Биоинформатика

Алекса Вучковић, 3ц

Фебруар 2021.

Садржај

1 Увод

Биоинформатика (грч. bios - живот; енгл. informatics) је интердисциплинарна област која развија методе и алате за разумевање биолошких података. Као интердисциплинарно поље науке, биоинформатика комбинује информационе технологије, статистику, математику и инжињерство како би анализирала и интерпретирала биолошке податке. Биоинформатика се користи у анализама симулација биолошких појава користећи математичке и статистичке технике.

Биоинформатика је заједнички термин за област биолошких студија које користе компјутерско програмирање као део своје методологије, и као референца за специфичне анализе "тока података" које се често користе, посебно у подручју геномике. Типична примена биоинформатике подразумева идентификацију кандидата гена и нуклеотида. Често је циљ њихове идентификације боље разумевање генетске основе разних болести, специфичних прилагођавања организама, жељених особина (нпр. у пољопривредним културама), или разлика између популација. У мање формалном типу, биоинформатика такође покушава да открије организационе принципе унутар нуклеинских киселина и протеинских секвенци.

2 Циљ рада

Циљ рада је био обрађивање основних проблема биоинформатике кроз решавање како једноставнијих тако и комплекснијих и разноврснијих задатака, као и приближавање појма биоинформатике.

Осим тога желео сам да покажем да са знањем програмирања и биологије стеченим у школи можемо да учествујемо у оваквим пројектима који доприносе промоцији науке.

3 Израда задатака у програмском језику C++

Полазна тачка приликом избора тема су били задаци са сајта `https://rosalind.info`. Росалинд је платформа за учење биоинформатике и програмирања кроз решавање задатака. Сваки задатак је структуиран тако да корисник добије информацију коју треба да обради коришћењем програма креираног за ту специфичну ситуацију и на основу резултата обраде се испитује да ли алгоритам, као и његова имплементација представљају решење задатка.

Сви програми које ћете видети у даљем тексту су успешно прошли тестирање на сајту. Осим тога уз решења задатака дато је и објашњење теорије из биологије потребне за разумевање задатка.

Битно је напоменути да сам као аутор тежио да пишем поједностављен, као и модуларан код тако да се неки делови могу користити више пута као што је на пример функција за читање кода у ФАСТА формату која следи.

3.1 Пребројавање азотних база полинуклеотидног ланца ДНК

За почетак ћемо урадити најједноставнији задатак као увод у део рада са кодом. Наш задатак је да пребројимо број азотних база полинуклеотидног ланца ДНК.

```
#include<iostream>
using namespace std;
int br[4];
int main()
{
    string s,ab="ACGT";
    cin>>s;
    for(int i=0;i<s.length();i++) for(int j=0;j<4;j++) if(ab[j]==s[i]) br[j]++;
    for(int j=0;j<4;j++) cout<<br[j]<<' ';
    return 0;
}
```

```
Ulaz:
AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAAGAGTGTCTGATAGCAGC
Izlaz:
20 12 17 21
```

3.2 ФАСТА формат

У биоинформатици и биохемији, ФАСТА формат је формат заснован на тексту који представља нуклеотидне секвенце или секвенце аминокиселина (протеина), у којима су нуклеотиди или аминокиселине представљени помоћу једнословних кодова. Формат такође омогућава именима секвенци и коментарима да претходе секвенцама. Формат потиче из софтверског пакета ФАСТА, али је сада постао готово универзални стандард у области биоинформатике.

Испод можете видети функцију за читање фајла у ФАСТА формату коју ћемо користити у свим преосталим задацима које обрађујемо у овом раду.

```
bool fasta(vector<pair<string,string>> &a,string filename)
{
    ifstream ulaz;
    ulaz.open(filename);
    if(!ulaz)
    {
        cout<<"Doslo je do greske prilikom otvaranja fajla";
        return 1;
    }
    string pom;
    ulaz>>pom;
    while(!ulaz.eof())
    {
        string s0=pom.erase(0,1),s1,s2;
        ulaz>>s1;
        while(!ulaz.eof()&&s1[0]!='>')
        {
            s2.append(s1);
            ulaz>>s1;
        }
        pom=s1;
        a.pb({s0,s2});
    }
    return 0;
}
```

