os.path.join(conf['data_path'], 'slides_metadata.csv')
rnaseq_metadata_file = os.path.join(conf['data_path'], 'rnaseq_metadata.csv')
```

[65]:
```
slides_path = os.path.join(conf['data_path'], 'slides', 'svs')
if not os.path.exists(slides_path):
    os.mkdir(slides_path)
```

[74]:
```
rnaseq_path = os.path.join(conf['data_path'], 'rnaseq')

if not os.path.exists(rnaseq_path):
    os.mkdir(rnaseq_path)
```

Lectura de Metadatos

Obtiene el *id* de los ficheros a descargar a partir de los metadatos descargados anteriormente.

[75]:
```
slides_df = pd.read_csv(slides_metadata_file, sep='|')
slides_df = slides_df[['file_name', 'file_id', 'file_size', 'sample_id',␣
 ↪'experimental_strategy']]
slides_df.head(3)
```

[75]:

	file_name	file_id	\
0	TCGA-2J-AABR-01A-01-TS1.svs	e1daa3de-3f76-44fb-ba6b-f60af4943ef3	
1	TCGA-US-A77G-11A-01-TSA.svs	dfa7125a-e250-4abf-a528-608dded99751	
2	TCGA-S4-A8RM-01A-01-TSA.svs	1b62a0d6-56d1-4629-af6c-b7d1255a42cc	

```
   file_size         sample_id experimental_strategy
0     381.90  TCGA-2J-AABR-01A          Tissue Slide
1      24.74  TCGA-US-A77G-11A          Tissue Slide
2     187.91  TCGA-S4-A8RM-01A          Tissue Slide
```

[76]:
```
rnaseq_df = pd.read_csv(rnaseq_metadata_file, sep='|')
rnaseq_df = rnaseq_df[['file_name', 'file_id', 'file_size', 'sample_id',␣
 ↪'workflow_type']]
rnaseq_df.head(3)
```

[76]:
```
                               file_name  \
0  TCGA-YY-A8LH-01A_HTSeq-FPKM-UQ.txt.gz
1   TCGA-H6-A45N-01A_HTSeq-Counts.txt.gz
2   TCGA-RB-AA9M-01A_HTSeq-Counts.txt.gz

                                file_id  file_size         sample_id  \
0  4ac5c2da-497f-4fb4-80db-c7e774c1873a       0.53  TCGA-YY-A8LH-01A
1  78bb8d49-54aa-43a1-aec4-31da818cdb14       0.25  TCGA-H6-A45N-01A
2  82d7d3b5-85bc-46b9-b9cd-2bdeb279dc0f       0.26  TCGA-RB-AA9M-01A

     workflow_type
0  HTSeq - FPKM-UQ
1   HTSeq - Counts
2   HTSeq - Counts
```

Slides

Las imágenes histológicas están en formato SVS de alta resolución, por tanto la descarga será lenta y ocuparán mucho en disco.

[77]:
```
summary = slides_df.groupby('experimental_strategy').agg({'file_name': 'size',␣
 ↪'file_size': 'sum'})
summary = summary.rename(columns={'file_name': 'count', 'file_size': 'total_size␣
 ↪(gb)'})
summary['total_size (gb)'] = round(summary['total_size (gb)'] / 1000, 2)

summary
```

[77]:
```
                      count  total_size (gb)
experimental_strategy
Tissue Slide            257            57.48
```

Dependiendo del estudio que se desee realizar se puede que únicamente queramos utilizar aquellas imágenes que tengan su correspondiente información genénica. En ese caso se filtra el DataFrame de láminas haciendo que la muestra esté también en el DataFrame de RNA-SEQ.

[78]:
```
rna_seq_samples = rnaseq_df['sample_id'].unique()
slides_df = slides_df[slides_df['sample_id'].isin(rna_seq_samples)]
```

[79]:
```
summary = slides_df.groupby('experimental_strategy').agg({'file_name': 'size',␣
↪'file_size': 'sum'})
summary = summary.rename(columns={'file_name': 'count', 'file_size': 'total_size␣
↪(gb)'})
summary['total_size (gb)'] = round(summary['total_size (gb)'] / 1000, 2)

summary
```

[79]:

experimental_strategy	count	total_size (gb)
Tissue Slide	209	51.8

La función *api_download_iterative* descarga todas las imágenes en la ruta indicada, con el parámetro *multiprocess* es posible realizar descargas en paralelo, se recomienda utiliar en número de cores del procesador como número de descargas en paralelo. Se incluyen barras de progreso para ver el estado de las descargas.

[ ]:
```
api_download_iterative(slides_df, slides_path, multiprocess=8)
```

RNA-Seq

Los ficheros de RNA-Seq al ser de texto plano ocuparán mucho menos y la descarga es rápida.

[53]:
```
summary = rnaseq_df.groupby('workflow_type').agg({'file_name': 'size', 'file_size':␣
↪'sum'})
summary = summary.rename(columns={'file_name': 'count', 'file_size': 'total_size␣
↪(gb)'})
summary['total_size (gb)'] = round(summary['total_size (gb)'] / 1000, 2)

summary
```

[53]:

workflow_type	count	total_size (gb)
HTSeq - Counts	182	0.05
HTSeq - FPKM	182	0.09
HTSeq - FPKM-UQ	182	0.09

De nuevo se pueden filtrar aquellas muestras que estén pareadas con datos histológicos, en ese caso son todas.

[54]:
```
slides_samples = slides_df['sample_id'].unique()
rnaseq_df = rnaseq_df[rnaseq_df['sample_id'].isin(slides_samples)]
```

[55]:
```
summary = rnaseq_df.groupby('workflow_type').agg({'file_name': 'size', 'file_size':␣
↪'sum'})
summary = summary.rename(columns={'file_name': 'count', 'file_size': 'total_size␣
↪(gb)'})
summary['total_size (gb)'] = round(summary['total_size (gb)'] / 1000, 2)

summary
```

[55]:
```
                 count  total_size (gb)
workflow_type
HTSeq - Counts     182            0.05
HTSeq - FPKM       182            0.09
HTSeq - FPKM-UQ    182            0.09
```

[ ]:
```
files = api_download_iterative(rnaseq_df, rnaseq_path, multiprocess=8)
```

Puesto que las imágenes viene comprimidas en formato *GNU ZIP* será necesario descomprimirlas. Este paso itera sobre todas los ficheros en la ruta de descarga y descomprime aquellos con extensión *.gz*.

[59]:
```
for file_name in files:

    source_filepath = os.path.join(rnaseq_path, file_name)
    dest_filepath = re.sub('\.gz$', '', source_filepath)

    gunzip(source_filepath, dest_filepath)
    os.remove(source_filepath)
```