

用于视频下载/视频转音频/音频分割基于R语言的轻量流程化脚本

Author: Wang Jiahao

Date: 2020-10-27

GitHub: <https://github.com/JiahaoWongg/Bilidown>

用于视频下载/视频转音频/音频分割基于R语言的轻量流程化脚本

- 一、简介
- 二、功能介绍
 - ① CMD模式
 - ② 文件模式
- 三、环境配置
 - ① 语言环境配置
 - ② 安装you-get 和 ffmpeg
- 三、下载脚本
- 四、脚本运行测试
 - ① 打开终端并切换工作路径
 - ② 脚本运行测试
- 五、参数设置
 - ① CMD模式
 - ② 文件模式
 - file文件格式要求
 - 该模式参数设置
- 六、开始实战
 - ①CMD模式
 - 单P下载
 - 多P下载
 - ②文件模式
- 视频弹幕文件下载，并以 `txt` 格式输出
 - 使用方法
 - 运行
 - 对比
- FFmpeg的一些常用功能
 - 功能介绍
 - ① 格式转换
 - ② 视音频切割
 - 合并音视频

一、简介

话说这也算是第二个版本了，[第一个版本](#)是写在新冠暴发之初，因无法返校无聊至极而写。第一版所实现的功能不多，因此一直有改进的想法。

脚本所实现的功能看标题大家也能猜出一二，构思由来已久，但我接触代码的世界在当时也才半年而已。随着知识的增加和实际需求的提出，便萌生了用R语言实现这些功能的想法了。

我比较喜欢听音乐，特别是周杰伦的歌，而其歌曲的扒带也是我的最爱，B站有许多优质的歌曲扒带视频，因此我就想要把这些视频转为音频，放到手机音乐播放器里。后来又实现了对音频文件的分割，并将这些过程流程化，你只需要提供视频链接和简单的参数的就可实现这些功能。总共也才200来行代码，为了便于阅读，其中大概含有 3/4 的注释行和空行。

先说这么多，接下来具体介绍脚本所实现的功能。

二、功能介绍

根据提供视频链接 [URL](#) 来源的不同，将功能划分为两大功能模式：来源于命令行 [CMD](#) 和来源于文本文件。

① CMD模式

- 视频URL直接由命令行提供，即直接在终端 (windows 系统是 win + x + c 打开) 里输入URL
- 根据URL获取视频文件，默认下载最高清晰度格式，VIP 视频不能下载，不能解析的URL不能下载
- 如果所下载的视频文件为FLV格式，则将其转为更容易播放的MP4格式
- 如果你愿意，你可以将MP4转为MP3音频格式，便于听
- 如果你愿意，你可以将MP3分割为（既保留）你想要的时间段
- 根据你所指定的视频和音频的文件储存路径（既文件夹），将结果文件储存在指定路径内
- 所有的这些只需一行代码即可出结果

