splitBarcode软件使用手册

版本号	作者	日期	新增功能
0.1.0	赵福祥	2018.10.22	支持se和pe模式,支持单双barcode拆分
0.1.1	赵福祥	2018.10.25	优化速度;增加-n -m选项分别用于控制线程个数和内存最大值; 编译为静态可执行文件
0.1.2	赵福祥	2018.11.05	修改特性:当用户输入内存参数时,忽略对可用内存的判断
0.1.3	赵福祥	2018.11.05	修复bug: 拆分结果不一致问题; SequenceStat中barcode标识不一致问题
0.1.4	赵福祥	2019.03.22	修复bug:统计二义性reads异常
0.1.5	赵福祥	2019.04.17	修复bug:cycle数过长导致整型溢出
0.1.6	赵福祥	2019.04.24	修改特性:-r参数,更改双barcode反转顺序从整体反转到两个barcode单独反转

- splitBarcode软件使用手册
  - 软件简介
    - 简介
    - 参数说明
  - 使用示例
    - 数据准备
    - PE双barcode
    - PE单barcode
    - SE双Barcode
    - SE单barcode
  - 查看结果
    - Barcode拆分结果文件
    - Barcode序列统计文件
    - Fastq统计文件

软件简介

简介

软件名称:**splitBarcode**

最新版本:**0.1.6**

功能:读取指定的fastq文件并拆分barcode,支持se和pe模式,支持单barcode和双barcode,支持w indow s和linux平台

使用环境:w indow s 10或linux(centos 7.x),无需编译安装,直接拷贝对应版本的程序即可使用

使用说明: 命令行下运行 `splitBarcode`, 按下回车,可以看到程序提示:

```
$./bin/splitBarcode
USAGE:
  Version 0.1.6:

  cmd <barcode> <fq> [-OPTION]

OPTION:
  -2 FILE               Fastq for PE mode.[None]
  -o DIR                Output dir for decoded fastq.[None]
  -b INT INT INT        Barcode information : startCycle, length, mismatchNum.[Last 10 cycle with 1
 mismatch]
  -r                    Apply reverse complement of barcode sequence.[False]
  -n THREAD             Set the maximum thread numbers.[CPU number]
  -m MEMORY             Set the maximum memory(GB).[Available memory]

[ERROR] : The parameters number is not correct!
```

注意:linux系统运行命令需要可执行权限,可以通过 `chmod 755 splitBarcode` 命令赋予程序可执行权限

参数说明

本软件需要输入两个必须参数: , 如果只输入此两个参数,则默认se模式,拆分fastq文件的最后10bp,容错为1（默认只拆分单barcode,请确认输入正确的barcode文件）

另有可选参数:

- *-2* 为2链fastq文件,有此参数表示pe模式
- *-o <输出目录>* 保存拆分结果目录,默认为参数fastq文件所在目录
- *-b <起始位置 长度 容错>* 三个参数以空格分隔,表示一个barcode拆分的信息;如果使用双barcode,需要输入两个 -b 参数的信息；默认拆分最后10 bp,容错为1
- *-n <最大线程数>* 设置线程数,为同时处理压缩的最大线程个数,其值必须为大于0的整数,如20
- *-m <最大内存>* 设置内存上限,注意单位为GB,其值必须为大于1的整数或浮点数,如100
- *-r* 此可选参数表示对barcode序列进行反向互补,需要用户根据实际测序情况进行选择,默认不进行反向互补

