GTX Compressor 使用报告

人和未来生物科技有限公司

2016 年 10 月

摘要

GTX Compressor 是 Genetalks 公司 GTX Lab 实验室开发的面向大型数据（数 GB 甚至数 TB 数据，尤其是生物信息数据）上云，而量身定制的复杂通用数据压缩打包系统，可以对任意基因测序数据以及数据目录进行高压缩率的快速打包，形成单个压缩数据文件，以方便存储档与远程传输、校验。区别于以往的压缩工具，GT Compressor 系统着力于**高压缩率，高速率，方便的数据抽取**。

GTX Compressor 可以在 AWS C4.8xlarge 机器（或同配置服务器），**以超过 114MB/s 的速度，将接近 200GB 大小的 33 个质量数的 FASTQ 文件（NA12878_1.fastq），在 29 分钟内压缩到原大小的 13%**，而对于 X10 等只有 **7 个质量数的 FASTQ 数据，其压缩率更可以达到 5.5%。**

考虑商业使用时，用户会将大量 FASTQ 样本集中打包进一个压缩包中，因此，数据的随机抽取是一个重要的使用特性。GTX Compressor 的数据压缩引擎与存储格式从设计之初，就允许用户可以不用展开压缩包，即可随机抽取其中的任何一个文件的任何一个部分。GTX Compressor 不仅提供命令行抽取文件以及文件中内容，更提供基于 Python 的丰富 API 解包接口，允许用户使用程序自动地、灵活地枚举压缩包中的文件，并使用丰富方便的数据抽取接口，对压缩数据中任意文件，或是文件中任意部分进行解压，并可以将解压过程集成进用户自己的数据自动化处理流程中。

系统特点

该数据打包压缩系统的特点：

- **高压缩比：**采用 Context Model 压缩技术，配合多种优化的预测模型，平衡系统并发度与内存资源消耗后，能达到极高的压缩率。对 FASTQ 文件，压缩率最高可达 5.58%。
- **高性能：** GT Compressor 充分发挥了 CPU 的并发性以及计算能力。在普通 20 核服务器上，最高能够以接近 114MB/s 的流量输入数据并压缩完毕。
- **专通结合：**GT Compressor 即包含专门针对生物信息领域 FASTQ 文件的特殊压缩框架和技术；也同时支持对任意二进制文件或文本文件的高倍率压缩。
- **All in One：**目录、子目录、多个文件以及各种文件类型均可压缩打包成一个文件，该文件被称为多流文件，可以方便地进行传输、存储与校验。
- **增删随意：**将多流文件包视为一个压缩文件系统，可以在里面任意删除和添加文件。
- **直接抽取：**GT Compressor 的多流文件格式允许解压器或 Python 解压库可以任意抽取其中的某一个压缩文件，而不用完全解压所有包中文件后才能提取指定文件内容。此外，GT Compressor 中的压缩算法设计之初，既考虑了用户解压所需的“随机寻址”能力，工程实现上，通过开放 Python API 接口允许用户甚至不需要完整解开目标文件，即可以”随机寻址”方式对压缩包中的某个文件的某个部分进行读取。
- **接口丰富：** 提供基于 Python 的丰富 API 解包接口，允许用户使用程序自动地、灵活地枚举压缩包中的文件，并使用丰富方便的数据抽取接口，对压缩数据中任意文件，或是文件中任意部分进行解压，并可以将解压过程集成进用户自己的数据自动化处理流程中。

软件操作手册（beta 版）

2.1 命令行说明

执行 ./gtz -h，输出命令行帮助说明

```
USAGE:
    ./gtz  [-t] [-n <string>] [-l <string>] [-i] [-d] [-a] [-b] [-g
        <number>] [-o <string>] [--] [--version] [-h] <file names> ...
```

通用选项说明：

-h：输出以上命令行帮助信息

--version：输出 gt_compress 程序的版本号

压缩选项说明：

-i：压缩时增加索引，主要用于在压缩文件中快速检索 fastq 文件的某段内容，该选项会降低压缩速度

-a：追加模式，本次压缩的内容会追加到压缩文件中

-g：分组加速压缩，分组越多，需要的 cpu 和内存越多，压缩速度越快。不指定该值时，程序会根据 cpu 和内存自动选择最优值

-o：指定压缩文件名，不指定时，默认为 out.gtz

file_name：需要压缩的文件或目录, 若不指定，则从标准输入中读入数据

解压选项说明：

-n：指定需要检索输出的行数，输出的行数为该值*4，负数为向后检索

-l：指定需检索的起始行，起始行在 fastq 中的位置为（该值-1)*4

-o：指定输出文件名，使用-n 或-l 时需要指定该选项，否则不需要该选项

-t: 解压数据输出至标准输出

file_name：gtz 格式的压缩文件

2.2 压缩和解压示例

一、压缩一个文件

```
$ ./gtz -o sample.fastq.gtz sample.fastq
Powered by GTXLab of Genetalks.
Compressing sample.fastq
[##################################################] 100%
66646536-->8430564 in 0h0m4s [compress ratio:12.65%]
```