② 文件模式

- 视频URL不是直接输入的，而是储存在单独的一个文件里。这样你可以实现多个视频的下载与后续处理
- 每一个URL循环一次①中所述的过程，实现批量下载

如果对于代码和命令完全陌生的朋友，或许有些地方不明白，不过没关系，具体操和参数设置作将在下一节具体讲述。

三、环境配置

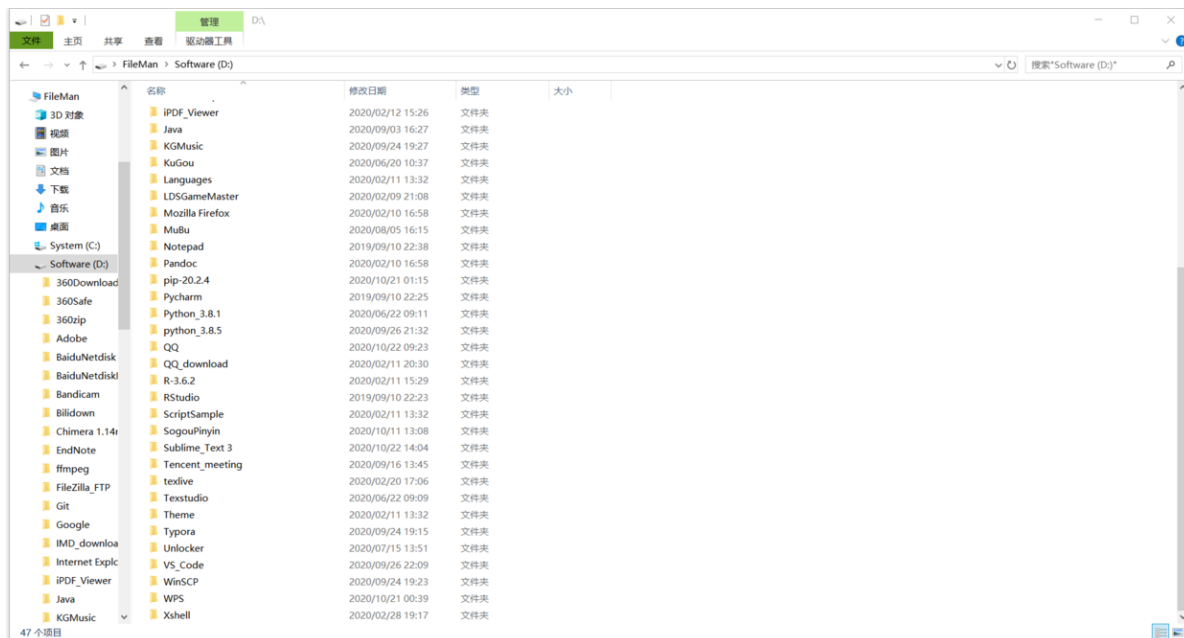
这一节是重头戏，也是最难的部分，使用不好的朋友要多读读。

另外，由于我电脑是window系统，所以我只讲在window里的操作方法。

① 语言环境配置

需要的语言环境及工具：Python、R、you-get、ffmpeg，你已经有的就选择性安装

接下来——介绍，不过首先讲一点，安装软件的时候要指定安装路径，不要总把软件放在C盘的 `Program Files` 里。安装的路径必须要知道，后面需要用到。比如我将D盘作为软件盘，找软件路径的时候非常方便：



- Python: 下载地址: [Python](#) (用于安装you-get 和 ffmpeg)
- R: 下载地址: [R](#), 用于运行脚本
- 添加R和Python的环境变量, 参照: [Windows 系统添加环境变量](#)
- 检验两个语言环境是否安装成功:

快捷键 `win + X + C` 打开命令提示符面板 (以下简称 `终端`) :

```
1 Microsoft Windows [版本 10.0.18362.1082]
2 (c) 2019 Microsoft Corporation。保留所有权利。
3
4 C:\Users\Jay Chou>
```

运行:

```
1 python --version & Rscript --version
```

得到以下结果, 则安装并环境配置成功! 恭喜~

```
1 C:\Users\Jay Chou>python --version & Rscript --version
2 Python 3.8.5
3 R scripting front-end version 3.6.2 (2019-12-12)
4
5 C:\Users\Jay Chou>
```

② 安装you-get 和 ffmpeg

- you-get: 下载视频文件和弹幕文件之类的
- ffmpeg: 文件格式转换

运行:

```
1 pip install you-get & pip install FFmpeg
```

运行:

```
1 you-get --version & ffmpeg -version
```

前三行得到以下结果, 则安装成功!

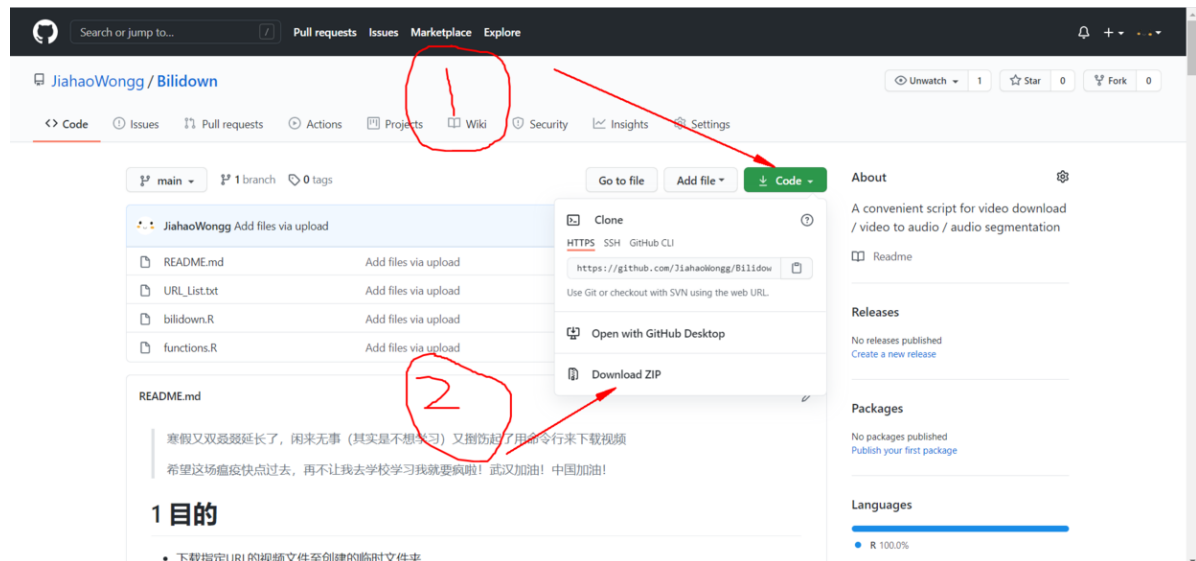
```
1 C:\Users\Jay Chou>you-get --version & ffmpeg -version
2 you-get: version 0.4.1456, a tiny downloader that scrapes the web.
3 ffmpeg version git-2020-01-15-0dc0837 Copyright (c) 2000-2020 the FFmpeg
  developers
4 built with gcc 9.2.1 (GCC) 20200111
```