另:如果设置线程数或内存,程序会参考进行内存的分配；如果没有设置,程序会根据实际可用内存和CPU个数来进行内存的自动分配

使用示例

数据准备

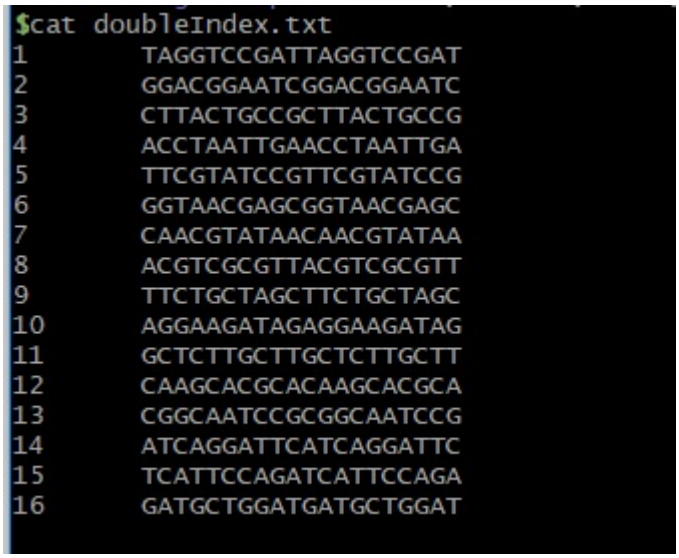
现假设有数据如下:

```
-rw-r--r-- 1 bcbuild ST_BI 377 2019/02/28 09:14 doubleIndex.txt
-rw-r--r-- 1 bcbuild ST_BI 217 2019/02/28 09:16 singleIndex.txt
-rwxr-xr-x 1 bcbuild ST_BI 11M 2018/11/05 21:49 splitBarcode
-rw-r--r-- 1 bcbuild ST_BI 62M 2019/02/28 09:14 test_1.fq.gz
-rw-r--r-- 1 bcbuild ST_BI 63M 2019/02/28 09:14 test_2.fq.gz
```

其中,

- **test_1.fq.gz** 是pe测序中的一链数据,长度为100bp
- **test_2.fq.gz** 是pe测序中的二链数据,长度为120bp(包含barcode长度20bp)
- **splitBarcode** 为可执行程序
- **singleIndex.txt** 是由两列数据组成的用来拆分的文本文件,第一列是barcode的ID,第二列是barcode序列,这里长度是10bp,两列之间用空格或Tab分隔,其格式如下图:

- **doubleIndex.txt** 与singleIndex.txt类似, 不同之处是第二列数据barcode长度是20bp,其格式如下图:



PE双barcode

PE100双barcode拆分,一链长度100,二链长度100,两个barcode在二链最后20bp
输入命令:

```
./splitBarcode doubleIndex.txt test_1.fq.gz -2 test_2.fq.gz -o result_pe_double_index -b 200 10 1 -b 210 10 1 -r
```

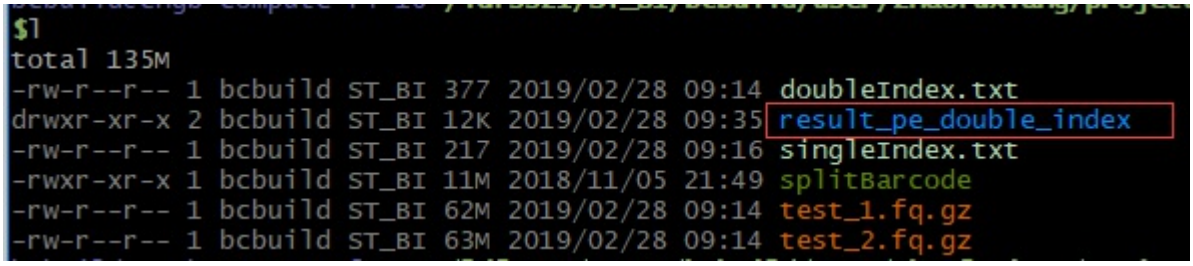
命令解析:

