课程代码：BI290

上海交通大學

本科生课程论文
（2018-2019 学年春季学期）

论文题目：生物信息学差异比较软件 **Biodiff** 程序设计

学生姓名：张　　学号：**517111910078**

课程名称：生物计算编程语言

授课教师：韦朝春

写作时间：**2019** 年 **6** 月

目录

生物信息学差异比较软件 Biodiff 的程序设计

张

上海交通大学 生命科学技术学院

摘　要：Biodiff 是一个简单的生物信息学软件，能够根据要求比对两个生物信息数据文件的信息，并输出不同的输出文件夹。该程序有两个主要功能，一个是比较两个文件每条信息的范围是否重叠，另一个功能是比较两个文件每条信息的名字是否重叠。经过算法优化，该程序可以较快完成比对，并对错误输入信息进行较为全面的提示。

关键词：生物信息学，差异比较

Programming of Biodiff – A Bioinformatics Difference Comparison Software

ZHANG

School of Life Sciences and Biotechnology, Shanghai Jiao Tong University

Abstract: Biodiff is a simple bioinformatics software that compares the information of two bioinformatics data files to different requirements and outputs different output folders. The program has two main functions. One is to compare whether the criteria of everyone of information in two files overlaps. The other function is to compare whether the name of everyone of information in the two files overlaps. After algorithm optimization, the program can complete the comparison faster and provide a more comprehensive prompt for the error input information.

Keywords: Bioinformatics, Difference comparison

1　程序设计思路

本程序总体思路如下：

对于存储方式，采用结构体数组存储，且为了避免浪费内存，在存储前先得到每个文件原始的行数，之后按照各自的行数声明对应大小的结构体数组，这样对于不同大小的文件均能够节省不同的内存。

对于区间比对方式，首先对每一个文件数据结构体数组按照起始值由小到大快速排序，如果起始值相等则按照终止值由小到大排序。排序后进行重叠标记，依据是一个数据的起始值在另一个数据的区间内，代表两数据重叠，各自标记为发生重叠。经过两次正反标记后，按照事先的要求输出文件。

对于名字比对方式，采用了字典树的数据结构，将两个数据集名字各自形成一个字典树。比对时用一个数据集的数据在另一个数据集的字典树中查找，如果

前者名字和后者完全相同或者是后者的前缀，则代表二者重叠，并实时输出到事先要求的输出文件。

对于帮助函数，采用了常用的帮助信息格式，使整个提示信息更加规整，阅读更方便，避免因为换行而带来误解。

对于错误提示，针对可能的多数错误进行了错误提示并退出程序，具体可见第二部分内容。

2 具体程序内容

源代码在报告最后单独附录。

2.1 原始数据及命令信息预处理模块

2.1.1 创建输出结果文件夹

本部分通过程序第 242-252 行 result_mkdir()函数实现。

创建一个输出结果文件夹 result，如果文件夹已经存在则不输出，否则新建并设置权限读写。需要注意的是如果是在 Windows 系统中编译，第 249 行不需要设置权限。

2.1.2 获得命令行选项及其信息

本部分通过程序里第 258-335 行的 get_option()函数实现。

通过内建的 getopt()函数，设定了五个可接受的参数，即-c 和-n 两种模式，再加上-a 和-b 两个范围，以及-h 获得帮助。其中帮助信息在宏定义里进行了定义，可以在程序开头看到。

-a 和-b 的选项信息存储到了 file_colomn 字符串数组中，用以之后进行提取处理。在选项获取结束后，文件地址通过 optind 和 argv 进行获取，并存储到 file_path 字符串数组中备用。

2.1.3 提取处理区域

本部分通过程序第 350-385 行 get_region()函数实现。

为了处理从 get_option()函数中获得的 file_column 字符串数组，先判断输入的模式，然后对-c 通过格式化读取进行提取，-n 用直接的 atoi()函数转换为数字。其中 file_col 是用来存储两个文件各自区域的数组，如果是-n 模式则每个文件对应的第二个区域为-1 作为指示。

2.1.4　获得文件原始行数

本部分通过程序第 388-406 行 get_line_num()函数实现。

为了快速获得文件的原始行数，我通过文件流读取和 fgets()移动按行位置指针，计数获得文件行数。其中对于空文件，由于会至少读一次，所以得到的行数为 1，但空文件不是正确的输入文件，因此输出警告并退出程序。

