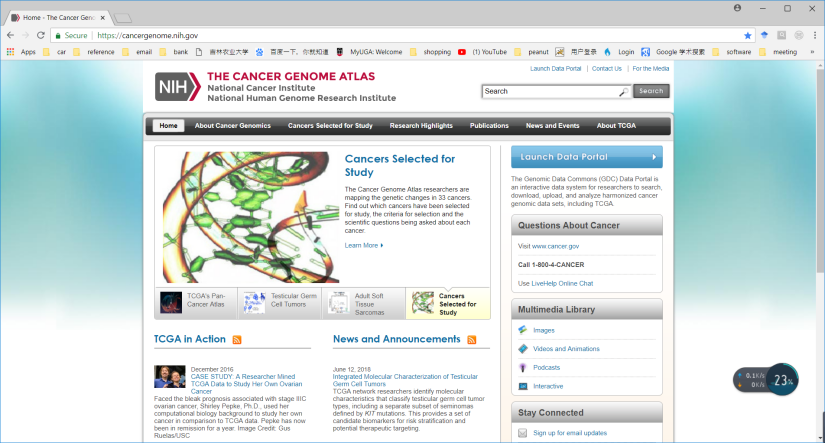
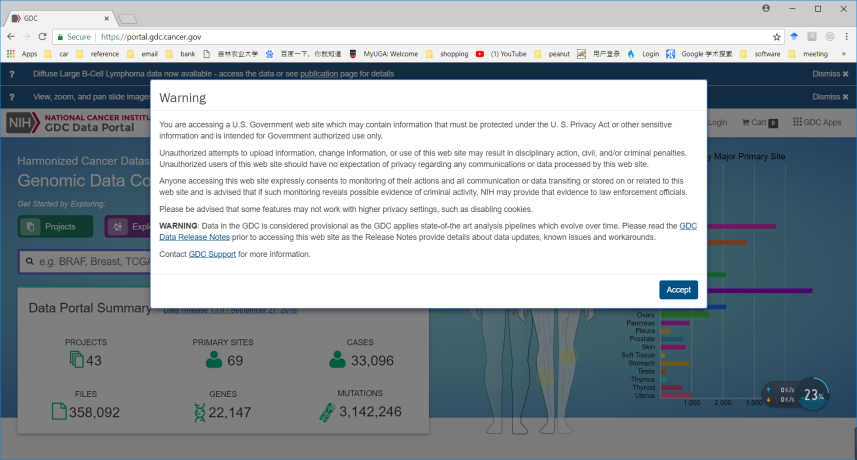
## 2018年10月4日

**如何从TCGA下载数据，并用GIT查看数据-20181004**

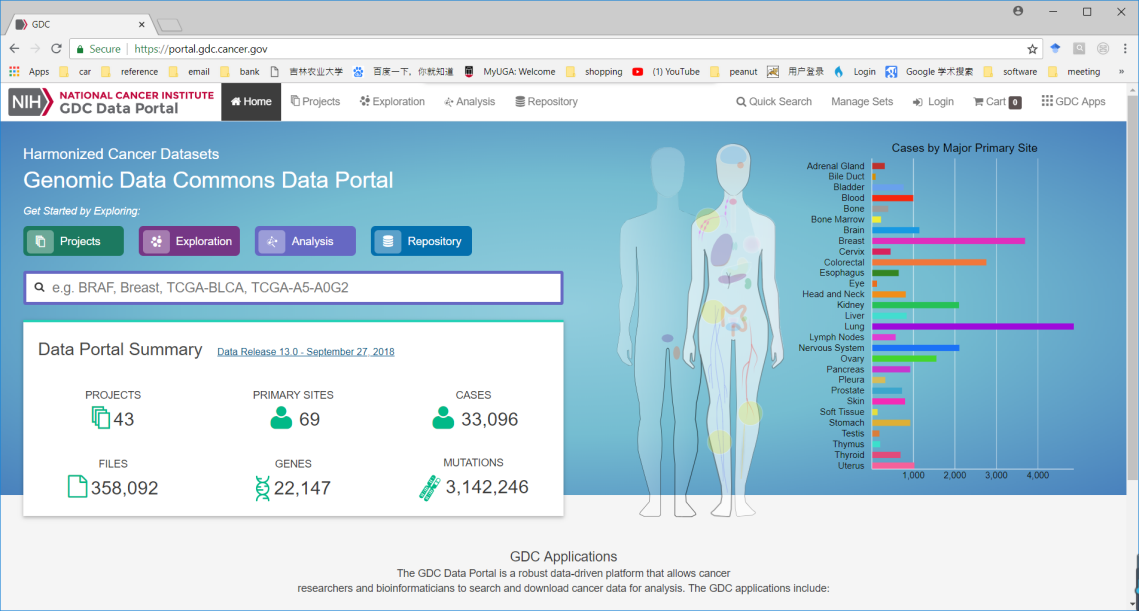
1. 登陆TCGA网址（Google直接搜索TCGA 网址https://cancergenome.nih.gov/），Home页面下，点击“Launch Data Portal”.



在Warning提示框中，直接选择：Accept。



2.选择“Repository(资源库)”.