Python安装的软件不需要额外添加环境变量

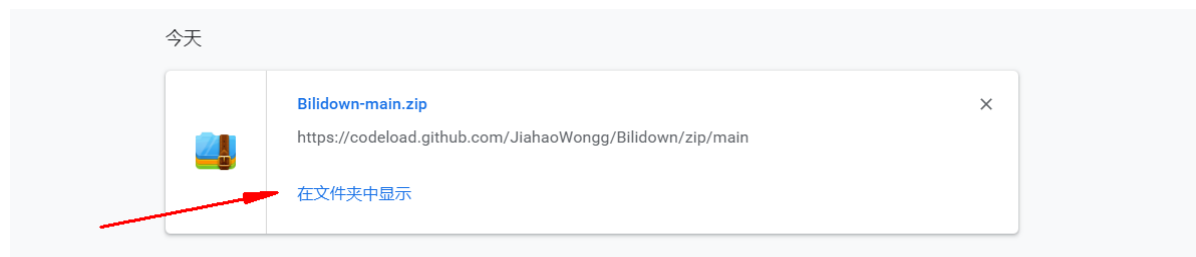
至此, 你已经完成了最难的部分, Congratulations !

三、下载脚本

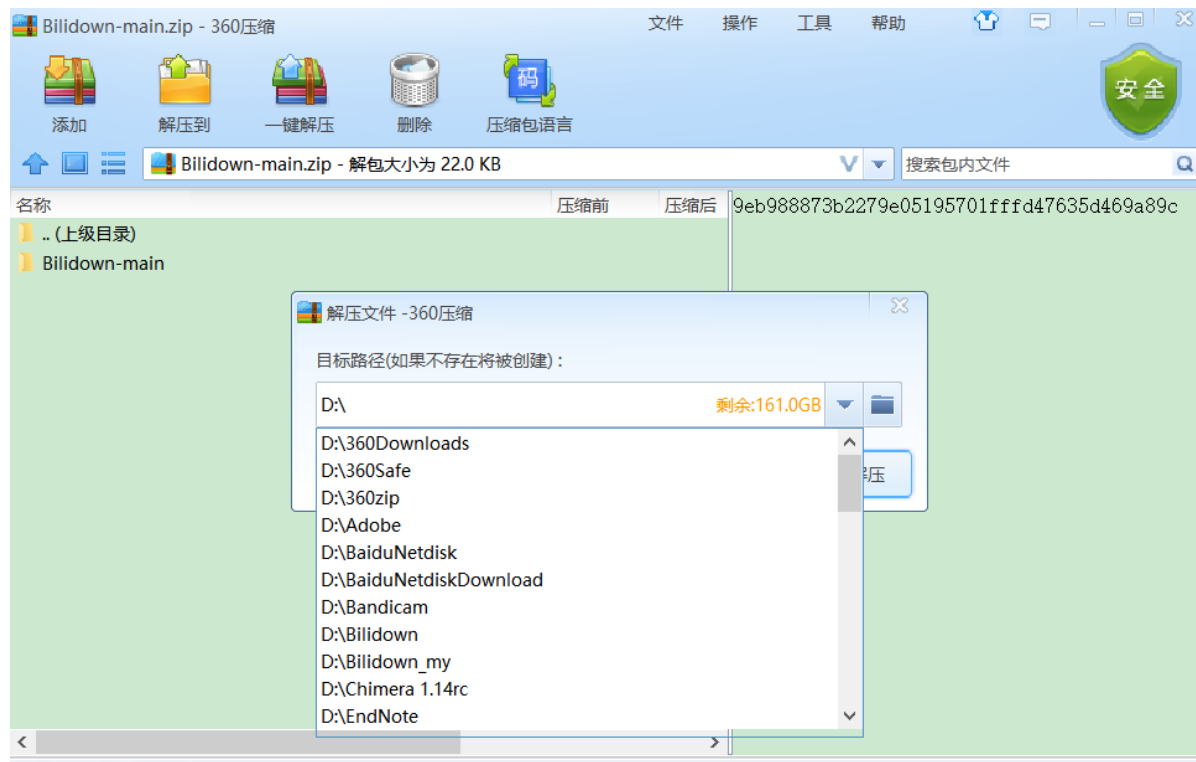
访问: <https://github.com/JiahaoWongg/Bilidown>



