第七组练习6报告

作业基本要求

1. 以课程大作业分组为单位
2. 下载GENBANK格式的基因组序列文件，从中提取CDS序列，保存为FASTA格式的文件
3. 整个项目至少包含两个动态共享库库-libgenbank.so以及libfasta.so ，包含文件解读和格式的数据结构
4. 项目必须有完整的Makefile链
5. 项目必须包含测试数据和测试程序，用make test运行
6. 项目开发过程中建立dev分支和每个开发者对应的分支，必须能查询到每个开发者的开发合并路径
7. 项目最后合并到master分支并最终发布到github或者gitee的仓库中，每个开发者的账户中最后都包含一个拷贝
8. 报告必须包含核心代码的解析和测试数据的运行测试
9. 最好写gdb调试过程中怎么debug的

libgenbank库

定义CDS结构体

确定需要提取的信息：基因名、蛋白id、蛋白序列等

```
typedef struct CDS{
    char locus_tag[MAXWORD];//基因名
    char protein_id[MAXWORD];

    char *translation;//蛋白序列

    char left[MAXWORD] ;
    char right[MAXWORD];

    int end;//5表示5'部分，3表示3'部分，0表示既不是5‘也不是3’

    int flag_complement;//0表示正常，1表示在互补链上。

} CDS;
```

定义函数

1. 统计genbank文件中CDS出现次数

```
int countCDS (FILE *fp)//统计一共多少个CDS要提取
{
    int len;
    char line[MAXLINE];
    int count_CDS = 0;
    while (fgets(line, MAXLINE, fp) != NULL){
        len = strlen(line);
        int i=0;
        while(isspace(line[i]))//跳过开头空格
            i++;
```

```
        if (line[i]=='C' && line[i+1]=='D' && line[i+2]=='S' &&
isspace(line[i+3])) //判断是否为CDS
            {
                count_CDS++;
                continue;
            }
    }

    return count_CDS;
}
```

2. 读取CDS中所需信息（用于写在fasta文件中第一行的注释信息中）

```

void read_CDS(CDS cdsdata[], char *filename, int *CDS_lines){
    FILE *fp;
    fp = fopen(filename,"r");
    int len,CDS_line=0,nlines = 0;
    int count = 0 ;
    cdsdata[count].locus_tag[0] = '\0';//初始化结构体内的值
    cdsdata[count].protein_id[0] = '\0';
    char line[MAXLINE];
    while (fgets(line, MAXLINE, fp) != NULL){
        len = strlen(line);
        nlines++;
        int i=0;
        while(isspace(line[i]))//跳过开头空格
            i++;
        if (line[i]=='C' && line[i+1]=='D' && line[i+2]=='S' &&
isspace(line[i+3]))//判断是否为CDS
            {
                CDS_line = nlines;
                i+=3;

                if (strcmpr(line, "complement"))
                    cdsdata[count].flag_complement = 1;
                else cdsdata[count].flag_complement = 0;

                while(!('0'<=line[i]&&line[i]<='9')){//跳过非数字
                    if(line[i]=='<')
                        cdsdata[count].end = 5;
                    else if(line[i]=='>')
                        cdsdata[count].end = 3;
                    else cdsdata[count].end = 0;

                    i++;
                }
                int j = 0;
                while('0'<=line[i]&&line[i]<='9')
                        cdsdata[count].left[j++] = line[i++];//基因起始位
置
                    cdsdata[count].left[j]='\0';
                i+=2;

                int k = 0;
                while('0'<=line[i]&&line[i]<='9')
                    cdsdata[count].right[k++] = line[i++];//基因终止位置
```

```
                cdsdata[count].right[k]='\0';

                count++;
                continue;

            }

            if(strcmpr(line,"/locus_tag=") && (nlines == CDS_line+1) )
            {
                int k=0;
                while(line[k]!='"')
                    k++;
                k++;
                int j = 0;
                while(line[k]!='"')
                    cdsdata[count-1].locus_tag[j++]=line[k++];
                cdsdata[count-1].locus_tag[j]='\0';
            }

            if(strcmpr(line,"/protein_id="))
            {
                int k=0;
                while(line[k]!='"')
                    k++;
                k++;
                int j = 0;
                while(line[k]!='"')
                    cdsdata[count-1].protein_id[j++]=line[k++];
                cdsdata[count-1].protein_id[j]='\0';

            }

            if(strcmpr(line,"/translation="))
                CDS_lines[count -1]=nlines;//把translation行号存在数组里


    }
}
```