2.1.5　读取并存储文件数据

本部分通过程序第 409-464 行 read_data()函数实现。

为了排除空行，首先将读取到的每一行存储到临时字符串中，先判断是否为空行，如果不是则继续处理，是的话就进入下一行。不是空行时，首先将整行数据存储到对应的结构体 data 中，然后对临时字符串按照制表符、换行符、分号和引号进行循环分割，保存下 file_col 存储的区域的数据到对应的结构体元素中，其中判断应该处理的数据类型通过 file_col 的第二个数是否为-1 判断（见 2.2）。

需要注意的是，当读取的是第一列时，不应该进行后续循环，因此单独进行处理，不在循环中进行处理。

最终函数将返回除去空行后实际读取的行数，在快速排序时将会用到这一数据。

2.1.6　打开文件读取文件流

本部分通过程序第 467-477 行 open_file()函数实现。

打开文件直接使用内建函数 fopen()即可，本函数只是增加了判断是否成功打开或文件是否存在。

2.1.7　原始数据处理主函数

本部分通过程序第 480-493 行 compose_data()函数实现。

这部分将前述各函数综合起来作为数据处理总函数，并返回已经存储完成的数据集。其中分配内存依据为文件原始行数。

2.2 区域重叠判断功能及文件输出模块

2.2.1　判断是否重叠并标记

本部分通过程序第 499-526 行 is_overlap()函数实现。

比对的逻辑如下：假设两个数据集分别为 A 和 B，均已经按照起始值由小到大排序。当 B 的起始值位于 A 的起始值和终止值之间，代表两个数据发生了重

叠，对两个数据同时标记。在循环处理的开始时，如果 B 的起始值小于 A 的起始值，则向后一个，直到不小于为止；此后只要 B 的起始值小于等于 A 的终止值，则仍然重叠；直到 B 的起始值大于 A 的终止值，此后数据都不会重叠，退出此次循环，A 数据集移到下一个，重复上述过程。需要注意的是，B 数据不需要每次从头开始，只需要从上一次开始比对时的起始位置开始即可，因为在这个位置之前的 B 数据一定不会与 A 此时任何一个数据重叠。

2.2.2　标记后的数据输出至文件

本部分通过程序第 551-560 行 cprint()函数实现。

对于已经标记后的数据，如果重叠则输出至对应的 A&B_A.gtf 或 A&B_B.gtf 文件，否则输出至 A-B.gtf 或 B-A.gtf 文件。

2.2.3　对数据集进行快速排序

本部分通过程序第 563-581 行 end_cmp()函数和 sort_data()函数实现。

前一个函数是内建函数 qsort()的输入函数参数，意思是按照起始值由小到大排序，如果起始值相同则按终止值从小到大排序。

后一个函数主要是使用 qsort()函数对数据集进行快速排序。

2.2.4　区域重叠判断并输出文件主函数

本部分通过程序第 529-548 行 region_overlap()函数实现。

本部分主要是创建并打开输出文件，对两个数据集分别进行一次标记，然后根据标记结果输出至指定文件，最后关闭文件流。

2.3 名字重叠判断功能及文件输出模块

2.3.1　字典树节点的创建及初始化

本部分通过程序第 587-596 行 create_node()函数实现。

首先为节点分配一个内存，然后将该节点的字符存入，并初始化其所有子节点为 NULL。最后返回该节点。

2.3.2　字典树节点的扩展

本部分通过程序第 599-609 行 append_node()函数实现。

首先判断该字符对应的子节点是否存在，如果不存在则用上一个函数创建节点，否则直接返回。返回值代表是否扩展了新的节点。

2.3.3　字典树增加单词

本部分通过程序第 612-634 行 add_word()函数实现。

对单词的每个字符依次查找，如果某个字符对应节点为 NULL，则代表该单词不存在，不断增加新的节点，并在最后一个节点将 flag 变为 true 代表单词的结束。

2.3.4　创建数据集字典树

本部分通过程序第 637-649 行 create_tree()函数实现。

对数据集的每一个名字进行读取并添加进字典树中，返回该字典树的根节点用来后续进行查找。

2.3.5　在字典树中查找名字是否重叠

本部分通过程序第 652-670 行 search_word 函数实现。

对一个名字的每个字符依次在字典树中查找，如果查找到某一节点不存在或到达了名字的最后一个字符，则退出循环。此时重叠分为两种情况，第一种是此时名字查找到了最后一个字符，说明字典树中存在一个单词与该名字完全相同；第二种是此时字典树已经查找到了单词的最后一个，只不过名字未查找完，这说明名字的前缀包含在字典树中，也是重叠关系。如果不是以上两种情况，则代表没有发生重叠。最后返回值代表是否发生重叠。