- *./splitBarcode* 程序名称
- *doubleIndex.txt* 第一个必需参数, 输入的barcode文件
- *test_1.fq.gz* 第二个必需参数, 一链的fastq文件
- *-2 test_2.fq.gz* 可选参数, 指定二链的fastq文件
- *-o result_pe_double_index* 可选参数, 指定拆分后结果的输出目录
- *-b 200 10 1 -b 210 10 1* 可选参数, 指定barcode拆分信息, 每次三个整数的组合分别表示 barcode起始位置,长度,和容错;此示例表示共两个barcode, 分别是起始位置200bp,长度10,容错1和起始位置210bp,长度10,容错1
- *-r* 表示barcode序列需要反向互补, 一般barcode序列在二链末尾都需要此参数, 具体是否需要反向互补由用户确定

运行过程如下图:

```
$./splitBarcode doubleIndex.txt test_1.fq.gz -2 test_2.fq.gz -o result_pe_double_index -b 200 10 1 -b 210 10 1 -r
Use PE mode
system totalPhysMem: 125(Gb) validPhysMem 103(Gb)
available memory: 103(GB)
using thread number: 48
read fastq file1 time(s): 2.59832
read fastq file2 time(s): 2.78502
go to endFastq
finish write fq total time(s): 79.3504
```

运行结果如下图:



可以看到当前路径下出现了一个新的目录result_pe_double_index, 存储此次拆分的结果

PE单barcode

一链长度100,二链长度110,单barcode在二链最后10bp
输入命令为:

```
./splitBarcode singleIndex.txt test_1.fq.gz -2 test_2.fq.gz -o result_pe_single_index -b 210 10 1 -r
```

命令解析:

- *./splitBarcode* 程序名称
- *singleIndex.txt* 第一个必需参数, 输入的barcode文件
- *test_1.fq.gz* 第二个必需参数, 一链的fastq文件
- *-2 test_2.fq.gz* 可选参数, 指定二链的fastq文件
- *-o result_pe_single_index* 可选参数, 指定拆分后结果的输出目录
- *-b 210 10 1* 可选参数, 指定barcode拆分信息, 每次三个整数的组合分别表示 barcode起始位置,长度,和容错;此示例表示共一个barcode,起始位置210bp,长度10,容错1
- *-r* 表示barcode序列需要反向互补, 一般barcode序列在二链末尾都需要此参数, 具体是否需要反向互补由用户确定

运行过程如下图:

```
$./splitBarcode singleIndex.txt test_1.fq.gz -2 test_2.fq.gz -o result_pe_single_index -b 210 10 1 -r
Use PE mode
system totalPhysMem: 125(Gb) validPhysMem 103(Gb)
available memory: 103(GB)
using thread number: 48
read fastq file1 time(s): 2.38202
read fastq file2 time(s): 2.50791
go to endFastq
finish write fq total time(s): 80.9929
```

结果保存在目录result_pe_single_index

SE双Barcode

一链长度100,双barcode在最后20bp
输入命令为:

```
./splitBarcode doubleIndex.txt test_2.fq.gz -o result_se_double_index -b 100 10 1 -b 110 10 1 -r
```

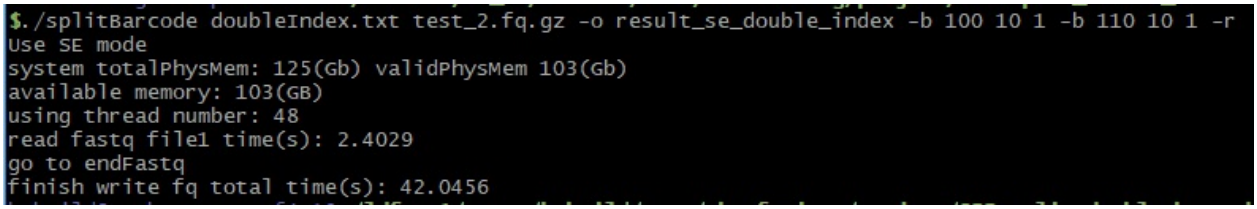
命令解析:

