哈尔滨工业大学计算机科学与技术学院

实验报告

课程名称：生物信息学

课程类型：专业课

实验项目名称：

大规模基因组序列表示

与索引系统设计与实现

实验题目：基因组序列建立自索引表示

班级：1803401

学号：1180300516

姓名：梁兆瀛

|  |  |  |
| --- | --- | --- |
| 设计成绩 | 报告成绩 | 指导老师 |
|  |  |  |

1. 实验目的

给定基因组序列（大肠杆菌E. Coli, 长度~5Mbp）作为输入，建立算法和系统，可以为基因组序列建立自索引表示

自索引结构可以选择CSA或者BWT

1. 实验原理

采用BWT算法：

（1）首先将原字符串S开头添加一个结束字符，一般取$，得到S'。

（2）将S'往右轮转，得到一个数组M。

（3）对这个数组按照字典序排序，得到 M'。

（4）存储最后一列

1. 测试结果及分析

空间消耗较大，目前只能实现小规模数据的排序

1. 经验体会

在大规模数据下，很多消耗空间大的算法是不能实现的，必须采用其他算法实现。一般复杂度为O（n）级别。

1. 附录：源代码（带注释）

#include <iostream>

#include <string>

#include <algorithm>

#include <string.h>

using namespace std;

int preSum[10000000];

char v[10000000];

//编码，生成last数组

int getLastArray(char \*lastArray,const string &str){ ///子串排序

int len=str.size();

string array[len];

for(int i=0;i<len;i++){

array[i] = str.substr(i);

}

sort(array,array+len);

for(int i=0;i<len;i++){

lastArray[i] = str.at((2\*len-array[i].size()-1)%len);

}

return 0;

}

int getCountPreSum(int \*preSum,const string &str){

int len=str.size();

memset(preSum,0,len\*sizeof(int));

for(int i=0;i<str.size();i++){

if(str.at(i) == '#')

preSum[0]++;

else

preSum[str.at(i)-'A'+1]++;

}

for(int i=1;i<len - 1 ;i++)

preSum[i] += preSum[i-1];

return 0;

}

//解码，使用last数组，恢复原来的文本块

int regainTextFromLastArray(char \*lastArray,char \*reGainStr,int \*preSum){

int len=strlen(lastArray);

int pos=0;

char c;

for(int i=len-1;i>=0;){

reGainStr[i] = lastArray[pos];

c = lastArray[pos];

pos = preSum[c-'A']+count(lastArray,lastArray+pos,c);

i--;

}

return 0;

}

int main (){

char c;

int i = 0;

FILE \*fp;//文件指针

fp = fopen("test.txt","r");//以文本方式打开文件。

if(fp == NULL) //打开文件出错。

return -1;

while((c = getc(fp))!=EOF)

{

if(c != '\n')

{

v[i] = c;

i++;

}

}

v[i] = '#';

string str = v;

int len=str.size();

char \*lastArray = new char[len+1];

char \*reGainStr = new char[len+1];

lastArray[len]='\0';

reGainStr[len]='\0';

getCountPreSum(preSum,str);

getLastArray(lastArray,str);

regainTextFromLastArray(lastArray,reGainStr,preSum);

cout<<" str: "<<str<<endl;

cout<<"lastArray : "<<lastArray<<endl;

cout<<"reGainStr : "<<reGainStr<<endl;

delete lastArray;

delete reGainStr;

return 0;

}