2.3.6　名字全部比对

本部分通过程序第 673-686 行 name_oversearch()函数实现。

通过对每一个数据的名字进行循环比对，判断每个名字是否在字典树中发生重叠，并根据返回值将数据输出到对应的文件中。

2.3.7　名字重叠判断并输出文件主函数

本部分通过程序第 689-702 行 name_overlap()函数实现。

本部分主要是创建并打开输出文件，对两个数据集分别进行名字全部比对，然后根据结果输出至指定文件，最后关闭文件流。

2.4 程序主函数

本部分通过程序第 189-239 行 main()函数实现。

本部分主要功能主要是计时和总体运行。使用的函数为前述各类综合函数，首先是 get_option()函数获取选项，然后 compose_data()函数对原始数据读取并处

理，之后根据不同选项分别进行对应的比对工作，其中-c 使用 sort_data()函数排序后 region_overlap()函数标记并输出；-n 使用 create_tree()函数生成数据集的字典树后 name_overlap()函数判断并输出。最后释放内存并输出程序总耗时。

2.5 错误提示类型及处理

2.5.1 参数数量控制

程序第 266-278 行实现。其中对于参数不足和超出分别进行提示，并建议使用-h 获得帮助，同时退出程序。

2.5.2 选项唯一控制

程序第 315-321 行实现。通过判断是否两个选项都被选择，来决定是否继续运行程序。如果同时被选择，则提示不能够同时选择并退出程序。

2.5.3 未知参数拒绝

程序第 301-305 行实现。提示输入了未定义的选项并建议获得帮助，同时退出程序。

2.5.4 路径缺失错误

程序第 322-333 行实现。通过判断调用选项后下一参数位置 optind 是否是总参数个数减一，来确定路径个数是否足够。如果不够则提示提供两个文件路径并退出程序。

2.5.5 范围输入控制

本错误类型分为两种。

第一种是未输入范围，在程序第 308-313 行实现，判断存入 file_column 的字符串长度是否为零，如果有一个不存在就提示范围未指定并退出程序。

第二种是范围与选项不相符，在程序第 359 和 368 行实现，通过第 338-348 行 coma_count()函数计算范围中的逗号个数，其中-c 有且只有一个逗号，-n 不能存在逗号。如果不合适则提示并退出。需要注意的是设置 file_col 第二个数字初始值为-2，如果格式不符合不执行赋值语句则数值不变，用来作为格式是否符合的判定。

2.5.6 空文件错误

程序第 398-402 行实现，在获得原始行数时如果为 1，则代表为空文件，提示空文件并退出程序。

2.5.7　文件不存在

程序第 471-473 行实现，如果文件不存在，则提示文件不存在并退出。

3　程序使用方法及编译

3.1 使用方法

本程序使用需要通过命令行实现，Linux/Unix/macOS 系统中在终端中运行，Windows 系统可用 Powershell 或 MinGW 运行。在两个系统中运行方式相同，以下不再区分。

在使用时，必须给出-a，-b 选项及其参数和两个输入文件路径。除此外由于默认模式为-c，因此可以选择给出模式选项-c 或-n 也可以不给出。但需要注意的是-c 和-n 不可同时使用。

各选项含义：

-c：范围重叠模式

-n：名字重叠模式

-a：指定第一个文件处理的列数

-b：指定第二个文件处理的列数

-h：获得程序帮助

命令行示例如下：

./Biodiff -a 3,4 -b 3,4 geneA.gtf geneB.gtf

./Biodiff -c -a 3,4 -b 3,4 geneA.gtf geneB.gtf

./Biodiff -n -a 0 -b 8 geneA.gtf geneB.gtf

3.2 编译过程

使用 gcc 编译即可，命令如下：

gcc -o Biodiff Biodiff.c

4　测试及结果

本部分为了获得测试结果，我写了一个 shell 脚本用来获得结果文件的行数及文件大小，命名为 view.sh，脚本代码如下：

```
wc -l ./result/A\&B_A.gtf
wc -l ./result/A-B.gtf
wc -l ./result/A\&B_B.gtf
wc -l ./result/B-A.gtf
ls -l ./result
```

4.1 无参数输入

```
[517111910078@mgt01 demo]$ ./Biodiff
Warning: the arguments you provide are not enough.
Options -a, -b must be chosen and two file paths must be provided.
If you want to get help, enter the following code:

        ./Biodiff -h
```