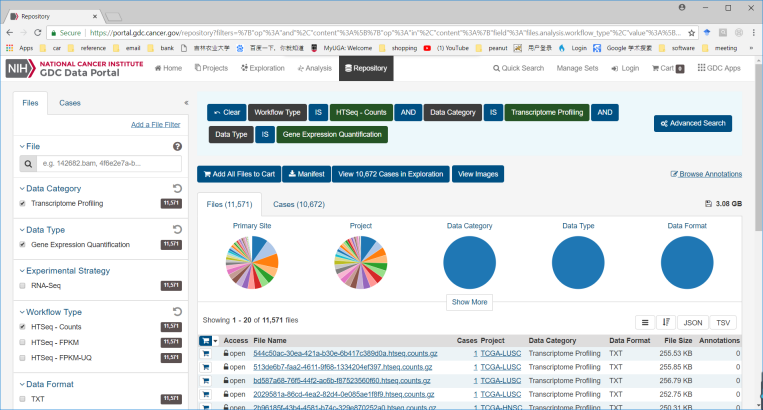
3左侧对筛选参数进行设置,如红色框标注所示，在File中选择Project的饼图中的病例：

Data category: （No.2）Transcriptome Profiling

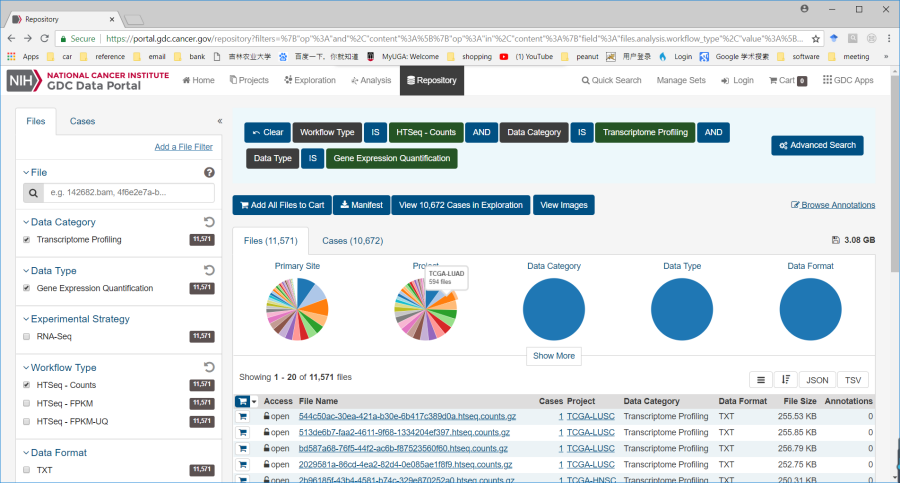
Data Type: Gene Expression Quan

Experimental Strategy: RNA-seq

Workfolw Type: HTseq-Counts

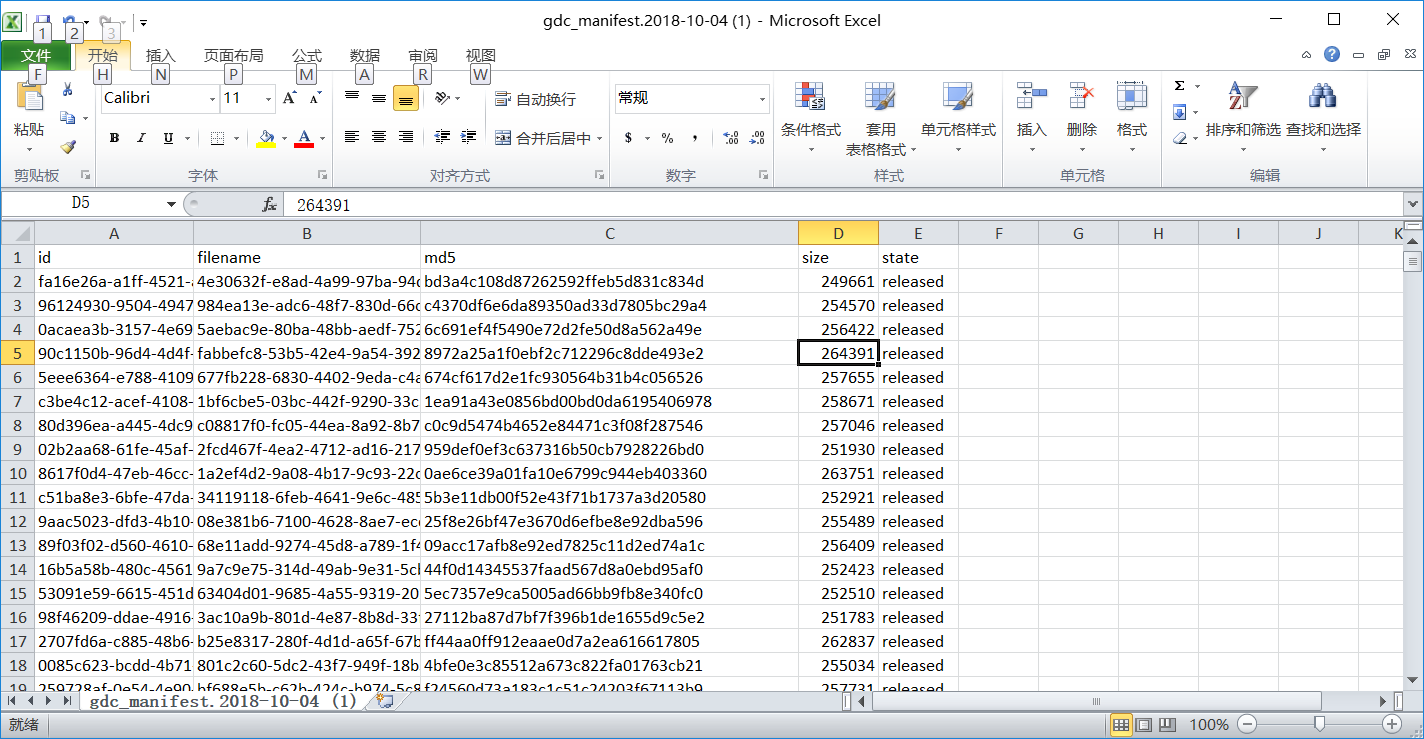