在浏览器下载历史内, 打开该文件所在路径:



解压, 路径我设为: D:\, 尽可能简单



将文件夹简化重命名为 Bilidown



以上设置即为初次使用该脚本所需的所环境配置和操作

以下为以后每次运次脚本时的流程

四、脚本运行测试

① 打开终端并切换工作路径

快捷键 `win + x + c` 打开命令提示符面板

运行：

```
1 d: & cd \Bilidown
```

结果：

```
1 C:\Users\Jay Chou>d: & cd \Bilidown
2
3 D:\Bilidown>
```

② 脚本运行测试

运行：

```
1 Rscript bilidown.R -h
```

结果：

```
1 D:\Bilidown>Rscript bilidown.R -h
2 Usage: bilidown.R [options]
3
4
5 Options:
6     --url=URL
7         video URL.
8
9     --list=LIST
10        whether download playlist.
11
12     --mp3=MP3
13        whether tranfer to mp3.
14
15     --file=FILE
16        whether get URL(s) from file.
17
18     --mp3Folder=MP3FOLDER
19        Mp3 output folder.
```

```

20
21     --mp4Folder=MP4FOLDER
22         Mp4 output folder.
23
24     --name=NAME
25         Rename output file.
26
27     --start=START
28         Start point of mp3 cut
29
30     --end=END
31         end point of mp3 cut
32
33     -h, --help
34         Show this help message and exit
35
36
37
38 D:\Bilidown>

```

出现这样的结果说明一切正常，其中 `Options` 即为运行脚本所需设的参数，可以阅读一下，根据提示差不多能看出其作用。

测试成功后我们就要进行真正的运行了，该脚本和其他命令行软件一样，都必须赋予指定格式的参数或文件，一点错误一般都是不允许的，或者结果可能会不是你想要的。所以下一节的参数设置也至关重要，决定了能不能得到你想要的结果！

接下来将——解释每一参数的作用和如何设置

五、参数设置

第二节内容提到，该脚本有两个使用模式，`URL来源于CMD模式` 和 `URL来源于文件模式`

因此参数设置也要分为这两部分讲解

首先要知道的是：

- 参数的使用方法为： `--参数名 参数值`，在下一节的案例中将会更详细地看到
- 每种模式你只需要提供与之对应的url来源即可，既参数url和file必须且只能存在其中一个
- 如果两个参数url和file都存在或都不存，脚本也会提示你必须要提供其中之一
- 下面所述的必选和可选，前者意为必须给这个参数赋值，而后者可赋可不赋
- 当给某个可选参数赋值时，其他的一些参数可能就变为了必选参数。比如你可以选择要不要对MP3文件进行分割，如果需要分割，则start参数需要赋值，这时end参数自然也就是必选的了

① CMD模式

- url；视频链接；该模式下为必选参数

即为视频链接，直接从视频观看页面的地址栏里粘贴即可，如下图红框内即为该视频的URL，复制下来即可



需要注意的是：URL要用双引号包围起来，因为有的URL中含有特殊字符，如 &，会被识别为命令字符，会发生报错

对应脚本为：

```
1 Rscript bilidown.R --url "https://www.bilibili.com/video/BV1dt4s11a7FX?from=search&seid=3497926814758339505"
```

别猴急运行，说运行了再运行

- list; 是否下载视频列表; 可选参数

我们都知道B站的视频是可以分P的，对于一个分P视频的其中一P的URL，我们可以选择只下载这一P或者下载所有P，当然也可以下载指定的P，不过you-get实现不了，作者可以实现，但作者太懒就没加这个功能。

