kmer_scan

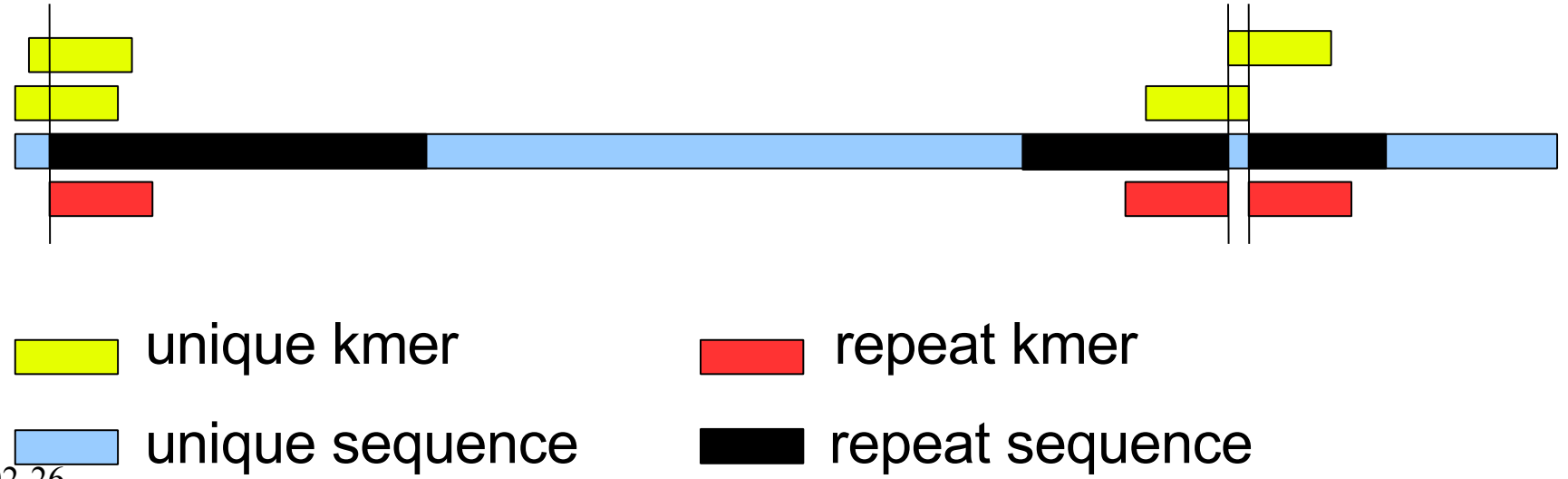
—李振宇、樊伟

大纲

- unique sequence 和 repeat sequence 的定义
- 程序的功能
- 程序的实现
- 程序的性能
- 程序的使用

unique sequence 和 repeat sequence 的定义

- unique sequence： 由连续的 unique 的 kmers 构成的序列
- repeat sequence： 由连续的 repeat 的 kmers 构成的序列
- 两者的边界确定：以 repeat sequence 的边界来划分



程序的功能

- 将基因组序列中的 unique sequence 和 repeat sequence 分别提取出来，输出到 *.unique 和 *.repeat 文件
- 不同频数的 kmer 的个数，输出到 *.kmer
- 统计相关信息

参考序列的数量、总长、不同的 kmer 的个数、 unique sequence 的数量和 repeat sequence 的数量、长度、最长的 unique sequence 和 repeat sequence 的长度

程序的实现

- Algorithm： 采用 hash 表存储。每个 hash 元素包含 kmer 对应的二进制数和 kmer 出现的频数。 hash 表的大小可动态增长。
- hash 元素：

```
typedef  struct Entity{
  uint64_t high; // 高 8 位存频数
  uint64_t low;
  }
```

kmer<=60， 频数 <=255

程序的性能

- 内存：主要由 reference 含有的不同的 kmer 的数量决定。

Memory = kmer_sum/load_factor*Entity_size

= kmer_sum/0.75*16(bytes) （默认参数）

- 时间：主要由 reference 的大小决定
- 初始 hash 表的大小的设置会对内存和时间产生较大的影响，建议手动设置。

以 119M 的 reference 为例：默认参数和设置合适的 hash 表大小的情况下，需要的内存和时间分别为 4.5G 、 6.5minutes 和 2.6G 、 4.5minutes

程序的使用

- 路径:

ifs1/GAG/assemble/lizhenyu/assenbly/kmer/bin/kmer_scan_v3/

- 输入： *.fa
- 输出： *.kmer 、 *.unique 、 *.repeat 、 *.log

程序的使用

- 用法:

/path/kmer_scan [option] *.fa

-k<int> Kmer length(<=60), default=50

-m<int> initial size of hash table, default=1024*1024

-l<float> load factor of hash table, default=0.75

-h usage information

- 例:

/path/kmer_scan ./test.fa

/path/kmer_scan -k 55 -m 16700000 ./test.fa

谢谢大家！

2010-02-26