1. 在Project（饼图）中选择TCGA转录数据的病原类型，如“TCGA-LUSC”，点击“Manifest(出货单)”下载到电脑本地盘中，重命名为“gdc\_manifest.2018-10-08-LUSC”。这样，我们就从TCGA数据库下载转录水平的数据到自己的电脑中。



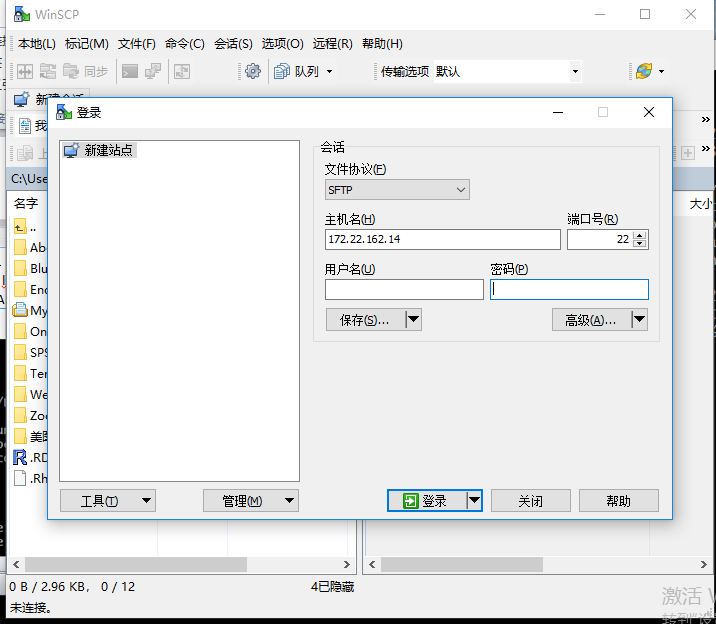
下载

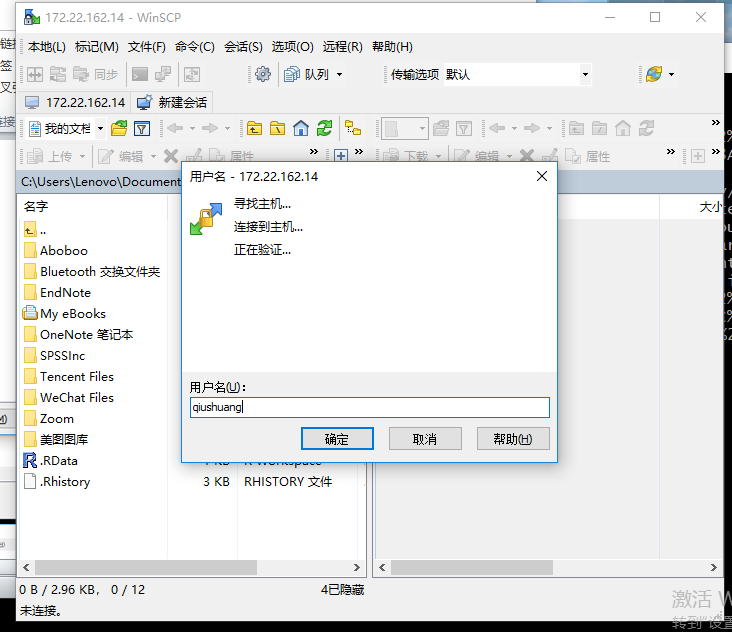
第1列文件id，第2列文件名，第3列md5，第4列文件大小，第5列状态：发布或者未发布。