默认是不下载所有P的，不赋值即可。如果想下载所有P，则需要赋值为 `TRUE`

下载全P的情况下此时的代码：

```
1 Rscript bilidown.R --url "https://www.bilibili.com/video/BV1dt411a7FX?from=search&seid=3497926814758339505" --list TRUE
```

- mp3, 是否将MP4文件转化为MP3文件, 可选参数

默认不转换，如果你只想下载视频，则不需赋值；如果你还想得到音频，则需赋值为 `TRUE`

- mp3Folder; MP3文件保存路径; 可选参数，当MP3为 `TRUE` 时为必选参数

电脑里可以原先不存在你所指定的文件夹，不存在时脚本会帮你创建；下同

- mp4Folder; MP4文件保存路径; 必选参数

插播一条：文件保存路径必须使用 绝对路径，windows绝对路径以盘符开头，如：

`D:/Bilidown/MP4`

- name; 是否对下载的文件重命名; 可选参数

默认不重命名，需要重命名的话需要将名字赋值给name参数

如在以上全为 `TRUE` 的情况下，将下载的视频及转的音频均命名为 牛逼脚本下载的东东 的代码为：

```
1 Rscript bilidown.R --url "https://www.bilibili.com/video/BV1dt411a7FX?
  from=search&seid=3497926814758339505" --list TRUE --mp3 TRUE --mp3Folder
  D:/Bilidown/MP3 --mp4Folder D:/Bilidown/MP4 --name 牛逼脚本下载的东东
```

则输出文件为: 牛逼脚本下载的东东.mp3 和 牛逼脚本下载的东东.mp4

- start 及 end

显然这两个也是成对出现的, 如果你不想将MP3文件分割为你想要的部分, 则两个都赋值即可
若果想分割, 则需要分别对它们进行赋值

赋的值为时间点, 格式为 00:00:00, 既时分秒

比如你只想保留转的MP3文件的10秒至1分10秒之间的部分, 代码为:

```
1 Rscript bilidown.R --url "https://www.bilibili.com/video/BV1JE411D74q?
  from=search&seid=12356753915334889298" --mp3 TRUE --mp3Folder
  D:/Bilidown/MP3 --mp4Folder D:/Bilidown/MP4 --name 牛逼脚本下载的东东 --start
  00:00:10 --end 00:01:10
```

这就是该模式下最完整的代码, 整理好看一点:

```
1 Rscript bilidown.R --url "https://www.bilibili.com/video/BV1dt411a7FX?
  from=search&seid=3497926814758339505"
2 --mp3 TRUE
3 --mp3Folder D:/Bilidown/MP3
4 --mp4Folder D:/Bilidown/MP4
5 --name 牛逼脚本下载的东东
6 --start 00:00:10
7 --end 00:01:10
```

一目了然。注意我把list参数给去掉了, 因为, 如果你下载全P视频, 那你对多个视频在同样的时间点进行切割显然是不符合逻辑的

② 文件模式

file文件格式要求

由于既要考虑用户的体验而格式不能太复杂, 同时又要考虑作者的能力范围, 因为要实现多个视频的流程化处理, 该文件内容稍许复杂, 其实也不是很复杂, 就是将CMD模式的一些参数值写入该文件中而已, 需要注意的是文件内容格式必须遵守规定, 具体规定如下:

- 每一行是相互独立的下载信息, 文件格式为 **txt** 格式
- 对于某一行来说, 你**最多**可以提供6个下载信息, 既CMD模式中提到的 `url mp3 list name start end`
- 只需要提供**参数值**即可, 参数名不需提供, 每个参数值之间必须由**空格**作为分隔
- 当然, 你也可以提供**少于**6个的下载信息, 但必须遵循的是, 参数值的顺序是**固定**的, 即按照 `url mp3 list name start end` 的顺序, 如果你所提供的参数在这6个里面**不是连续**的, 比如你只想提供 `url name` 两个参数, 则你必须将中间空缺的位置的值设为 **no**, 既 `url no no name`, 右侧空缺的参数则不需写 **no**, 直接回车即可
- 例如, 你将一个视频的URL放在文件里, 你想将下载的视频转换为音频, 但只下载此URL所代表的单P视频, 不需要重命名, 而需要将音频分割, 则在URL文件里你可以这样写:

```
1 "https://www.bilibili.com/video/BV1dt411a7FX?
  from=search&seid=3497926814758339505" TRUE no no 00:00:10 00:01:10
```




```

18 Transcoding flv file to mp4 file
19 Transcoding mp4 file to mp3 file
20 Cutting mp3 file
21 Copying & rename files and remove tmpFolder
22 Done!
23
24 D:\Bilidown>