3. 提取CDS中translation序列（用于fasta文件的第二行序列）

```
void read_translation (FILE * fp,int count,CDS *cdsdata,char *line)
{
    char Line[MAXLINE];
    int right = atoi(cdsdata[count-1].right);
    int left = atoi(cdsdata[count-1].left);
    char *p,temp[right-left];
    p = malloc(right-left);
    int j=0,k = 0;
    while(line[k]!='"') //跳到序列处
        k++;
    k++;
    while(line[k] != '\0') //存储第一行
        temp[j++]=line[k++];

  while(fgets(Line, MAXLINE, fp) != NULL){
    int l = 0;
```

```
    while (isspace(Line[l]))//跳过空格
        l++;
    while (Line[l]!='\0'){
        temp[j++]=Line[l++];
        if (Line[l] == '"'){
            temp[j] = '\0';
            break;}
        }
    if (Line[l] == '"'){
        temp[j] = '\0';
        break;
        }
    }
    strcpy(p,temp);
    cdsdata[count-1].translation = p;
    printf("translation:%s\n\n",cdsdata[count-1].translation);
    free(p);
}
```

libfasta库

writefasta函数

将提取的信息输入到fasta格式文件中

```
void writefasta(char *filename,CDS cdsdata[],int count){
    int i;
    FILE *fp;
    strtok(filename,".");
    strcat(filename,".fasta");
    if((fp = fopen(filename,"w")) == NULL)
    {
        printf("Error:The file is not exist.");
        exit(1);
    }

    for(i=0;i<=count-1;i++)
    {
        fputs(">",fp);
        if(cdsdata[i].locus_tag != '\0'){
            fputs("locus_tag:",fp);
            fputs(cdsdata[i].locus_tag,fp);
        }
        if(cdsdata[i].protein_id != '\0'){
            fputs("\tprotein_id:",fp);
            fputs(cdsdata[i].protein_id,fp);
        }
        if(cdsdata[i].end != 0){
            if(cdsdata[i].end == 3)
                fputs("\tend:3'",fp);
            if(cdsdata[i].end == 5)
                fputs("\tend:5'",fp);
        }
        if(cdsdata[i].flag_complement == 0)
            fputs("\tnot complentmentary chain",fp);
        if(cdsdata[i].flag_complement == 1)
```

```
            fputs("\tcomplementary chain",fp);

        fputs("\n",fp);
        fputs(cdsdata[i].translation,fp);
        fputs("\n",fp);
    }

    fclose(fp);
}
```

main函数

实现程序功能：genbank格式向fasta格式转化

```
int main()
{
    char filename[100];
    printf("please input your filename(path):\n");
    scanf("%s",filename);
    FILE *fp;
    fp = fopen(filename,"r");
    printf("open\n");

    int count;
    count = countCDS(fp);//统计共多少CDS
    fclose(fp);
    printf("count:%d\n",count);

    int CDS_translation_lines[count];
    CDS cdsdata[count];
    read_CDS(cdsdata,filename,CDS_translation_lines);//把除translation以外的信息提取并存储
    printf("read\n");

    int i=0;//结果检查
    for(i=0;i<count;i++)
    {

 printf("%d.locus_tag:%s\tprotein_id:%s\tleft:%s\tright:%s\tend:%d\tflag:%d\n",i+1,cdsdata[i].locus_tag,cdsdata[i].protein_id,cdsdata[i].left,cdsdata[i].right,cdsdata[i].end,cdsdata[i].flag_complement);
    }

    for (i=0;i<count;i++)//提取并存储translation
    {
        FILE *fp2;
        char line2[MAXLINE];
        fp2 = fopen(filename,"r");
        int lines = 0;
        while(fgets(line2, MAXLINE, fp2) != NULL)
        {
            //printf("%s\n",line2);
            lines++;
            if (lines == CDS_translation_lines[i])
            {
                read_translation (fp2,i+1,cdsdata,line2);
```