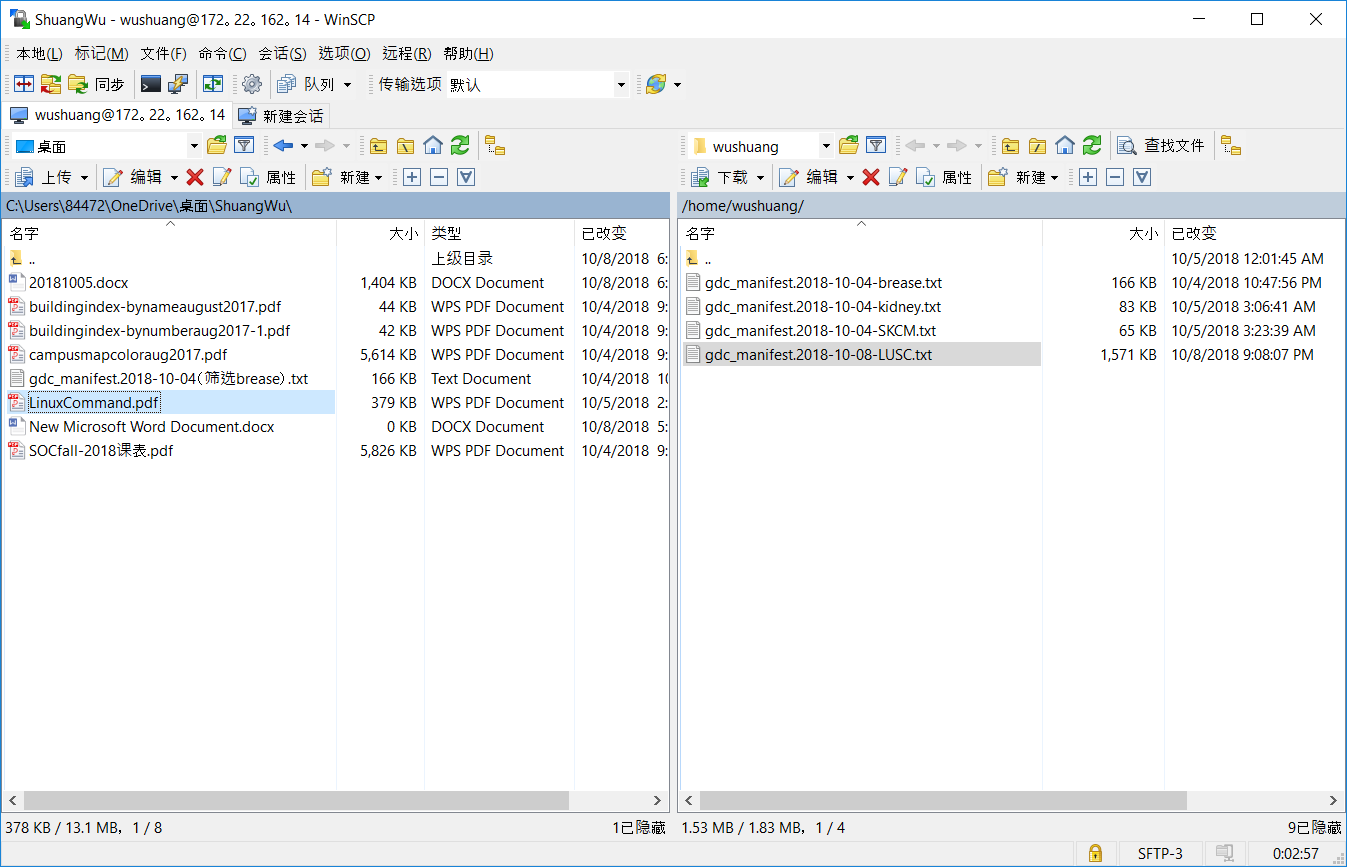


接下来，我们将自己下载的数据文件（文本文件.txt）利用工具“WinSCP（SFTP客户端，下载安装到本机）”将数据上传到服务器。在Windows系统下利用Git Bash（版本控制，下载安装到本机）运行Linux命令来查看下载的数据文件。