```

结果:

MP4 文件:

> FileMan > Software (D:) > Bilidown > MP4					
名称	日期	类型	大小	时长	
 周杰伦-秘密花园(扒带).mp4	2020/10/23 23:13	MP4 - MPEG-4 ...	22,463 KB	00:02:05	

MP3文件:

> FileMan > Software (D:) > Bilidown > MP3					
名称	#	标题	参与创作的艺术家	唱片集	
 周杰伦-秘密花园(扒带).mp3					



可以看到时长只有一分钟, 因为我取的是 00:00:10 至 00:01:10

多P下载

多P下载的也不必重命名了

命令:

```

1 Rscript bilidown.R --url "https://www.bilibili.com/video/BV17t411k7et?
  from=search&seid=3759713561223284474" --mp3 TRUE --list TRUE --mp3Folder
  D:/Bilidown/MP3 --mp4Folder D:/Bilidown/MP4

```

屏幕显示的下载过程:

```

1 D:\Bilidown>Rscript bilidown.R --url
  "https://www.bilibili.com/video/BV17t411k7et?
  from=search&seid=3759713561223284474" --mp3 TRUE --list TRUE --mp3Folder
  D:/Bilidown/MP3 --mp4Folder D:/Bilidown/MP4
2
3 Downloading video file ...
4 site:          Bilibili
5 title:         你能猜到这是周杰伦哪一首歌吗? 答案在2P (P1. 你能猜到这是周杰伦哪
  一首歌吗? 答案在2P)
6 stream:
7   - format:     dash-flv720
8     container:   mp4
9     quality:     高清 720P
10    size:        4.3 MiB (4464751 bytes)
11    # download-with: you-get --format=dash-flv720 [URL]

```

```

12
13 Downloading 你能猜到这是周杰伦哪一首歌吗？答案在2P (P1. 你能猜到这是周杰伦哪一首歌吗？
   答案在2P).mp4 ...
14 100% ( 4.3/ 4.3MB) [2/2] 473
   kB/s
15 Merging video parts... Merged into 你能猜到这是周杰伦哪一首歌吗？答案在2P (P1. 你能
   猜到这是周杰伦哪一首歌吗？答案在2P).mp4
16
17 Skipping captions or danmaku.
18 site: Bilibili
19 title: 你能猜到这是周杰伦哪一首歌吗？答案在2P (P2. 致敬科比_考古MV《天地一斗》)
20 stream:
21 - format: dash-flv720
22 container: mp4
23 quality: 高清 720P
24 size: 45.2 MiB (47435509 bytes)
25 # download-with: you-get --format=dash-flv720 [URL]
26
27 Downloading 你能猜到这是周杰伦哪一首歌吗？答案在2P (P2. 致敬科比_考古MV《天地一
   斗》).mp4 ...
28 100% ( 45.2/ 45.2MB) [2/2] 92
   kB/s
29 Merging video parts... Merged into 你能猜到这是周杰伦哪一首歌吗？答案在2P (P2. 致敬
   科比_考古MV《天地一斗》).mp4
30
31 Skipping captions or danmaku.
32 Transcoding flv file to mp4 file
33 Transcoding mp4 file to mp3 file
34 Copying & rename files and remove tmpFolder
35 Done!
36
37 D:\Bilidown>

```

结果：

MP4文件：

FileMan > Software (D:) > Bilidown > MP4					
名称	日期	类型	大小	时长	
你能猜到这是周杰伦哪一首歌吗？答案在2P (P1. 你能猜到这是周杰伦哪一首歌吗？答案在2P).mp4	2020/10/23 23:54	MP4 - MPEG-4 ...	4,370 KB	00:00:17	
你能猜到这是周杰伦哪一首歌吗？答案在2P (P2. 致敬科比_考古MV《天地一斗》).mp4	2020/10/23 23:54	MP4 - MPEG-4 ...	46,443 KB	00:03:05	



MP3文件：

FileMan > Software (D:) > Bilidown > MP3					
名称	#	标题	参与创作的艺术家的	唱片集	
你能猜到这是周杰伦哪一首歌吗？答案在2P (P1. 你能猜到这是周杰伦哪一首歌吗？答案在2P).mp3					
你能猜到这是周杰伦哪一首歌吗？答案在2P (P2. 致敬科比_考古MV《天地一斗》).mp3					