二、压缩多个文件（示例：2 个文件）

```
$ ./gtz -o sample.fastq.gtz sample.fastq id
Powered by GTXLab of Genetalks.
Compressing sample.fastq id
[##################################################] 100%
[##################################################] 100%
220018887-->47402215 in 0h0m16s [compress ratio:21.54%]
```

三、压缩目录（目录下有 5 个文件）

```
$ ./gtz -o data.gtz ../../fastq/data
Powered by GTXLab of Genetalks.
Compressing ../../fastq/data
[##################################################] 100%
[##################################################] 100%
[##################################################] 100%
[##################################################] 100%
[##################################################] 100%
318888897-->40975316 in 0h0m23s [compress ratio:12.85%]
```

四、压缩管道数据

```
$ cat sample.fastq | ./gtz -o sample.fastq.gtz
Powered by GTXLab of Genetalks.
Compressing
66646536-->8430561 in 0h0m3s [compress ratio:12.65%]
```

五、解压压缩文件

```
./gtz -d sample.fastq.gtz
Powered by GTXLab of Genetalks.
[##################################################] 100%
8430561-->66646536 in 0h0m5s
```

注：解压到压缩时的相对目录文件中

六、解压压缩目录

```
$ ./gtz -d data.gtz
Powered by GTXLab of Genetalks.
[##################################################] 100%
[##################################################] 100%
[##################################################] 100%
[##################################################] 100%
[##################################################] 100%
40975316-->318888897 in 0h0m32s
```

七、解压压缩文件至管道

```
./gtz -t -d sample.fastq.gtz | head -n 8
Powered by GTXLab of Genetalks.
@ERR194147.1 HSQ1004:134:C0D8DACXX:1:1104:3874:86238/1
GGTTCCTACTTCAGGGTCATAAAGCCTAAATAGCCCACACGTTCCCCTTAAATAAGACATCACGATGGATCACAGGTCTATCACCCTATTAACCACTCACG
+
CC@FFFFFHHHHHJJJFHIIJJJJJIHJIIJJJJJJJIIGIJJIJJJIJJIJIJJJJJJJJIJHHHHFFFDEEEEEEEEDDDCDDEEDDDDDDDDD
@ERR194147.2 HSQ1004:134:C0D8DACXX:2:2104:2852:75174/1
ACTTCAGGGTCATAAAGCCTAAATAGCCCACACGTTCCCCTTAAATAAGACATCACGATGGATCACAGGTCTATCACCCTATTAACCACTCACGGGAGCTC
+
@BBFDFFFHFFHHIHIJJJJFIHHFHFHJCIHFHIJJJJJJIJIJJJIHIJJJJJJBEGIGHIHGHHHEFCDFFEDEEDEEDDD?CCCDDDDDDDDDC
```

注：gtz 的解压数据重定向给 head。第 1 行为 gtz 的标准错误输出，后 8 行为 head 输出。

2.3 Python 支持及相关工具脚本的使用

本软件提供了对 python 的支持与对接，通过开发及执行工具脚本，可以更方便的使用软件。

Python 库名称 gtz，导入 Python 库时确保 gtz.so 文件与 Python 程序在同一目录或该文件所在目录指定在 PYTHONPATH 环境变量中，使用时仅需：

```
import gtz
```

Python GTZ 库提供如下接口：

```
new_decompressor ( gtz_file_name )
```

输入：gtz_file_name 即 gtz 文件的文件名，若不是当前路径的文件，需指定包含完整路径的文件名；

输出： decompressor 类对象，该类对象的使用方式如下实例代码：

```
import gtz
dcomp = new_decompressor( “my_gtz.gtz” )
for ( file_name, file_size, total_fastq_lines_group, this_file_dh ) in dcomp:
    ....
```

其中，decompressor 对象以及内置了迭代器方法，允许用户轻松使用迭代器方法枚举压缩文件中的文件名，文件大小，文件中的行数，以及一个针对这个文件的解压 handle 对象：this_file_dh。