左侧：本机文件夹 右侧：服务器文件夹



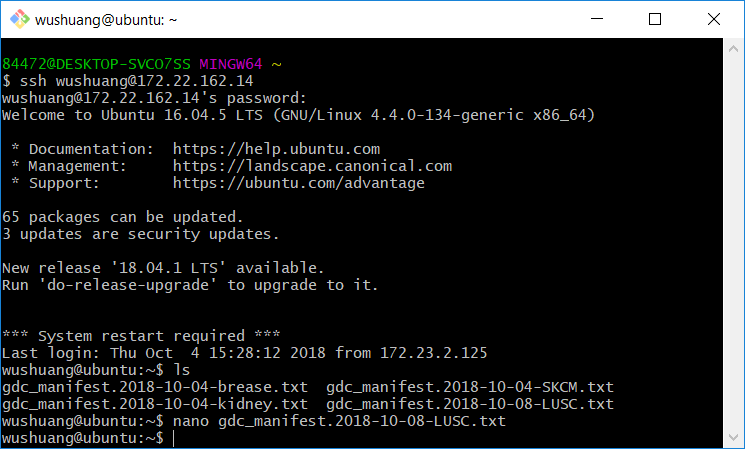




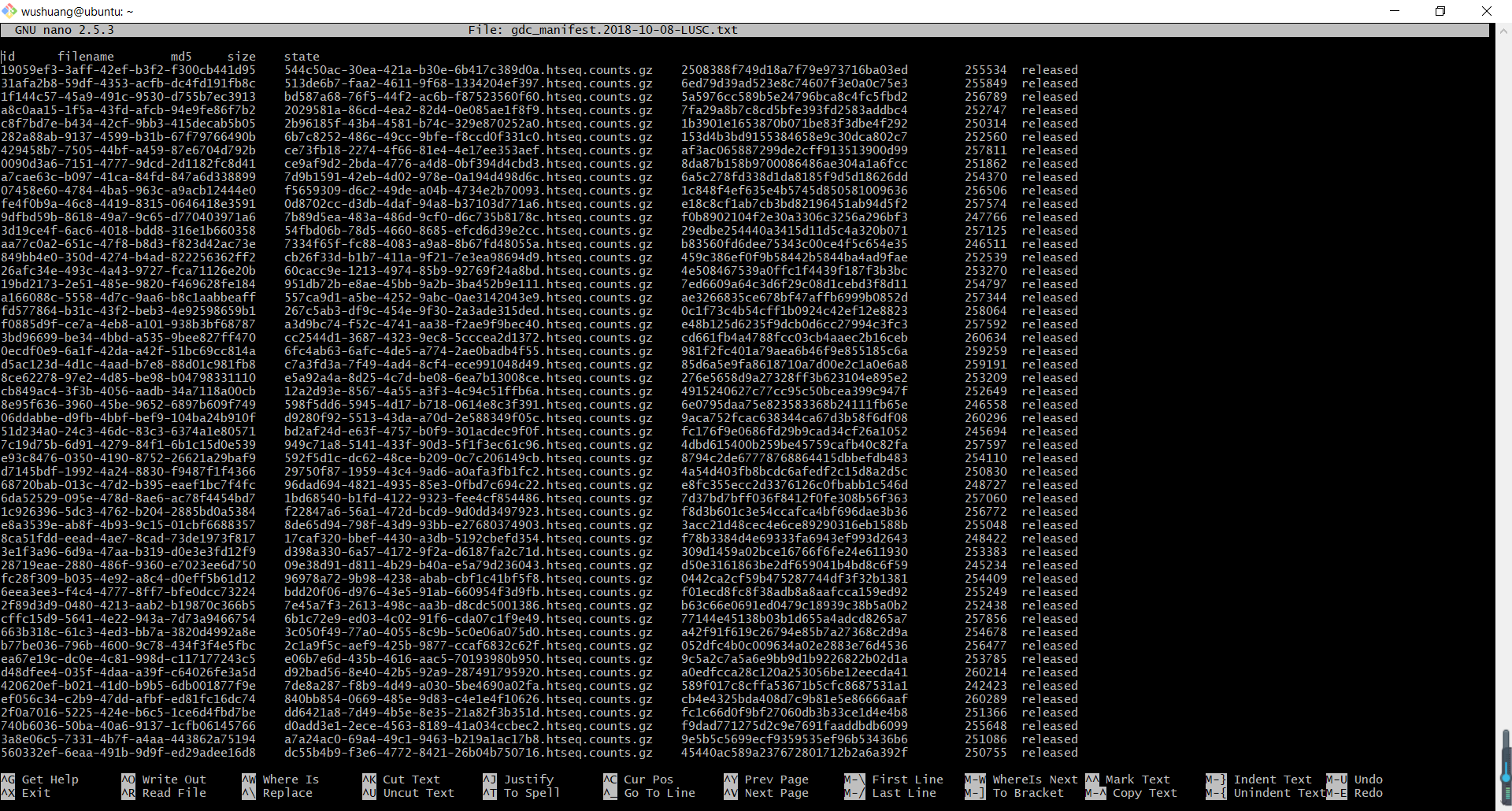
服务器文件夹

本地文件夹

打开Git Bash，在系统提示符$ 输入 ssh 用户名@服务器地址 并回车，在password后输入自己的密码，在提示输入命令符$ 输入 ls，显示所有上传文件。输入 nano 命令+文件名，选择自己要观察的某一项疾病数据，从TCGA下载的数据通过Git观察到的结果如下。



数据链接结果显示：



得到结果后，输入快捷键control x可退出查看，跳上面一条

## 2018年10月8日星期一

注意：linux里面大小写是不一样的；红色的是zip文件；白色是文本文件；绿色是可执行的文件；

1.路径问题

分相对绝对和相对路径

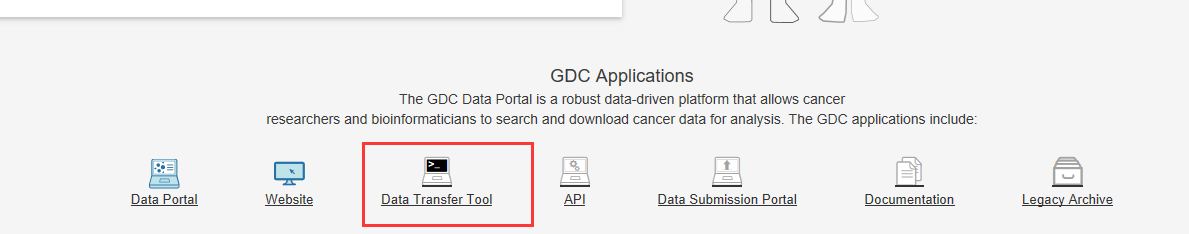
绝对路径，下划线这种的叫////，第一个“/”指的是根目录

没有/的叫相对路径，从当前目录开始算

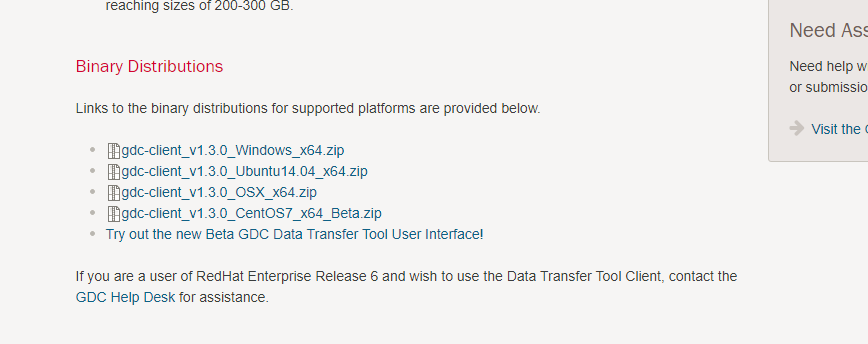
2.找到网站上的文件的网址并下载到服务器中

①找网址

TCGA上去主页后选择launch data portal后拽到底下有



下面点进去后是，这几个是我们以后下载数据要用到的工具



Ubuntu是我们实验室服务器的系统

一般选择Ubuntu或者Linux

鼠标放在文件附近右键复制链接地址

②下载到服务器

到git操作界面粘贴网址

wget <http://....从网站上下载数据到服务器> wget或者用curl 同等功能

wget <https://gdc.cancer.gov/system/files/authenticated%20user/0/gdc-client_v1.3.0_Ubuntu14.04_x64.zip>