②文件模式

切记URL文件内容格式！！！！

我准备了几个测试数据，保存在同一文件夹下的`test.txt`文件中，有5个链接，给与了不同的处理模式。大家也可以在当前文件夹下找到。在使用的时候按照这样的格式创建就行了，记得运行的代码文件名要改成你自己写的。


```
42 Transcoding flv file to mp4 file
43 Transcoding mp4 file to mp3 file
44 Copying & rename files and remove tmpFolder
45 Done!
46
47 Processing serial number 3, Remian 2
48
49 Downloading video file ...
50 site:          Bilibili
51 title:         周杰伦在快本给吴昕的三键成曲扒带
52 stream:
53   - format:    dash-flv
54     container:  mp4
55     quality:   高清 1080P
56     size:      31.1 MiB (32599288 bytes)
57     # download-with: you-get --format=dash-flv [URL]
58
59 Downloading 周杰伦在快本给吴昕的三键成曲扒带.mp4 ...
60 100% ( 31.1/ 31.1MB)  [2/2] 96
61 kB/s
62 Merging video parts... Merged into 周杰伦在快本给吴昕的三键成曲扒带.mp4
63
64 Skipping captions or danmaku.
65 Transcoding flv file to mp4 file
66 Transcoding mp4 file to mp3 file
67 Copying & rename files and remove tmpFolder
68 Done!
69
70 Processing serial number 4, Remian 1
71
72 Downloading video file ...
73 site:          Bilibili
74 title:         周杰伦编曲扒带第一张专辑《反方向的钟》编曲扒带
75 stream:
76   - format:    dash-flv
77     container:  mp4
78     quality:   高清 1080P
79     size:      52.9 MiB (55428095 bytes)
80     # download-with: you-get --format=dash-flv [URL]
81
82 Downloading 周杰伦编曲扒带第一张专辑《反方向的钟》编曲扒带.mp4 ...
83 100% ( 52.9/ 52.9MB)  [2/2] 100
84 kB/s
85 Merging video parts... Merged into 周杰伦编曲扒带第一张专辑《反方向的钟》编曲扒带.mp4
86
87 Skipping captions or danmaku.
88 Transcoding flv file to mp4 file
89 Transcoding mp4 file to mp3 file
90 Cutting mp3 file
91 Copying & rename files and remove tmpFolder
92 Done!
93
94 Processing serial number 5, Remian 0
95
96 Downloading video file ...
97 site:          Bilibili
```


> FileMan > Software (D:) > Biliidown > MP3

名称	#	标题	参与创作的艺术	唱片集
我把周杰伦婚礼进行曲扒带编出来了.mp3				
周杰伦_《蒲公英的约定》完整扒带还原-林迈可编曲_(P1_蒲公英的约定).mp3				
周杰伦_《蒲公英的约定》完整扒带还原-林迈可编曲_(P2_蒲公英的约定伴奏).mp3				
周杰伦-反方向的钟(扒带).mp3				
周杰伦-秘密花园(扒带).mp3				
周杰伦-周杰伦快本三键成曲.mp3				

虽然写URL文件有些麻烦，但现在看来是不是挺值得的(๑)~

视频弹幕文件下载，并以txt格式输出

结果展示：

```
1  晚安
2  劳动力内卷元凶
3  我去你mother的
4  你睡的着？你什么出息了？深井冰
5  早上看的    中午睡的    晚上走的
6  这个老师说的是上课睡觉的人 不要恶意曲解
7  别骂了别骂了，我这就滚去学习
8  昨天刚去文都听了一轮汤哥的课
9  昨天刚去文都听完汤哥的课
10 没有
11 不睡觉干嘛
12 可我真的好困啊
13 晚安
14 哈哈哈哈哈我没出息了
15 晚安
16 别骂了别骂了
17 你怎么睡的嫂子啊？
18 不好意思，我不睡觉了
19 睡前必看
20 我的闹钟铃声
21 郭老师别看这些没有用的
22 郭老师别看这些没有用的东西
23 汤爷
24 汤神这是几天没睡觉了？
25 循环中
26 对不起>人<
27 马鞍山钢铁厂马
28 看他那样得几天不睡觉啊
29 你有灵魂吗
```