4.2 帮助提示

```
[517111910078@mgt01 demo]$ ./Biodiff -h
#-------------------------------------------------------------------------#
#                             Biodiff                                     #
#                       Author: Zhang Zhizhuo                             #
#     A bioinformatics program to compare two data files and output       #
#-------------------------------------------------------------------------#
# Program function:                                                       #
#   This program can find differences between two files containing        #
#bioinformatics data.                                                     #
#                                                                         #
#   If provided two files A (from-file) and B (to-file), program will     #
#generate all lines in A-B, A&B_A, A&B_B and B-A in terms of the range    #
#given.The file format of file A and B can be different.                  #
#                                                                         #
#   There will be two styles for comparison: one is coordinate based      #
#(option –c ) and the other is name based (option –n).The default style   #
#is option -c.                                                           #
#                                                                         #
#   The two styles were described as follows.                             #
#                                                                         #
#   1)Coordinate-based diff. Two or more columns from file A and B will   #
#be selected and compared to check if the two regions overlap. If two     #
#regions from the two files overlap, then these two regions will be put   #
#into to A&B_A and A&B_B; those regions in A but not in A&B will be put   #
#into A-B; and those in B but not in A&B will be put into B-A.            #
#                                                                         #
#   2)Name-based diff. Two columns from file A and B will be selected     #
#and compared in terms of string comparison. Users need to specify the    #
#column numbers in two files to be compared. If their names “overlap”,    #
#it should generate 4 result files corresponding to A&B_A, A&B_B, A-B,    #
#and B-A, where A&B_A contains those lines from file A and overlapping    #
#with some entries in file B;A&B_B contains lines from file B and over-   #
#lapping with entries in file A; A-B contains those lines from file A     #
#and with no overlapping entries in B; and B-A stands for those lines     #
#from file B but with no overlapping entries in A.                        #
#-------------------------------------------------------------------------#
```

```
# Usage:                                                              #
#   This program can be used by following code when execute:          #
#                                                                     #
#       ./Biodiff [options] from-file to-file                         #
#                                                                     #
#   In which,                                                         #
#       option             --- the option you choose.                 #
#                              See also OPTION in Reference           #
#       from-file, to-file --- the absolute or relative path to the two#
#                              files                                  #
#                                                                     #
# Reference:                                                          #
#    OPTION:                                                          #
#       -c output according to coordinate-based diff                  #
#       -n output according to name-based diff                        #
#       -a specify the criteria of from-file(following by the criteria)#
#       -b specify the criteria of to-file(following by the criteria)  #
#       -h get the help of this program                               #
#---------------------------------------------------------------------#
# Example:                                                            #
#   ./Biodiff -a 3,4 -b 3,4 geneA.gtf geneB.gtf                       #
#   ./Biodiff -c -a 3,4 -b 3,4 geneA.gtf geneB.gtf                    #
#   ./Biodiff -n -a 0 -b 8 geneA.gtf geneB.gtf                        #
#---------------------------------------------------------------------#
```

4.3 范围重叠模式

4.3.1　小文件对小文件

使用 POGK.gtf 和 hg19_chr1_repeatmasker_10m.gtf 两个文件。

```
[517111910078@mgt01 demo]$ ./Biodiff -c -a 3,4 -b 3,4 POGK.gtf hg19_chr1_repeatmasker_10m.gtf
Reading POGK.gtf
------------------Read over-------------------
Reading hg19_chr1_repeatmasker_10m.gtf
------------------Read over-------------------
Sorting POGK.gtf
------------------Sort over-------------------
Sorting hg19_chr1_repeatmasker_10m.gtf
------------------Sort over-------------------
------------------Write over------------------
The results are in /share/home/2017/517111910078/demo/result
Program time: 0.010000 s.
```

运行结果：

```
[517111910078@mgt01 demo]$ ./view.sh
0 ./result/A&B_A.gtf
470 ./result/A-B.gtf
0 ./result/A&B_B.gtf
17299 ./result/B-A.gtf
total 1716
-rw-r--r-- 1 517111910078 ug2017       0 Jun 13 12:59 A&B_A.gtf
-rw-r--r-- 1 517111910078 ug2017       0 Jun 13 12:59 A&B_B.gtf
-rw-r--r-- 1 517111910078 ug2017   47890 Jun 13 12:59 A-B.gtf
-rw-r--r-- 1 517111910078 ug2017 1705425 Jun 13 12:59 B-A.gtf
```