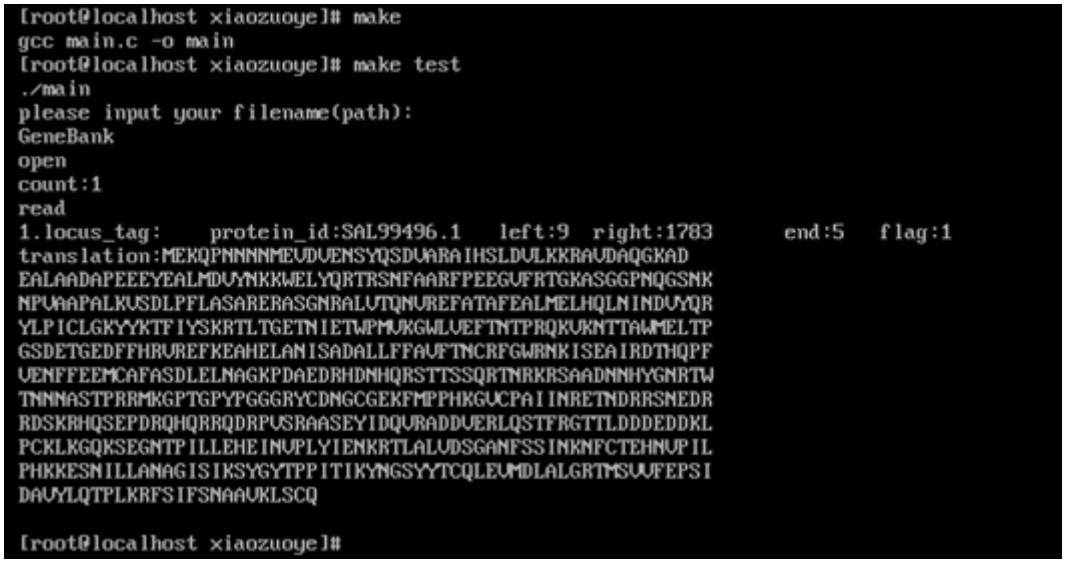
```
                break;
            }

        }
        fclose(fp2);
    }

    writefasta(filename,cdsdata,count);

    return 0;
}

```

make file文件

```
main : main.c libgenbank.so libfasta.so fasta.h main.c
    gcc main.c -o main
test :
    ./main
clean :
    rm main
```

代码测试

运行程序:



查看fasta文件:

第二个测试文件：CDS_test.txt 运行后生成CDS_test.fasta文件
命令行：

```
[root@localhost xiaozuoye]# make test
./main
please input your filename(path):
CDS_test.txt_
```

```
1.locus_tag:    protein_id:SAL99496.1   left:9  right:1783      end:5   flag:1
2.locus_tag:MAB_0174    protein_id:YP_001700928.1       left:1  right:810      end:5   flag:0
3.locus_tag:MAB_0175    protein_id:YP_001700929.1       left:1001       right:1975      end:0   flag
:0
translation:MEKQPNNNNMEVDVENSYQSDVARAIHSLDVLKKRAVDAQGKAD
EALAADAPEEEYEALMDVYNKKWELYQRTRSMFAARFPEEGVFRTGKASGGPNQGSNK
NPVAAPALKVSDLPFLASARERASGNRALVTQNVREFATAFEALMELHQLNINDVYQR
YLPICLGKYYKTFIYSKRTLTGETNIETWPMVKGWLVEFTNTPRQKVKNTTAWMELTP
GSDETGEDFFHRVREFKEAHELANISADALLFFAVFTNCRFGWRNKISEAIRDTHQPF
VENFFEEMCAFASDLELNAGKPDAEDRHDNHQRSTTSSQRTNRKRSAADNNHYGNRTW
TNNNASTPRRMKGPTGPYPGGGRYCDNGCGEKFMPPHKGVCPAIIMRETNDRRSNEDR
RDSKRHQSEPDRQHQRRQDRPVSRAASEYIDQVRADDVERLQSTFRGTTLDDDEDDKL
PCKLKGQKSEGNTPILLEHEINVPLYIEMKRTLALVDSGANFSSIMKNFCTEHNVPIL
PHKKESNILLANAGISIKSYGYTPPITIKYNGSYYTCQLEVMDLALGRTMSVVFEPSI
DAVYLQTPLKRFSIFSNAAVKLSCQ

translation:MGPSNGLTRFTVAVSVLTVTVLFGWGAVQRRWIADDGLIVLRTV
RNLLAGNGPVFNKGERVEANTSTLWTYLTYLGGWAGGGMRLEYVALTLSLVLSLVGVA
LAILGTARLYAPLLAGRAAVMVPAGMLVYIAIPPARDFATSGLENGLVLAYLGGLWLM
MVKWAQAVRTPVTLDRRGGPHQPVDPRIVHAQRDQRDVLSRRFTVGLAFLAGLSVLIR
PELALIGGGFLVMMLVAARGFMSRIWIVVAGGALPVLYQIFRMGYYGLLVPSTAIAKD
ASGSKWGQGFVYLQNMNSPYLIWIPAVLLIALGVAAYQARRGQWAVRQVAAPGYGWLA
RLVQNPTAVVVFMLVSGFVQGVVWIRQGGDFMHARVLLTPVFCMLLPISVVPLAAPDS
AAFTPKKARLLTAATIGLFAGIAGWSVVAVNSPGMAGDGTKVTYSGIVDERRFYAQAT
GVAHPLTAADYLNYPRMRAVLVAIDNTPDGALLLPSGNYDQWDVAPAIPPPPPIPHGY
RGPHAVLFTNLGMLGMNLGLDVRIVDQIGLAMPLAAHTARITDGRIGHDKNLFPDWMI
ADGPWLKRYPYIPRYIDQDAVAEAVEALKCPQTDAMLSAVRKPLSPRLFVSNMLHSYE
FTTYRIDRVPRFELARCGLPMPKLDTPSYTGLPATGP

translation:MSVRVKARRVLSALLAAFVMPVSMAAAMTINPATAHAFSREGLP
VEYLDVYSNSMGRNIRVEFQGGGPKAVYLLDGLRAQDDFNGWDINTAAFEWFYQSGIS
VVMPVGGQSSFYTDWYSPSALNKQPYTYKWETFLTQELPAYLATNKQISATGNGVVGL
SMSGGAALILAAFHPAQFRFAGSLSGFLNPSTIFMTNAIRVAMLDAGSYSVDNMWGPP
WDPAWRRNDPTVQAQALVAAGTRLYIYCAPGGSTPIDDNTDAGVALSASSLESLAVAG
MKAFQQAYTAAGGRNANFVFPASGNHSWPYWGQQLQALKGDLIATLNG
```