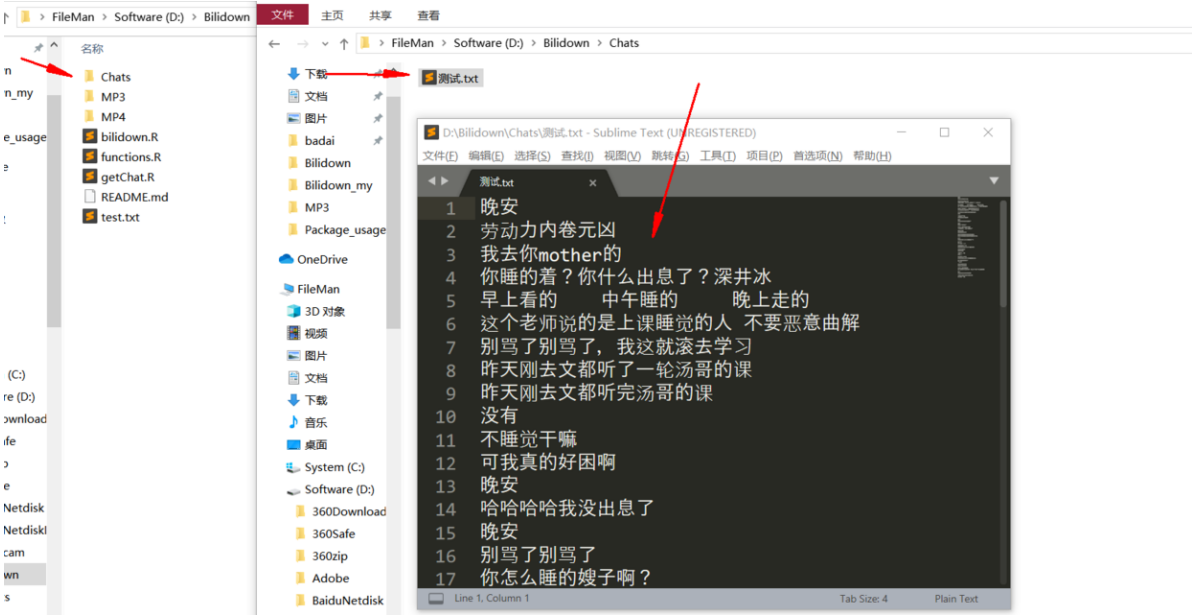
排在上面的是最新的发布的弹幕，越往下越晚

使用方法

- 脚本：位于同一目录下的 `getChat.R`
- 参数
 - url；视频链接；必选参数
 - name；重命名；可选参数
 - 不重命名会默认命名为`chat.txt`，所以为了防止文件重复，最好命名一下
 - outFolder；指定txt文件保存位置；可选参数
 - 指定路径可以不存在，脚本会自动检测

运行

```
1 Rscript getChat.R --url "https://www.bilibili.com/video/BV1aw41ly7Ap?from=search&seid=1744832395255800797" --name 测试 --outFolder Chats
```



对比



FFmpeg的一些常用功能

文件格式的转换，视音频的切割都是使用ffmpeg软件完成的，我只是加以包装整合。

如果小伙伴们想单独完成上述某一功能的话，ffmpeg还需略懂一点的

功能介绍

同样也是在终端里运行，直接使用 ffmpeg 命令即可，具体常用功能如下

① 格式转换

- 不同视频格式之间的转换，视音频之间的转换等
- 比如输入文件为：xxx.format1，输出文件为：xxx.format2
- 即要把 xxx 文件从格式 format1 转换为 format2
- 命令为：

```
1 | ffmpeg -i xxx.format1 `xxx.format2`
```

- 以flv转mp4为例：

```
1 | ffmpeg -i test.flv `test.mp4`
```

② 视音频切割

- 视频同样是可以切割的，不过我没有加入这个功能，不过用法和音频切割完全一样
- 以音频切割为例：

```
1 | ffmpeg -ss 00:00:10 -t 00:01:00 -i input.mp3 -c copy output.mp3
```

其中：

- -i 输入的音频
- -c copy 用原来的编码并复制到新文件中
- -ss 起始时间
- -t 截取音频时间长度

另外，搜资料时看到一位[老兄写道](#)音视频还可以有转场效果，ffmpeg的功能真强大：

从0s开始淡入3s

```
1 | ffmpeg -i input.mp3 -af "afade=t=in:ss=0:d=3" out.mp3
```

从50秒开始淡出5s

```
1 | ffmpeg -i input.mp3 -af "afade=t=out:st=50:d=5" out.mp3
```

参数 -af 为音频滤镜，afade 为音频淡入淡出。相应的如果是对视频的操作，这里的参数 afade 处可以使用 fade。

合并音视频

```
1 | ffmpeg -i input1.mp4 -i input2.mp4 -i input3.mp4 -lavfi hstack=inputs=3  
output.mp4
```

需要——指定输入文件和输入文件个数