- *./splitBarcode* 程序名称
- *doubleIndex.txt* 第一个必需参数, 输入的barcode文件
- *test_2.fq.gz* 第二个必需参数, 一链的fastq文件(测试需要,因此使用包含barcode序列的二链当作一链)
- *-o result_se_double_index* 可选参数, 指定拆分后结果的输出目录
- *-b 100 10 1 -b 110 10 1* 可选参数, 指定barcode拆分信息, 每次三个整数的组合分别表示 barcode起始位置,长度,和容错;此示例表示共两个barcode, 分别是起始位置100bp,长度10,容错1和起始位置110bp,长度10,容错1

- *-r* 表示barcode序列需要反向互补, 一般barcode序列在二链末尾都需要此参数, 具体是否需要反向互补由用户确定

注:正常se不需要反转barcode序列,此次反转是因为使用二链数据来模拟se测序

运行过程如下图:



结果保存在目录result_se_double_index

SE单barcode

一链长度110,单barcode在最后10bp
输入命令为:

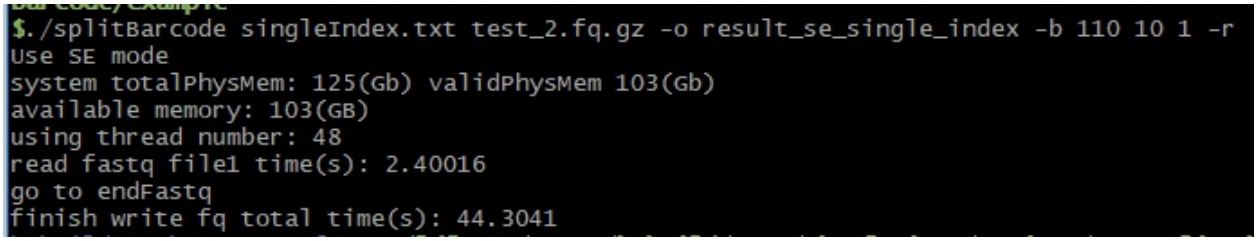
```
./splitBarcode singleIndex.txt test_2.fq.gz -o result_se_single_index -b 110 10 1 -r
```

命令解析:

- *./splitBarcode* 程序名称
- *singleIndex.txt* 第一个必需参数, 输入的barcode文件
- *test_2.fq.gz* 第二个必需参数, 一链的fastq文件(测试需要,因此使用包含barcode序列的二链当作一链来使用)
- *-o result_se_single_index* 可选参数, 指定拆分后结果的输出目录
- *-b 110 10 1* 可选参数, 指定barcode拆分信息, 每次三个整数的组合分别表示 barcode起始位置,长度,和容错;此示例表示共一个barcode,起始位置110bp,长度10,容错1
- *-r* 表示barcode序列需要反向互补, 一般barcode序列在二链末尾都需要此参数, 具体是否需要反向互补由用户确定

注:正常se不需要反转barcode序列,此次反转是因为使用二链数据来模拟se测序

运行过程如下图:



结果保存在目录result_se_single_index

查看结果

运行结束后可以在用户指定的输出目录中查看结果,结果包括拆分后的fastq文件,fastq 统计文件,barcode 拆分结果等文件.