3.一些命令

①cd是change director改文件地址

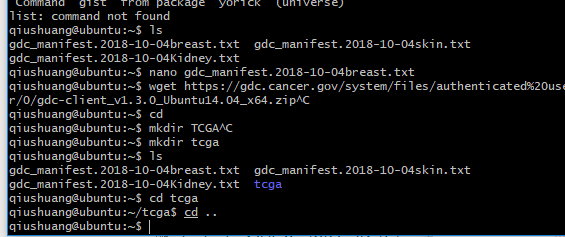
输入cd相当于回到/home/qiushuang文件夹

②mkdir tcga （make directory）相当于建立新的命名为tcga的文件夹（蓝色标识）

③cd tcga 进入tcga文件夹

④cd .. 回到上一级目录一个点“.”是当前目录，”..”是回到上一级目录

⑤cd ../..是回到上上一级目录



⑥pwd（working directory）显示当前绝对路径(不确定自己在哪的时候查一下自己在哪)

⑦删除的一些命令

rmdir删除文件夹（只有文件夹是空的时候才能删掉，不是空的会报错）

rm是删除（要慎用，删了就没了）

rm –rf 文件名 （删除文件夹的）

⑧查命令的帮助

输入

command名 -–help 或者 Command名 –h

⑨列出我们目录下的文件

ls –trhl 指的是ls –t –r –h –l一起执行 把当前文件按照时间反向排序列出来(最下面是最新的文件)

ls –rshl 最近的在最上面

ll 是ls –l的意思

-t 是按修改时间排列(时间倒叙排列，最早的在最上面)

–r是时间反序

-tr最新的文件在最下面

-l（long）显示更多信息

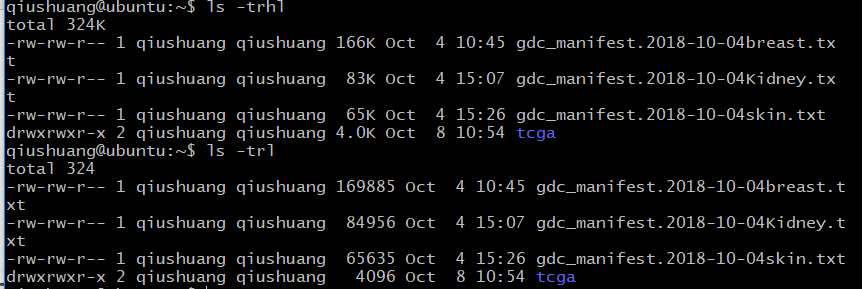
见图 第一个“-”表示他就是一个文件 d是文件夹；r=read w=write x=excute是执行

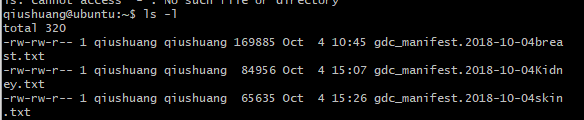
前三位(rw-)表示用户对他的权限，如rw- (意为有读和写的权限)

中三位(rw-)是group对他的权限

后三位(r--)是其他人对他的权限,意思是只能读，不能写和执行

–h human readable （把文件的大小以kb的形式列出来，和l一起用，不单独用） trhl 列出的文件大小不太准





⑩快捷键

Control l 清屏

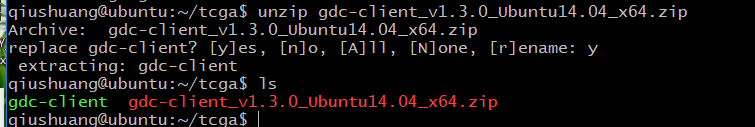
Control c 强制停止；

Control d 退出

Control x 回到上一个命令

Control shift V是粘贴

⑾unzip 文件名 解压



\*\*绿色的文件是可执行的文件

常见的文件是下面这三种，左侧是文件类型，右侧是解压的命令

zip unzip

tar.gz tar –xzvf

gz gunzip (会把原始压缩包删掉)

⑿文件移动和复制

cp ~/文件名 ~/tcga/文件新名字

将一个文件从绝对路径home里复制到tcga文件夹中并赋予新的名字

~指的是我自己的home中,等于 /home/qiushuang/

若此时在tcga文件夹中

cp ../文件名 ./文件新名字

将上一级目录中的文件名复制到tcga中

若此时在home中

cp 文件名 ./tcga/

cp -r 是复制文件夹及文件夹中的内容的，输入cp –r BBB AAA

mv 是移动和改名

将home下的BBB文件夹移动到AAA中，输入mv BBB AAA (注意和cp的区别，不用加-r)