运行时间：0.01s

4.3.2　小文件对中文件

使用 hg19_chr1_repeatmasker_10m.gtf 和 VHC.gtf 两个文件。

```
[517111910078@mgt01 demo]$ ./Biodiff -c -a 3,4 -b 3,4 VHC.gtf hg19_chr1_
Reading VHC.gtf
------------------Read over-------------------
Reading hg19_chr1_repeatmasker_10m.gtf
------------------Read over-------------------
Sorting VHC.gtf
------------------Sort over-------------------
Sorting hg19_chr1_repeatmasker_10m.gtf
------------------Sort over-------------------
------------------Write over------------------
The results are in /share/home/2017/517111910078/demo/result
Program time: 0.430000 s.
```

运行结果：

```
[517111910078@mgt01 demo]$ ./view.sh
12439 ./result/A&B_A.gtf
315729 ./result/A-B.gtf
2052 ./result/A&B_B.gtf
15247 ./result/B-A.gtf
total 33016
-rw-r--r-- 1 517111910078 ug2017  1169534 Jun 13 13:10 A&B_A.gtf
-rw-r--r-- 1 517111910078 ug2017   202344 Jun 13 13:10 A&B_B.gtf
-rw-r--r-- 1 517111910078 ug2017 30926104 Jun 13 13:10 A-B.gtf
-rw-r--r-- 1 517111910078 ug2017  1503081 Jun 13 13:10 B-A.gtf
```

运行时间：0.43s

4.3.3　小文件对大文件

使用 hg19_chr1_repeatmasker_10m.gtf 和 A_big1.gtf 两个文件。

```
[517111910078@mgt01 demo]$ ./Biodiff -c -a 3,4 -b 3,4 A_big1.gtf hg19_chr
Reading A_big1.gtf
------------------Read over-------------------
Reading hg19_chr1_repeatmasker_10m.gtf
------------------Read over-------------------
Sorting A_big1.gtf
------------------Sort over-------------------
Sorting hg19_chr1_repeatmasker_10m.gtf
------------------Sort over-------------------
------------------Write over------------------
The results are in /share/home/2017/517111910078/demo/result
Program time: 5.260000 s.
```

运行结果：

```
[517111910078@mgt01 demo]$ ./view.sh
200240 ./result/A&B_A.gtf
4197671 ./result/A-B.gtf
9980 ./result/A&B_B.gtf
7319 ./result/B-A.gtf
total 428340
-rw-r--r-- 1 517111910078 ug2017  19018758 Jun 13 13:13 A&B_A.gtf
-rw-r--r-- 1 517111910078 ug2017    983988 Jun 13 13:13 A&B_B.gtf
-rw-r--r-- 1 517111910078 ug2017 417885073 Jun 13 13:13 A-B.gtf
-rw-r--r-- 1 517111910078 ug2017    721437 Jun 13 13:13 B-A.gtf
```

运行时间：5.26s

4.3.4　中文件对中文件

使用 VHC.gtf 和 VLC.gtf 两个文件。

```
[517111910078@mgt01 demo]$ ./Biodiff -c -a 3,4 -b 3,4 VHC.gtf VLC.gtf
Reading VHC.gtf
------------------Read over-------------------
Reading VLC.gtf
------------------Read over-------------------
Sorting VHC.gtf
------------------Sort over-------------------
Sorting VLC.gtf
------------------Sort over-------------------
------------------Write over------------------
The results are in /share/home/2017/517111910078/demo/result
Program time: 0.660000 s.
```

运行结果：

```
[517111910078@mgt01 demo]$ ./view.sh
201082 ./result/A&B_A.gtf
127086 ./result/A-B.gtf
112388 ./result/A&B_B.gtf
127082 ./result/B-A.gtf
total 54092
-rw-r--r-- 1 517111910078 ug2017 19636264 Jun 13 13:16 A&B_A.gtf
-rw-r--r-- 1 517111910078 ug2017 10893541 Jun 13 13:16 A&B_B.gtf
-rw-r--r-- 1 517111910078 ug2017 12459374 Jun 13 13:16 A-B.gtf
-rw-r--r-- 1 517111910078 ug2017 12391965 Jun 13 13:16 B-A.gtf
```