以PE双barcode为例说明.
查看result_pe_double_index目录, 确认存在拆分后的fastq文件和barcode拆分结果文件:

```
$l result_pe_double_index/
total 121M
-rw-r--r-- 1 bcbuild ST_BI  710 2019/02/28 09:35 BarcodeStat.txt
-rw-r--r-- 1 bcbuild ST_BI 424K 2019/02/28 09:35 SequenceStat.txt
-rw-r--r-- 1 bcbuild ST_BI  19K 2019/02/28 09:35 V100003766_L01_10_1.fq.fqStat.txt
-rw-r--r-- 1 bcbuild ST_BI 4.9M 2019/02/28 09:35 V100003766_L01_10_1.fq.gz
-rw-r--r-- 1 bcbuild ST_BI  19K 2019/02/28 09:35 V100003766_L01_10_2.fq.fqStat.txt
-rw-r--r-- 1 bcbuild ST_BI 4.4M 2019/02/28 09:35 V100003766_L01_10_2.fq.gz
-rw-r--r-- 1 bcbuild ST_BI  19K 2019/02/28 09:35 V100003766_L01_11_1.fq.fqStat.txt
-rw-r--r-- 1 bcbuild ST_BI 3.8M 2019/02/28 09:35 V100003766_L01_11_1.fq.gz
-rw-r--r-- 1 bcbuild ST_BI  18K 2019/02/28 09:35 V100003766_L01_11_2.fq.fqStat.txt
-rw-r--r-- 1 bcbuild ST_BI 3.5M 2019/02/28 09:35 V100003766_L01_11_2.fq.gz
-rw-r--r-- 1 bcbuild ST_BI  19K 2019/02/28 09:35 V100003766_L01_1_1.fq.fqStat.txt
-rw-r--r-- 1 bcbuild ST_BI 4.8M 2019/02/28 09:35 V100003766_L01_1_1.fq.gz
-rw-r--r-- 1 bcbuild ST_BI  18K 2019/02/28 09:35 V100003766_L01_12_1.fq.fqStat.txt
-rw-r--r-- 1 bcbuild ST_BI 2.1M 2019/02/28 09:35 V100003766_L01_12_1.fq.gz
-rw-r--r-- 1 bcbuild ST_BI  17K 2019/02/28 09:35 V100003766_L01_12_2.fq.fqStat.txt
-rw-r--r-- 1 bcbuild ST_BI 1.9M 2019/02/28 09:35 V100003766_L01_12_2.fq.gz
-rw-r--r-- 1 bcbuild ST_BI  19K 2019/02/28 09:35 V100003766_L01_1_2.fq.fqStat.txt
-rw-r--r-- 1 bcbuild ST_BI 4.3M 2019/02/28 09:35 V100003766_L01_1_2.fq.gz
-rw-r--r-- 1 bcbuild ST_BI  19K 2019/02/28 09:35 V100003766_L01_13_1.fq.fqStat.txt
-rw-r--r-- 1 bcbuild ST_BI 4.1M 2019/02/28 09:35 V100003766_L01_13_1.fq.gz
-rw-r--r-- 1 bcbuild ST_BI  18K 2019/02/28 09:35 V100003766_L01_13_2.fq.fqStat.txt
-rw-r--r-- 1 bcbuild ST_BI 3.7M 2019/02/28 09:35 V100003766_L01_13_2.fq.gz
-rw-r--r-- 1 bcbuild ST_BI  19K 2019/02/28 09:35 V100003766_L01_14_1.fq.fqStat.txt
-rw-r--r-- 1 bcbuild ST_BI 3.7M 2019/02/28 09:35 V100003766_L01_14_1.fq.gz
```

Barcode拆分结果文件

文件名:**BarcodeStat.txt**
记录barcode拆分结果,拆分率是拆分程序的一个重要结果指标.

此文件共五列:

1. barcode编号
2. 无错拆分出来的reads个数
3. 有容错拆分出来的reads个数
4. 全部拆分出来的reads个数(即2列与3列之和)
5. 全部拆分出来的reads个数占总reads的百分比