⒀cat/head/tail 文件名 （这几个命令是来看文件内容的）

假设文件一共1000000行

cat 不会改文件，会把都显示在屏幕中，要慎用

先看大概多大，如果很小，那就直接打开看看就好，但如果文件太大，看文件内容就容易卡斯

head 只看是头几行 head –n 2只看前两行 head –n -1 是看除掉最后一行的所有行

tail 只看后几行

less 看的少一点，是打一屏幕，后面有冒号，按上下会给翻页，按q（quit）就能退出来

head和tail都可以这样用，设置看几行。如head -20 意思是看前20行

⒁wc –l 文件名 查行数 (word count)

md5

## 2018年10月10日 星期三

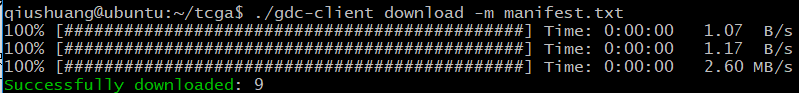
学习如何把下载的数据拼到一块

1.从周易文件夹中拷贝一个名为manifest.txt的文件



2.用gdc软件下载manifest里面的文件

qiushuang@ubuntu:~/tcga$ ./gdc-client download -m manifest.txt



./ gdc-client要执行此文件 让他下载manifest文件（TCGA里面的9个文件）

-m的意思可以去查gdc –help查

结果中一共有9个文件夹

3.\*代表任意长度的任意字符串

⑴ls \*.txt 只列举txt的文件

⑵ls \*/ 列举出所有的文件夹

要看正则表达式（regex （Regular Expression））里面的符号：

如.，+，t,\*，\,ln,lw,ls,{|d},^,$,[],[^]等

glob里面的星号

glob 是一种特殊的模式匹配，最常见的是通配符拓展，也可以将 glob 模式设为精简了的正则表达式

|  |  |
| --- | --- |
| \* | 匹配任意长度任意字符 |
| ? | 匹配任意单个字符 |
| [list] | 匹配指定范围内（list）任意单个字符，也可以是单个字符组成的集合 |
| [^list] | 匹配指定范围外的任意单个字符或字符集合 |
| [!list] | 同[^list] |
| {str1,str2,...} | 匹配 srt1 或者 srt2 或者更多字符串，也可以是集合 |

| **字符** | **作用** |
| --- | --- |
| IFS | 由 < space > 或 < tab > 或 < enter > 三者之一组成 |
| CR | 由 < enter > 产生 |
| = | 设定变量 |
| $ | 作变量或运算替换 |
| > | 重导向标准输出 |
| < | 重导向标准输入 |
| \ |  |
| & | 重导向文件描述符，或将命令静默执行 |
| ( ) | 将其内的命令置于 nested subshell 执行，或用于运算或命令替换 |
| { } | 将其内的命令置于 non-named function 中执行，或用在变量替换的界定范围 |
| ; | 在前一个命令结束时，而忽略其返回值，继续执行下一个命令 |
| && | 在前一个命令结束时，若返回值为 true，继续执行下一个命令 |
| \ | \ |
| ! | 执行 history 中的命令 |

周易manifest里面下载后9个文件一共分三类：counts.gz，counts，logs

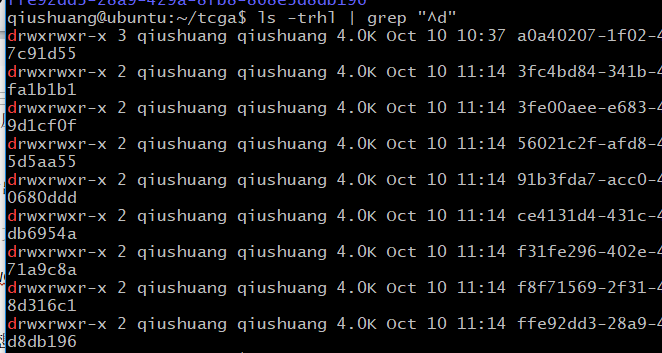
counts.gz是压缩文件，counts是counts.gz解压后的文件，logs是一个文件夹

4.grep

竖线（|）在Linux里很重要，表示A 的输出当作B的输入

^表示以谁谁谁开始

ls -trhl | grep "^d" 可列出所有文件夹(d代表文件夹)



想看目录有多少总的文件数，文件夹和文件均显示出来，使用如下命令

qiushuang@ubuntu:~/tcga$ ls | wc -l



5.循环（for语句）

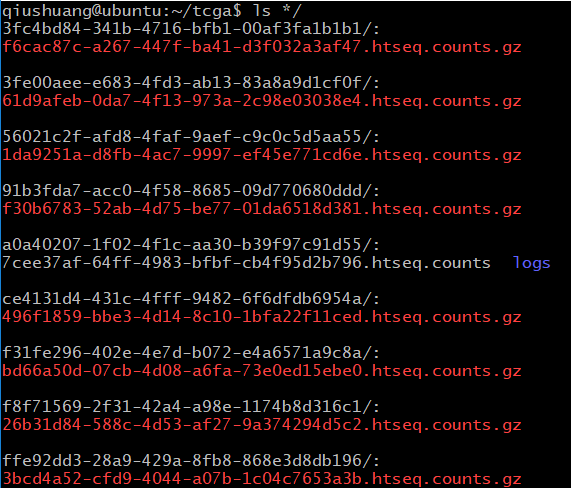
假设有一个X=【1，2，3，4，5…10000】想计算每一个的数加1

计算X+[1]

For i in x

print (i+1)