运行时间：0.66s

4.3.5　中文件对大文件

使用 VHC.gtf 和 A_big1.gtf 两个文件。

```
[517111910078@mgt01 demo]$ ./Biodiff -c -a 3,4 -b 3,4 VHC.gtf A_big1.gtf
Reading VHC.gtf
------------------Read over-------------------
Reading A_big1.gtf
------------------Read over-------------------
Sorting VHC.gtf
------------------Sort over-------------------
Sorting A_big1.gtf
------------------Sort over-------------------
------------------Write over------------------
The results are in /share/home/2017/517111910078/demo/result
Program time: 6.040000 s.
```

运行结果：

```
[517111910078@mgt01 demo]$ ./view.sh
328168 ./result/A&B_A.gtf
0 ./result/A-B.gtf
1595066 ./result/A&B_B.gtf
2802845 ./result/B-A.gtf
total 458012
-rw-r--r-- 1 517111910078 ug2017  32095638 Jun 13 13:18 A&B_A.gtf
-rw-r--r-- 1 517111910078 ug2017 158520635 Jun 13 13:18 A&B_B.gtf
-rw-r--r-- 1 517111910078 ug2017         0 Jun 13 13:18 A-B.gtf
-rw-r--r-- 1 517111910078 ug2017 278383196 Jun 13 13:18 B-A.gtf
```

运行时间：6.04s

4.3.6　大文件对大文件

使用 A_big1.gtf 自比对。

```
[517111910078@mgt01 demo]$ ./Biodiff -c -a 3,4 -b 3,4 A_big1.gtf A_big1.gtf
Reading A_big1.gtf
------------------Read over-------------------
Reading A_big1.gtf
------------------Read over-------------------
Sorting A_big1.gtf
------------------Sort over-------------------
Sorting A_big1.gtf
------------------Sort over-------------------
------------------Write over------------------
The results are in /share/home/2017/517111910078/demo/result
Program time: 17.660000 s.
```

运行结果：

```
[517111910078@mgt01 demo]$ ./view.sh
4397911 ./result/A&B_A.gtf
0 ./result/A-B.gtf
4397911 ./result/A&B_B.gtf
0 ./result/B-A.gtf
total 853328
-rw-r--r-- 1 517111910078 ug2017 436903831 Jun 13 13:21 A&B_A.gtf
-rw-r--r-- 1 517111910078 ug2017 436903831 Jun 13 13:21 A&B_B.gtf
-rw-r--r-- 1 517111910078 ug2017         0 Jun 13 13:20 A-B.gtf
-rw-r--r-- 1 517111910078 ug2017         0 Jun 13 13:20 B-A.gtf
```

运行时间：17.66s

4.4 名字重叠模式

4.4.1　小文件对小文件

使用 POGK.gtf 和 hg19_chr1_repeatmasker_10m.gtf 两个文件。

```
[517111910078@mgt01 demo]$ ./Biodiff -n -a 8 -b 8 POGK.gtf hg19_chr
Reading POGK.gtf
------------------Read over-------------------
Reading hg19_chr1_repeatmasker_10m.gtf
------------------Read over-------------------
Searching
------------------Search over-----------------
------------------Write over------------------
The results are in /share/home/2017/517111910078/demo/result
Program time: 0.190000 s.
```

运行结果：

```
[517111910078@mgt01 demo]$ ./view.sh
0 ./result/A&B_A.gtf
470 ./result/A-B.gtf
0 ./result/A&B_B.gtf
17299 ./result/B-A.gtf
total 1716
-rw-r--r-- 1 517111910078 ug2017       0 Jun 13 13:23 A&B_A.gtf
-rw-r--r-- 1 517111910078 ug2017       0 Jun 13 13:23 A&B_B.gtf
-rw-r--r-- 1 517111910078 ug2017   47890 Jun 13 13:23 A-B.gtf
-rw-r--r-- 1 517111910078 ug2017 1705425 Jun 13 13:23 B-A.gtf
```

运行时间：0.19s

4.4.2　小文件对中文件

使用 hg19_chr1_repeatmasker_10m.gtf 和 VHC.gtf 两个文件。

```
[517111910078@mgt01 demo]$ ./Biodiff -n -a 8 -b 8 VHC.gtf hg19_ch
Reading VHC.gtf
------------------Read over-------------------
Reading hg19_chr1_repeatmasker_10m.gtf
------------------Read over-------------------
Searching
------------------Search over-----------------
------------------Write over------------------
The results are in /share/home/2017/517111910078/demo/result
Program time: 0.350000 s.
```