Пример коришћења те функције у програму:

```
#include<bits/stdc++.h>
using namespace std;
#define pb push_back
#include"fasta.cpp"
int main()
{
    vector<pair<string,string>> a;
    if(fasta(a,"fasta.txt")) return 1;
    for(auto p:a) cout<<p.first<<endl<<p.second<<endl;
    return 0;
}
```

```
Ulaz:
>Rosalind_6404
CCTGCGGAAGATCGGCACTAGAATAGCC
AGAACCGTTTCTCTGAGGCTTCCGGCCT
TCCCTCCCACTAATAATTCTGAGG
>Rosalind_5959
CCATCGGTAGCGCATCCTTAGTCCAATT
AAGTCCCTATCCAGGCGCTCCGCCGAAG
GTCTATATCCATTTGTCAGCA
Izlaz:
Rosalind_6404
CCTGCGGAAGATCGGCACTAGAATAGCCAGAACCGTTTCTCTGAGGCTTCCGGCCTTCCCTCCCACTAATAATTCTGAGG
Rosalind_5959
CCATCGGTAGCGCATCCTTAGTCCAATTAAGTCCCTATCCAGGCGCTCCGCCGAAGGTCTATATCCATTTGTCAGCA
```

3.3 ГЦ део

У ФАСТА формату дат нам је полинуклеотидни ланац ДНК за више субјеката и наш задатак је да испишемо име субјекта са највећим садржајем ГЦ(гуанин-цитозин) дела.

У тексту који следи можемо видети како изгледа програм за овај задатак:

```
#include<bits/stdc++.h>
using namespace std;
#define pb push_back
#include"fasta.cpp"
bool sortby(pair<string,double> &a,pair<string,double> &b)
{
    return a.second>b.second;
}
int main()
{
    vector<pair<string,string>> a;
    if(fasta(a,"rosalind_gc.txt")) return 1;
    vector<pair<string,double>> b;
    for(auto p:a)
    {
        int br=0;
        for(int i=0;i<p.second.length();i++)
            if(p.second[i]=='G'||p.second[i]=='C') br++;
        b.pb({p.first,(double)100*br/p.second.length()});
    }
    sort(b.begin(),b.end(),sortby);
    //for(auto p:b) cout<<p.first<<endl<<fixed<<setprecision(6)<<p.second<<endl;
    cout<<b[0].first<<endl<<fixed<<setprecision(6)<<b[0].second<<endl;
    return 0;
}
```

```
Ulaz:
>Rosalind_6404
CCTGCGGAAGATCGGCACTAGAATAGCCAGAACCGTTTCTCTGAGGCTTCCGGCCTTCCC
TCCCACTAATAATTCTGAGG
>Rosalind_5959
CCATCGGTAGCGCATCCTTAGTCCAATTAAGTCCCTATCCAGGCGCTCCGCCGAAGGTCT
ATATCCATTTGTCAGCAGACACGC
>Rosalind_0808
CCACCCTCGTGGTATGGCTAGGCATTCAGGAACCGGAGAACGCTTCAGACCAGCCCGGAC
TGGGAACCTGCGGGCAGTAGGTGGAAT
Izlaz:
Rosalind_0808
60.919540
```

3.4 Рачунање масе протеина

Дат нам је протеин у облику ланца аминокиселина обележених енглеским алфабетом, као и маса сваке аминокиселине, а наш задатак је да за дати ланац израчунамо масу тог протеина.

У структури података mapa ћмо чувати податке потребне за израчунавање масе протеина на основу њиховог пептидног ланца. Сваку аминокиселину смо обележили са великим словом енглеске абецеде(сва слова осим B,J,O,U,X,Z). У даљем тексту можемо видети како изгледа mapa за првих 10 аминокиселина:

```
mapa["A"] = 71.03711;
mapa["C"] = 103.00919;
mapa["D"] = 115.02694;
mapa["E"] = 129.04259;
mapa["F"] = 147.06841;
mapa["G"] = 57.02146;
mapa["H"] = 137.05891;
mapa["I"] = 113.08406;
mapa["K"] = 128.09496;
mapa["L"] = 113.08406;
```

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    map<string,double> mapa;
    #include"protein_mass.h"
    string s;
    cin>>s;
    double sum=0;
    for(int i=0;i<s.length();i++)
    {
        string s1(1,s[i]);
        sum+=mapa[s1];
    }
    cout<<fixed<<setprecision(6)<<sum;
    return 0;
}
```

```
Ulaz:
SKADYEK
Izlaz:
821.392
```

3.5 Транзиције и трансверзије

У овом задатку дат нам је полинуклеотидни ланац пре и после мутација. Наш задатак је да израчунамо однос броја транзиција и броја трансверзија.

Транзиције су тип тачкасте мутације када се нуклеотидна база мења из једне пуринске базе у другу (А ↔ Г) или из једне пиримидинске у другу (Ц ↔ Т), а трансверзије су када се нуклеотидна база мења из пиримидинске у пуринску базу и обрнуто. У овом задатку тачкасте мутације су замена једне нуклетотидне базе. Овај однос нам даје брзу и корисну статистику за анализу генома.