最后一行为所有拆分出来的barcode统计的总和
可以看出总拆分率约95.19%
注:二义性reads认为没有拆分出来,故不计入此拆分文件

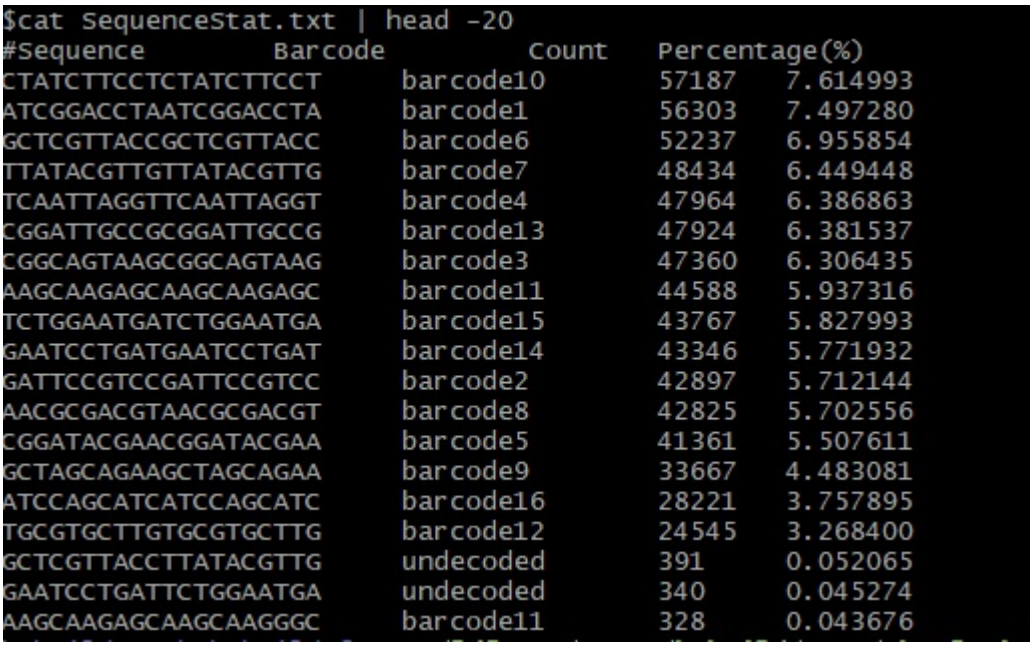
```
$cat BarcodeStat.txt
#Barcode        Correct         Corrected       Total   Percentage(%)
barcode1        56303   1082    57385   7.641359
barcode2        42897   862     43759   5.826927
barcode3        47360   778     48138   6.410033
barcode4        47964   535     48499   6.458103
barcode5        41361   675     42036   5.597493
barcode6        52237   794     53031   7.061582
barcode7        48434   519     48953   6.518558
barcode8        42825   906     43731   5.823199
barcode9        33667   434     34101   4.540873
barcode10       57187   859     58046   7.729377
barcode11       44588   1090    45678   6.082460
barcode12       24545   501     25046   3.335113
barcode13       47924   1128    49052   6.531741
barcode14       43346   658     44004   5.859551
barcode15       43767   1012    44779   5.962750
barcode16       28221   428     28649   3.814887
Total   702626  12261   714887  95.194008
```

Barcode序列统计文件

文件名:**SequenceStat.txt**
记录所有出现在barcode位置的序列统计信息;如果拆分率异常,可以通过检查此文件获取出现频率最高的序列,以分析是否给错barcode序列或者忘记了反转互补.

此文件共分四列:

1. barcode序列信息
2. 对应的barcode编号
3. barcode序列出现次数
4. barcode序列个数占总reads的百分比



```
$cat SequenceStat.txt | head -20
#Sequence        Barcode         Count   Percentage(%)
CTATCTTCCTCTATCTTCCT     barcode10       57187   7.614993
ATCGGACCTAATCGGACCTA     barcode1        56303   7.497280
GCTCGTTACCGCTCGTTACC     barcode6        52237   6.955854
TTATACGTTGTTATACGTTG     barcode7        48434   6.449448
TCAATTAGGTTCAATTAGGT     barcode4        47964   6.386863
CGGATTGCCGCGGATTGCCG     barcode13       47924   6.381537
CGGCAGTAAGCGGCAGTAAG     barcode3        47360   6.306435
AAGCAAGAGCAAGCAAGAGC     barcode11       44588   5.937316
TCTGGAATGATCTGGAATGA     barcode15       43767   5.827993
GAATCCTGATGAATCCTGAT     barcode14       43346   5.771932
GATTCCGTCCGATTCCGTCC     barcode2        42897   5.712144
AACGCGACGTAACGCGACGT     barcode8        42825   5.702556
CGGATACGAACGGATACGAA     barcode5        41361   5.507611
GCTAGCAGAAGCTAGCAGAA     barcode9        33667   4.483081
ATCCAGCATCATCCAGCATC     barcode16       28221   3.757895
TGCGTGCTTGTGCGTGCTTG     barcode12       24545   3.268400
GCTCGTTACCTTATACGTTG     undecoded       391     0.052065
GAATCCTGATTCTGGAATGA     undecoded       340     0.045274
AAGCAAGAGCAAGCAAGGGC     barcode11       328     0.043676
```