而每个解压 Handle 对象提供以下解压方法：

```
#!/usr/bin/env python
import ctypes
import sys
import gtz
import getopt

#args 为压缩包包名
def show_fileinfo(args):
   cl = gtz.new_decompressor(args)
   for (fn,fs,fg,dh) in cl:
      print "filename:",fn," filesize:",fs," filelines:",fg
```

```
./show_gtzfile_info.py  -f source.gtz
filename: source_dir/source1.fastq  filesize: 661381  filelines: 10000
filename: source_dir/source2.fastq  filesize: 6639641  filelines: 100000
filename: source_dir/source3.fastq  filesize: 66646536  filelines: 1000000
```

- this_file_dh .decompress（user_progress_callback_func）

该方法允许用户解压 this_file_dh 对象所对应的文件，文件名由之前迭代器给出，用户可以提供解压进度回调接口，解压过程中每前进 1%进度，就会调用 user_progress_callback_func（progress_per_cent），progress_per_cent 代表解压进度的浮点数，由解压器提供。

```
# arg1 为压缩包包名,arg2 为要解压的文件名称
def show_fileinfo(arg1,arg2):
   cl = gtz.new_decompressor(arg1)
   for (fn,fs,fg,dh) in cl:
      if fn == arg2:
         def print_progress(progress):
            print progress
         dh.decompress(print_progress)
```

```
./decompress_as_default_name.py –o source_dir/source1.fastq  -f source.gtz
[##################################################] 100%
```

- this_file_dh.decompress_index(start_lines, lines_num , to_which_file)。

此方法允许用户抽取 this_file_dh 对应文件的相应行数，输出到文件。接收参数 1：

为起始行数 2：需要获取的行数 3.输出文件名

（注意：输入的起始行数与需要获取的行数，都必须为 4 的整数倍）

```
#arg1 为压缩包包名，arg2 为要解压的文件名，arg3 为起始行数，arg4 为需要的行数，arg5 为输出的文件名。
def show_fileinfo(arg1,arg2,arg3,arg4,arg5):
  cl = gtz.new_decompressor(arg1)
  for (fn,fs,fg,dh) in cl:
    if fn == arg2:
      dh.decompress_index(int(arg3),int(arg4),arg5)
```

```
./decompress_index_assign_name.py -o source_dir/source1.fastq -f source.gtz -b 40 -s 2000 -i output.txt
[##################################################] 100%
$ wc -l output.txt
2000 output.txt
```