```
#include<bits/stdc++.h>
using namespace std;
#define pb push_back
#include"fasta.cpp"
int main()
{
    vector<pair<string,string>> a;
    if(fasta(a,"tranzicije_i_tranverzije.txt")) return 1;
    string s1=a[0].second,s2=a[1].second;
    int br1=0,br2=0;
    for(int i=0;i<s1.length();i++) if(s1[i]!=s2[i])
    {
        int x=(s1[i]=='A')+(s1[i]=='G'),y=(s2[i]=='C')+(s2[i]=='T');
        if(x+y==2||x+y==0) br2++;
        else br1++;
    }
    cout<<(float)br1/br2;
    return 0;
}
```

```
Ulaz:
>Pre_mutacije
GCAACGCACAACGAAAACCCTTAGGGACTGGATTATTTCGTGATCGTTGTAGTTATTGGA
AGTACGGGCATCAACCCAGTT
>Posle_mutacije
TTATCTGACAAAGAAAGCCGTCAACGGCTGGATAATTTCGCGATCGTGCTGGTTACTGGC
GGTACGAGTGTTCCTTTGGGT
Izlaz:
1.21428571429
```

3.6 Сплајсовање РНК/Ексони и интрони

Пре него што пређемо на захтев задатка морамо видети шта је сплајсовање:

У молекуларној биологији и генетици, сплајсовање је модификација РНК након транскрипције, у којој се интрони уклањају, а ексони се спајају. Оно је неопходно да би типична еукариоцка информациона РНК могла да се користи за произвођење коректног протеина путем транслације. У овом задатку дат нам је полинуклеотидни ДНК ланац, као и низ ланаца који представљају интроне. Циљ нам је да испишемо како би изгледао пептидни низ за дати ДНК ланац.

У структури података mapa ћемо чувати податке потребне за транслацију(синтезу протеина). Сваки кодон се транслира у неку од аминокиселина коју смо обележили са великим словима енглеске абецеде(сва слова осим B,J,O,U,X,Z). У даљем тексту можемо видети како изгледа mapa за првих 5 кодона:

```
mapa["UUU"]='F';
mapa["UUC"]='F';
mapa["UUA"]='L';
mapa["UUG"]='L';
mapa["UCU"]='S';
```

```
#include<bits/stdc++.h>
using namespace std;
#define pb push_back
#include"fasta.cpp"
int main()
{
    vector<pair<string,string>> a;
    if(fasta(a,"rna_splicing.txt")) return 1;
    for(auto &str:a) for(int i=0;i<str.second.length();i++)
        if(str.second[i]=='T') str.second[i]='U';
    string s=a[0].second;
    a.erase(a.begin());
    for(auto str:a) s.erase(s.find(str.second),str.second.length());
    map<string,char> mapa;
    #include"rna_splicing.h"
    for(int i=0;i<s.length();i+=3)
    {
        string s1(s.begin()+i,s.begin()+i+3);
        cout<<mapa[s1];
    }
    return 0;
}
```

У примеру који следи дати су нам почетни ДНК ланац као и два интрона која треба да уклонимо из тог ланца. Прво вршимо транскрипцију, тј преводимо дати полинуклеодидни ланац ДНК у полинуклеотидни ланац РНК(примарни транскрипт). Можемо уочити на којим местима се интрони појављују у РНК ланцу, а затим их и уклонити. Преостале ексоне спајамо и на крају их уз помоћ mape, коју смо помињали раније, транслирамо у пептидни ланац.

```
Ulaz:
>DNK_Lanac
ATGGTCTACATAGCTGACAAACAGCACGTAGCAATCGGTCGAATCTCGAGAGGCATAT
GGTCACATGATCGGTCGAGCGTGTTTCAAAGTTTGCGCCTAG
>Intron_1
ATCGGTCGAA
>Intron_2
ATCGGTCGAGCGTGT
Izlaz:
MVYIADKQHVASREAYGHMFKVCA
```

4 Закључак

Информатика је постала део свега па тако и биологије. Олакшала је истраживања и допринела формирању биологије као науке. Многи биолошки и биохемијски проблеми могу се ефикасно решити програмском имплементацијом одговарајућих алгоритама. С обзиром да живимо у времену када технологија брзо напредује, корисно је фокусирати се на тражењу што више информатичких решења проблема из ових области. Због тога мислим да је битно радити на развоју биоинформатике, као једне од најнапреднијих и најзначајнијих биолошких дисциплина.

5 Литература

[1] Задаци коришћени у раду `http://rosalind.info/problems/list-view/`

[2] Јелена Поповић, *Молекуларна биологија за 4. разред Математичке Гимназије*, `https://www.mg.edu.rs/uploads/files/images/stories/dokumenta/profesori/jelena-popovic/molekularna-biologija.docx`

[3] Таблица моноизотопних маса аминокиселина `http://rosalind.info/glossary/monoisotopic-mass-table/`

[4] Таблица транслације кодона у аминокиселине `http://rosalind.info/glossary/rna-codon-table/`

[5] Речник појмова и ређе коришћених израза `http://rosalind.info/glossary`