Fastq统计文件

文件名形如:**slide_lane_barcodeID.fq.fqStat.txt**
此文件包含对应barcode拆分结果的统计信息,如reads个数,GC含量,Q30,base个数等指标.
每一个barcodeID对应生成一个fastq统计文件.

#Name	rr/V100003766_L01_undecoded_1.fq.gz
#PhredQual	33
#ReadNum	36092
#row_readLen	100
#col	48
#BaseNum	3609200
#N_Count	598	0.016569
#GC%	42.23
#Q10%	99.34
#Q20%	96.75
#Q30%	86.44
#EstErr%	0.297012

#Pos	A	C	G	T	N	0	1	2	3	4	5	6	7	8
1	8521	9242	9186	9113	30	30	0	0	0	0	17	36	42	33
2	9233	8782	8931	9146	0	0	0	0	0	0	62	56	56	61
3	11315	8401	6405	9971	0	0	0	0	0	0	18	44	40	52
4	9096	8231	7777	10988	0	0	0	0	0	0	27	39	37	48
5	1761	9379	9782	15169	1	1	0	0	0	0	347	611	594	513
6	3481	5271	3354	23859	127	127	0	0	0	23	165	156	151	131
7	8023	10493	8628	8948	0	0	0	0	0	3	71	130	125	121
8	11663	9156	7383	7890	0	0	0	0	0	0	17	45	37	37
9	11957	7179	6909	10046	1	1	0	0	0	0	20	26	20	17
10	12146	6568	8119	9259	0	0	0	0	0	1	23	32	50	53
11	12742	6230	7898	9222	0	0	0	0	0	0	37	72	60	75
12	12485	6651	7141	9815	0	0	0	0	0	0	30	46	53	58
13	11630	6850	6657	10955	0	0	0	0	0	0	26	29	36	45
14	10686	7129	7039	11235	3	3	0	0	0	0	13	17	22	28
15	10122	7628	7457	10885	0	0	0	0	0	0	14	17	24	28
16	10344	7506	8120	10120	2	2	0	0	0	0	15	34	39	41
17	10437	7374	8443	9838	0	0	0	0	0	0	31	63	43	48
18	10935	7094	7974	10089	0	0	0	0	0	0	30	52	45	51
19	11000	6984	8025	10045	38	38	0	0	0	0	24	55	51	51
20	10771	7169	8096	10056	0	0	0	0	0	0	73	118	91	105
21	10801	7550	7674	10067	0	0	0	0	0	0	23	34	42	29
22	10773	7744	7351	10224	0	0	0	0	0	1	13	23	24	29
23	10989	7514	7534	10055	0	0	0	0	0	0	48	70	60	69
24	11137	7155	7578	10221	1	1	0	0	0	0	21	19	34	22
25	10771	7269	7804	10247	1	1	0	0	0	0	7	20	23	23
26	10475	7431	7993	10193	0	0	0	0	0	0	30	36	36	46
27	10469	7577	7855	10166	25	25	0	0	0	0	12	27	31	33
28	10588	7161	7876	10432	35	35	0	0	0	0	16	17	36	29
29	10405	7321	7509	10856	1	1	0	0	0	0	14	22	29	23
30	10449	7531	7314	10798	0	0	0	0	0	0	26	37	40	43