CDS_test.fasta文件内容：

```
>locus_tag:     protein_id:SAL99496.1   end:5'  complementary chain
MSVRVKARRVLSALLAAFVMPVSMAAAMTINPATAHAFSREGLP
VEYLDVYSNSMGRNIRVEFQGGGPKAVYLLDGLRAQDDFNGWDINTAAFEWFYQSGIS
VVMPVGGQSSFYTDWYSPSALNKQPYTYKWETFLTQELPAYLATNKQISATGNGVVGL
SMSGGAALILAAFHPAQFRFAGSLSGFLNPSTIFMTNAIRVAMLDAGSYSVDNMWGPP
WDPAWRRNDPTVQAQALVAAGTRLYIYCAPGGSTPIDDNTDAGVALSASSLESLAVAG
MKAFQQAYTAAGGRNANFVFPASGNHSWPYWGQQLQALKGDLIATLNG
>locus_tag:MAB_0174     protein_id:YP_001700928.1       end:5'  not complentmentary chain
MSVRVKARRVLSALLAAFVMPVSMAAAMTINPATAHAFSREGLP
VEYLDVYSNSMGRNIRVEFQGGGPKAVYLLDGLRAQDDFNGWDINTAAFEWFYQSGIS
VVMPVGGQSSFYTDWYSPSALNKQPYTYKWETFLTQELPAYLATNKQISATGNGVVGL
SMSGGAALILAAFHPAQFRFAGSLSGFLNPSTIFMTNAIRVAMLDAGSYSVDNMWGPP
WDPAWRRNDPTVQAQALVAAGTRLYIYCAPGGSTPIDDNTDAGVALSASSLESLAVAG
MKAFQQAYTAAGGRNANFVFPASGNHSWPYWGQQLQALKGDLIATLNG
>locus_tag:MAB_0175     protein_id:YP_001700929.1       not complentmentary chain
MSVRVKARRVLSALLAAFVMPVSMAAAMTINPATAHAFSREGLP
VEYLDVYSNSMGRNIRVEFQGGGPKAVYLLDGLRAQDDFNGWDINTAAFEWFYQSGIS
VVMPVGGQSSFYTDWYSPSALNKQPYTYKWETFLTQELPAYLATNKQISATGNGVVGL
SMSGGAALILAAFHPAQFRFAGSLSGFLNPSTIFMTNAIRVAMLDAGSYSVDNMWGPP
WDPAWRRNDPTVQAQALVAAGTRLYIYCAPGGSTPIDDNTDAGVALSASSLESLAVAG
MKAFQQAYTAAGGRNANFVFPASGNHSWPYWGQQLQALKGDLIATLNG
```

执行make clean，自动清除main文件：

```
[root@localhost xiaozuoye]# make clean
rm main
```

第七组小组分工

libgenbank.so --赖敏智

libfasta.so --韩德坤

main.c --赖敏智、韩德坤

makefile文件及代码测试 --王龙威

git多人协作处理发布 --陆心玥