有9个文件夹，每个文件夹中都有counts。Gz.logs文件，如下：白色是文件夹的名字，红色是文件夹中的文件



下面的dir1 2 3d等是每一个文件夹的路径

dir1 /\*.counts.gz

dir2 /\*.logs

……

dir9 /\*.counts

想要只解压counts.gz的文件夹，则使用如下命令

这是Linux里面的循环语句

for file in \*/\*.gz

do

gunzip “$file”(这里可以输入其他目的的命令，比如head –n 1 “$file” 等)

done

回车和分号都是代表要进行新的命令

下面的命令是解压所有gz文件，（注意：file只是我们随便起的名字，可以换成别的名字）

for file in \*/\*.gz; do gunzip "$file"; done

for X in Condition

（要记得加上.gz因为如果不加上“.“，可能tgz这样的文件也会被选进来）

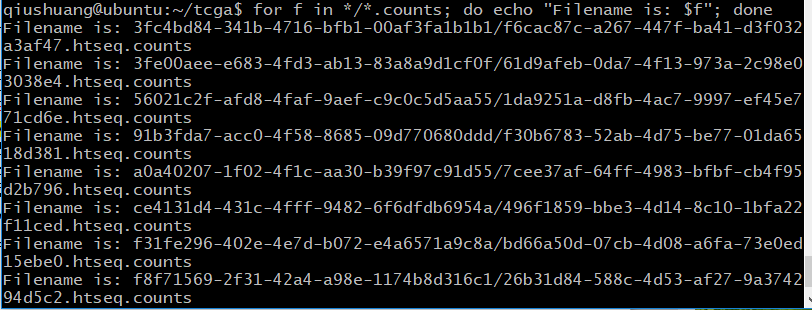
R和python里面的循环可能不一样但本质一样

$指定义

aaa=/home/qiushuang/manifest.txt

6.echo其实就是打印的意思

for f in \*/\*.counts; do echo "Filename is: $f"; done



f=~/tcga/manifest.txt（不能加引号）

如果加引号就要写全绝对路径，如下：f=“/home/qiushuang/tcga/manifest.txt”

echo "$f"

cat "$f"

二者的区别是echo只打印字符串，访问字符串本身，cat是打印文件内容

7.几个符号

⑴> 叫重定向符号，把所有输出的东西都放在文件里，不再放在屏幕里了（<和>这两个符号箭头指到哪里就是输出到哪里）

看如下例子：

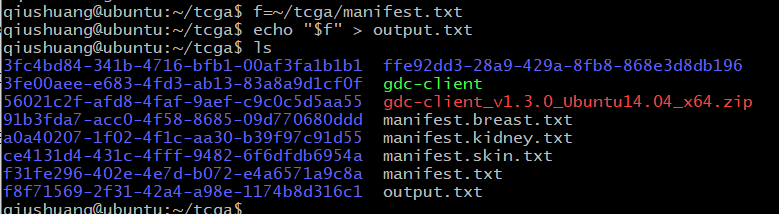
例子1：

f=vv

若输入echo “$f” 则输出vv，但如果输入echo “$f” > u.txt（随便起的一个txt文件），则把vv输出到u.txt



例子2（与例子1作用相同）

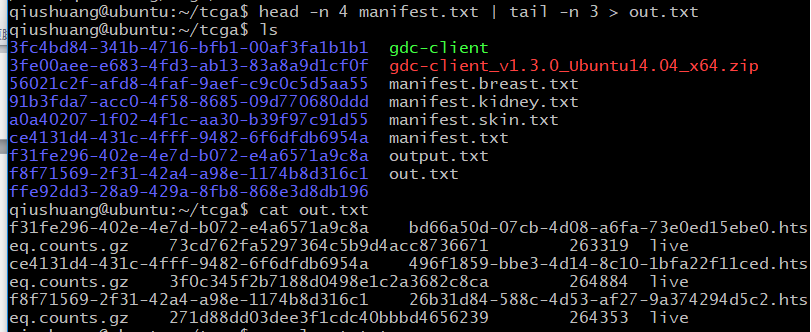


Output cat是这样的



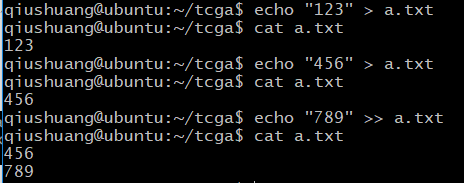
例子3：如果只想看一个文件的第二行到第四行,不看其他行，并存到out.txt中

首先使用head –n 4 manifest.txt的意思是挑出1-4行，然后使用“|“，意思是将1-4行的结果输出到后面，tail –n 3 manifest.txt的意思是看1-4行的后三行（即2-4行），> out.txt的意思是将2-4行存入到此文件中了。如下：



⑵想再要8，9，10行，怎么办？使用>>

>> 是追加（插入）文件，并不覆盖之前的文件，若还是用>符号后面的东西将覆盖掉之前的东西。



下面这个例子是将manifest文件中的后三行附加到刚刚仅有2-4行的out.txt中



假设一个文件(a.txt)一共20000行，想要将第20000行放到第一行，后面顺延第一行至第19999行

tail –n 1 a.txt > b.txt (将a文件中的第20000行放到b文件中的第一行)

head –n -1 a.txt >> b.txt (将a文件中的除最后一行的所有行都附加到刚刚的b文件中)

此时b文件中就是我们想要的

⑶&& 是与的意思

Command1 ….&& command2….

前一个命令执行成功才会执行后面的命令，若前面的没有执行成功则后面的也不会执行

Head –n -1 $f >

⑷||是或的意思

<