- this_file_dh. decompress_toline(start_lines, lines_num , user_callback)。

此方法允许用户抽取 this_file_dh 对应文件的相应行数，输出到用户的回调函数。

接收参数 1：为起始行数，接收参数 2：需要获取的行数 3.用户提供的回调函数

该回调函数接收一个字符串参数。解压器每次解压出一行即调用一次回调函数。

具体地，我们在比赛提交的压缩包中准备了多个示例 python 程序，同时，这些 python 程序也是可以提供比命令行更加丰富解压能力的小工具。

（注意：输入的起始行数与需要获取的行数，都必须为 4 的整数倍）

```
# arg1 为压缩包包名，arg2 为要解压的文件名，arg3 为起始行数，arg4 为需要的行数
def show_fileinfo(arg1,arg2,arg3,arg4):
  cl = gtz.new_decompressor(arg1)
  for (fn,fs,fg,dh) in cl:
    if fn == arg2:
      def get_decompress_line(decompress_line):
        print decompress_line,
      dh.decompress_toline(int(arg3),int(arg4),get_decompress_line)
```

```
$./decompress_index_getlines.py -o source_dir/source3.fastq  -f source.gtz  -b 40 -s 20
@ERR194147.11 HSQ1004:134:C0D8DACXX:4:1101:19001:189144/1
GATCACAGGTCTATCACCCTATTAACCACTCACGGGAGCTCTCCATGCATTTGGTATTTTCGTCTGGGGGGTATGCACGCGATAGCATTGCGAGACGCTGG
+
@@@FFFFFHFHHHJIIJJIIIJJJJJJJJJJIJJIIGHIJJJJJJJJJJJFHIGIJJIIIHHCEDDB>CDD@CDBDDDDDDDCCCDDDDDDDDBD@
@ERR194147.12 HSQ1004:134:C0D8DACXX:2:2206:12846:29283/1
CCAGCGTCTCGCAATGCTATCGCGTGCATACCCCCCAGACGAAAATACCAAATGCATGGAGAGCTCCCGTGAGTGGTTAATAGGGTGATAGACCTGTGATC
+
CCCFFFFFHHHHHJIJJJJJJJIADHIJIJJJJJGGIIJGHHFEFFEEEDEDDDDDDDBC@BDDDDD<BBD:A?:?CDEEDCD>@BCCDDCCDC>CCCA
@ERR194147.13 HSQ1004:134:C0D8DACXX:3:1104:20699:157340/1
CCAGCGTCTCGCAATGCTATCGCGTGCATACCCCCCAGACGAAAATACCAAATGCATGGAGAGCTCCCGTGAGTGGTTAATAGGGTGATAGACCTGTGATC
+
```

```
CCCFFFFFHHHHGJJJJJJJJJJGIJJJJJJJJJGJIJJHHHFFFFEEEDEDDDEDDDDDDDDDDDDDDDBD@C@@@CDEEDDD>BBDEDDDDDDDDEED
@ERR194147.14 HSQ1004:134:C0D8DACXX:3:1205:17329:12342/1
CCAGCGTCTCGCAATGCTATCGCGTGCATACCCCCCAGACGAAAATACCAAATGCATGGAGAGCTCCCGTGAGTGGTTAATAGGGTGATAGACCTGTGATC
+
CCCFFFFFHHHHGHJJJJJGIIJIJIJIGGIIJJJJJJIJJJHHHFFFFEECCCDDCDDDDDDDDDDDDDDABBD>CDCDDDEECDD??BDEDCCDDDDDDDE
@ERR194147.15 HSQ1004:134:C0D8DACXX:4:2206:2803:99615/1
GATCACAGGTCTATCACCCTATTAACCACTCACGGGAGCTCTCCATGCATTTGGTATTTTCGTCTGGGGGGTATGCACGCGATAGCATTGCGAGACGCTGG
+
@<=DBDDDH<AF><:B?EF@DECEHGCHFCAE;EEA0?D@GFFFFC8??FHIFHI@=CGE4;6@EH=CE?>&)+::C(89><>>>@C>>A:909B<2.000
```

2.3.1 执行 show_gtzfile_info.py 直接查看压缩文件的相关信息

```
$ ./show_gtzfile_info.py  -f source.gtz
filename: source_dir/source1.fastq  filesize: 661381  filelines: 10000
filename: source_dir/source2.fastq  filesize: 6639641  filelines: 100000
filename: source_dir/source3.fastq  filesize: 66646536  filelines: 1000000
```

注：filename:文件名　filesize：文件大小　filelines:文件行数（如果文件为非 fastq 格式，则只会正确显示 filename 与 filesize，filelines 默认为 1 ）。

相关调用的 api 说明：gtz.new_decompressor(args)。gtz 为 gtz.so 模块名称，它提供一个 new_decompressor 方法，此方法入参为压缩包的文件名。返回值为一个 python 元组。里面依次为文件名称，文件大小，文件行数，对应文件的操作指针。

```
10 def show_fileinfo(args):
11     cl = gtz.new_decompressor(args)
12     for (fn,fs,fg,dh) in cl:
13         print "filename:",fn," filesize:",fs,"filelines:",fg
```

2.3.2 执行 decompress_as_default_name.py 脚本直接解压压缩包里指定的文件

```
$ ./decompress_as_default_name.py –o source_dir/source1.fastq  -f source.gtz
[##################################################] 100%
```

相关调用的 api 说明：dh.decompress(print_progress)。dh 为对应文件的操作指针。decompress 为解压到默认文件名的方法。它入参为一个接收解压进度的函数指针。无返回值。

```
# arg1 为压缩包包名,arg2 为要解压的文件名称
11 def show_fileinfo(arg1,arg2):
12     cl = gtz.new_decompressor(arg1)
13     for (fn,fs,fg,dh) in cl:
14         if fn == arg2:
```

```
15          def print_progress(progress):
16              print progress
17          dh.decompress(print_progress)
```

2.3.3 执行 decompress_index_assign_name.py 检索文件中指定的内容

（说明：从压缩包里读取 source_dir/source1.fastq 文件的第 40 行开始，取 2000 行数据）

```
$ ./decompress_index_assign_name.py -o source_dir/source1.fastq -f source.gtz -b 40 -s 2000 -i output.txt
[##################################################] 100%
$ wc -l output.txt
2000 output.txt
```