运行结果：

```
[517111910078@mgt01 demo]$ ./view.sh
0 ./result/A&B_A.gtf
328168 ./result/A-B.gtf
0 ./result/A&B_B.gtf
17299 ./result/B-A.gtf
total 33012
-rw-r--r-- 1 517111910078 ug2017       0 Jun 13 13:24 A&B_A.gtf
-rw-r--r-- 1 517111910078 ug2017       0 Jun 13 13:24 A&B_B.gtf
-rw-r--r-- 1 517111910078 ug2017 32095638 Jun 13 13:24 A-B.gtf
-rw-r--r-- 1 517111910078 ug2017  1705425 Jun 13 13:24 B-A.gtf
```

运行时间：0.35s

4.4.3　小文件对大文件

使用 hg19_chr1_repeatmasker_10m.gtf 和 A_big1.gtf 两个文件。

```
[517111910078@mgt01 demo]$ ./Biodiff -n -a 8 -b 8 A_big1.gtf hg1
Reading A_big1.gtf
------------------Read over-------------------
Reading hg19_chr1_repeatmasker_10m.gtf
------------------Read over-------------------
Searching
------------------Search over-----------------
------------------Write over------------------
The results are in /share/home/2017/517111910078/demo/result
Program time: 4.470000 s.
```

运行结果：

```
[517111910078@mgt01 demo]$ ./view.sh
0 ./result/A&B_A.gtf
4397911 ./result/A-B.gtf
0 ./result/A&B_B.gtf
17299 ./result/B-A.gtf
total 428332
-rw-r--r-- 1 517111910078 ug2017         0 Jun 13 13:25 A&B_A.gtf
-rw-r--r-- 1 517111910078 ug2017         0 Jun 13 13:25 A&B_B.gtf
-rw-r--r-- 1 517111910078 ug2017 436903831 Jun 13 13:25 A-B.gtf
-rw-r--r-- 1 517111910078 ug2017   1705425 Jun 13 13:25 B-A.gtf
```

运行时间：4.47s

4.4.4　中文件对中文件

使用 VHC.gtf 和 VLC.gtf 两个文件。

```
[517111910078@mgt01 demo]$ ./Biodiff -n -a 8 -b 8 VHC.gtf VLC.gtf
Reading VHC.gtf
------------------Read over-------------------
Reading VLC.gtf
------------------Read over-------------------
Searching
------------------Search over-----------------
------------------Write over------------------
The results are in /share/home/2017/517111910078/demo/result
Program time: 0.620000 s.
```

运行结果：

```
[517111910078@mgt01 demo]$ ./view.sh
328168 ./result/A&B_A.gtf
0 ./result/A-B.gtf
239470 ./result/A&B_B.gtf
0 ./result/B-A.gtf
total 54084
-rw-r--r-- 1 517111910078 ug2017 32095638 Jun 13 13:27 A&B_A.gtf
-rw-r--r-- 1 517111910078 ug2017 23285506 Jun 13 13:27 A&B_B.gtf
-rw-r--r-- 1 517111910078 ug2017        0 Jun 13 13:27 A-B.gtf
-rw-r--r-- 1 517111910078 ug2017        0 Jun 13 13:27 B-A.gtf
```

运行时间：0.62s

4.4.5　中文件对大文件

使用 VHC.gtf 和 A_big1.gtf 两个文件。

```
[517111910078@mgt01 demo]$ ./Biodiff -n -a 8 -b 8 VHC.gtf A_big1.gtf
Reading VHC.gtf
------------------Read over-------------------
Reading A_big1.gtf
------------------Read over-------------------
Searching
------------------Search over-----------------
------------------Write over------------------
The results are in /share/home/2017/517111910078/demo/result
Program time: 5.050000 s.
```

运行结果：

```
[517111910078@mgt01 demo]$ ./view.sh
328168 ./result/A&B_A.gtf
0 ./result/A-B.gtf
4397911 ./result/A&B_B.gtf
0 ./result/B-A.gtf
total 458008
-rw-r--r-- 1 517111910078 ug2017  32095638 Jun 13 13:28 A&B_A.gtf
-rw-r--r-- 1 517111910078 ug2017 436903831 Jun 13 13:28 A&B_B.gtf
-rw-r--r-- 1 517111910078 ug2017         0 Jun 13 13:28 A-B.gtf
-rw-r--r-- 1 517111910078 ug2017         0 Jun 13 13:28 B-A.gtf
```