相关调用的 api 说明：dh.decompress_index(int(arg3),int(arg4),arg5)。此方法为抽取指定文件的相应行数，输出到文件。接收参数 1：为起始行数，接收参数 2：需要获取的行数 3.输出文件名

（注意：输入的起始行数与需要获取的行数，都必须为 4 的整数倍）

```
#arg1 为压缩包包名，arg2 为要解压的文件名，arg3 为起始行数，arg4 为需要的行数，arg5 为输出的文件名。
11 def show_fileinfo(arg1,arg2,arg3,arg4,arg5):
12     cl = gtz.new_decompressor(arg1)
13     for (fn,fs,fg,dh) in cl:
14         if fn == arg2:
15             dh.decompress_index(int(arg3),int(arg4),arg5)
```

2.3.4 执行 decompress_index_getlines.py 索文件中指定的内容 输出至屏幕

（说明：从压缩包里读取 source_dir/source3.fastq 文件的第 40 行开始，取 20 行数据）

```
$ $./decompress_index_getlines.py -o source_dir/source3.fastq  -f source.gtz  -b 40 -s 20
@ERR194147.11 HSQ1004:134:C0D8DACXX:4:1101:19001:189144/1
GATCACAGGTCTATCACCCTATTAACCACTCACGGGAGCTCTCCATGCATTTGGTATTTTCGTCTGGGGGGTATGCACGCGATAGCATTGCGAGACGCTGG
+
@@@FFFFFHFHHHJIIJJJIIJJJJJJJJJJIJJIJJIGHIJJJJJJJJJJJFHIGIJJJIIIHHCEDDB>CDD@CDBDDDDDDDCCCDDDDDDDDBD@
@ERR194147.12 HSQ1004:134:C0D8DACXX:2:2206:12846:29283/1
CCAGCGTCTCGCAATGCTATCGCGTGCATACCCCCCAGACGAAAATACCAAATGCATGGAGAGCTCCCGTGAGTGGTTAATAGGGTGATAGACCTGTGATC
+
CCCFFFFFHHHHHJIJJJJJJJIADHIJIJJJJJJGGIIJGHHFEFFEEEDEDDDDDDDBC@BDDDDD<BBD:A?:?CDEEDCD>@BCCDDCCDC>CCCA
@ERR194147.13 HSQ1004:134:C0D8DACXX:3:1104:20699:157340/1
```

```
CCAGCGTCTCGCAATGCTATCGCGTGCATACCCCCCAGACGAAAATACCAAATGCATGGAGAGCTCCCGTGAGTGGTTAATAGGGTGATAGACCTGTGATC
+
CCCFFFFFHHHHGJJJJJJJJJJGIJJJJJJJJJGJIJJHHHFFFFEEEDEDDDEDDDDDDDDDDDDDDDBD@C@@@CDEEDDD>BBDEDDDDDDDDEED
@ERR194147.14 HSQ1004:134:C0D8DACXX:3:1205:17329:12342/1
d
+
CCCFFFFFHHHHGHJJJJJGIIJIJIJIGGIIJJJJJJIJJJHHHFFFFEECCCDDCDDDDDDDDDDDDDDABBD>CDCDDDEECDD??BDEDCCDDDDDDDE
@ERR194147.15 HSQ1004:134:C0D8DACXX:4:2206:2803:99615/1
GATCACAGGTCTATCACCCTATTAACCACTCACGGGAGCTCTCCATGCATTTGGTATTTTCGTCTGGGGGGTATGCACGCGATAGCATTGCGAGACGCTGG
+
@<=DBDDDH<AF><:B?EF@DECEHGCHFCAE;EEA0?D@GFFFFC8??FHIFHI@=CGE4;6@EH=CE?>&)+::C(89><>>>@C>>A:909B<2.000
```

相关调用 api 的说明： dh.decompress_toline（ ）。此方法可以获取指定文件的的相应行数到 python 执行环境.它接收 3 个参数，参数 1：起始行数 参数 2：获取的行数 参数 3：函数指针

（注意：输入的起始行数与需要获取的行数，都必须为 4 的整数倍）

```
# arg1 为压缩包包名，arg2 为要解压的文件名，arg3 为起始行数，arg4 为需要的行数
11 def show_fileinfo(arg1,arg2,arg3,arg4):
 12    cl = gtz.new_decompressor(arg1)
 13    for (fn,fs,fg,dh) in cl:
 14       if fn == arg2:
 15          def get_decompress_line(decompress_line):
 16             print decompress_line,
 17          dh.decompress_toline(int(arg3),int(arg4),get_decompress_line)
```