运行时间：5.05s

4.4.6　大文件对大文件

使用 A_big1.gtf 自比对。

```
[517111910078@mgt01 demo]$ ./Biodiff -n -a 8 -b 8 A_big1.gtf A_big1.gtf
Reading A_big1.gtf
------------------Read over-------------------
Reading A_big1.gtf
------------------Read over-------------------
Searching
------------------Search over-----------------
------------------Write over------------------
The results are in /share/home/2017/517111910078/demo/result
Program time: 9.510000 s.
```

运行结果：

```
[517111910078@mgt01 demo]$ ./view.sh
4397911 ./result/A&B_A.gtf
0 ./result/A-B.gtf
4397911 ./result/A&B_B.gtf
0 ./result/B-A.gtf
total 853328
-rw-r--r-- 1 517111910078 ug2017 436903831 Jun 13 13:29 A&B_A.gtf
-rw-r--r-- 1 517111910078 ug2017 436903831 Jun 13 13:29 A&B_B.gtf
-rw-r--r-- 1 517111910078 ug2017         0 Jun 13 13:29 A-B.gtf
-rw-r--r-- 1 517111910078 ug2017         0 Jun 13 13:29 B-A.gtf
```

运行时间：9.51s

4.5 空文件

使用 ZNF10.gtf 空文件。

```
[517111910078@mgt01 demo]$ ./Biodiff -n -a 8 -b 8 ZNF10.gtf A_big1.gtf
WARNING: You have provided a file with no data in it!
```

5 总结

5.1 速度

File size	Running time/s	
	Coordinate-based	Name-based
Small vs Small	0.01	0.19
Small vs Medium	0.43	0.35
Small vs Big	5.26	4.47
Medium vs Medium	0.66	0.62
Medium vs Big	6.04	5.05
Big vs Big	17.66	9.51

表 1 运行时间统计

由表中可以看到，经过算法优化，各种文件比对运行速度均较快，满足程序目标。

5.2 内存

因为程序调用内存存储大小是按照文件大小确定，所以对内存的使用效率还是可以接受的，在服务器上运行时通过 top 命令查看内存占用很小。

5.3 问题及解决

5.3.1 存储数据

在编写代码过程中，遇到的第一个主要问题是如何快速读取和存储所有数据，并减少无谓的内存使用。经过思考后，我选择了是结构体数组，第一个原因是这样可以申请所需的内存大小不像一开始预设大小或不断放大造成内存的浪费；第二个原因是后面快排时可以直接使用，更加方便；第三个原因是这样读取的速度也非常的快，不会拖慢程序总时间。

5.3.2 范围比对模式逻辑

在处理范围比对功能时，我思考了多种比对方式，经过不断推演和编写代码运行后，发现之前想的几种方法都存在逻辑上少数特例无法处理的问题。最后选择了两遍标记的方法，因为数据经过排序后是整齐的，通过条件控制 continue 和 break 后实际循环次数并不多，比对速度非常快。同时这样的方法避免了之前的部分特例归错文件的问题。

5.3.3　名字比对逻辑

名字比对为了提高速度，采用了字典树的算法。第一稿代码处理时，将重叠解释为了完全相同，但后来老师在项目要求中对名字重叠进行了解释，也就是包含关系。观察并了解了名字命名方式后，我发现名字是以 chr 开头（即第一列基因名）加上数字编号的，也就是说可以通过前缀判断包含，然后修改了搜索逻辑，完成了最终版的名字比对。

5.3.4　帮助信息

第一版的提示信息我并没有像现在规划边界，导致阅读时如果窗口太窄会有些混乱，加上边界后可以提示用户增宽窗口以适应。

5.3.5　错误提示信息

在编写代码的过程中，我不断发现有一些需要进行提示的错误输入，根据特点我分别进行了错误提示，避免程序意外关闭或处理出现错误却没有信息提示，导致用户不知道问题所在。

6　感谢

感谢韦朝春老师一学期以来精彩的课程授课，在这门课上我学到了很多书本以外的知识，对 C 语言编程有了更清晰的理解。在本次大作业的编写中锻炼了很多的能力，收获颇多。同时感谢助教一学期以来的指导和帮助，批改作业的认真负责。

7　独立作业保证

本次大作业与同学就一些逻辑、算法等问题进行了探讨，除此外所有代码均为个人书写而成。

8　代码托管声明

本次大作业所有代码、文件均已上传至个人 github，其中还包括 Windows 下的代码以及编译后的执行文件。

如需要下载，可以使用命令 git clone git@github.com:reckonzzz/Biodiff.git。

附录

（源代码）