Tietorakenteiden harjoitustyö:
Sekvenssianalyysiä bioinformatiikkaan

Projektin sisältö

Projektissa toteutan joukon bioinformatiikassa käytettäviä sekvenssianalyysalgoritmeja. Nämä algoritmit pohjautuvat dynaamiseen ohjelmointiin ja niiden toiminta perustuu sekvenssien kohdistusten (alignment) pisteyttämiseen. Niissä käytetään kahta MxN-matriisia, jossa M ja N ovat syötesekvenssien pituudet, joista toiseen talletetaan kohdistusten pisteytykset ja toiseen reitti jolla kyseiseen pisteytykseen päästään. Yhtenäisten elementtien lisäksi eri algoritmit käyttävät erilaisia tapoja sekä pisteytykseen että tuloksen etsimiseen valmiista pistematriisista.

Algoritmien kuvaus

Kohdistuksella tarkoitetaan tässä yhteydessä tapaa asettaa kaksi sekvenssiä vierekkäin siten, että tietyt merkit ovat vastaavissa asemissa. Esimerkiksi olkoon meillä kaksi DNA-sekvenssiä: AGATGAC ja GGACTA. Näistä voidaan tehdä esimerkiksi seuraavanlainen kohdistus, josta havaitaan, että molemmilla sekvensseillä on yhteinen *alisekvenssi* TCTA.:

A **G** – **A** – **T** G **A** C

– **G** G **A** C **T** – **A** –

Bioinformatiikassa törmätään usein ongelmaan, jossa kahdesta tai useammasta sekvenssistä pyritään löytämään yhtäläisyyksiä. Luonnossa tapahtuvien mutaatioiden vuoksi samanlaisuudella ei kuitenkaan tarkoiteta identtisyyttä, mikä vaikeuttaa asiaa. Tarkoituksena on siis löytää mahdollisimman samanlaisia sekvenssejä, jollain samanlaisen määritelmällä. Näissä algoritmeissa lähdetään olettamuksesta, että sekvenssille voi luonnossa tapahtua mikä tahansa kolmesta vaihtoehdosta: siihen voi tulla uusi merkki, merkki voi muuttua toiseksi tai kadota kokonaan. Tapauksen mukaan näillä tapahtumilla saattaa olla eri todennäköisyydet, joten yhtäläisyyksiä etsittäessä tämä on otettava huomioon antamalla eriarvoisia pisteytyksiä eri tapahtumille. Joissain ongelmissa saatetaan joutua olettamaan, että mutaatioita ei tapahdu, tai että vain tietynlaisia tapahtuu. Usein halutaan myös, että sarjassa samanlaisia mutaatioita (esimerkiksi 20 peräkkäisen merkin katoamisessa) näille ei anneta samanlaista painoarvoa kuin yksittäiselle tapahtumalle.

Longest Common Subsequence (LCS) -algoritmi etsii esimerkissä näkyvän kaltaisia alisekvenssejä. Global Sequence Alignment (GSA) -algoritmi etsii parhaan tavan kohdistaa kaksi sekvenssiä siten, että koko algoritmin pituus otetaan huomioon. Local Sequence Alignment (LSA) -algoritmi puolestaan etsii sellaisen kohdistuksen, jossa on mahdollisimman suuri osa kohdistuksen sekvensseistä on identtistä. Näille algoritmeille voidaan antaa lisämääreenä erilainen käsittelytapa yllämainitun kaltaisille tapauksille joissa toistuvat mutaatiot halutaan käsitellä eri tavalla..

Tämäntyyppisiä ongelmia ratkaistaan analysoitaessa geenien DNA-sekvenssejä tai proteiinien aminohappoketjuja. Valitsin kyseisen aiheen, sillä bioinformatiikan tutkimus kiinnostaa minua ja halusin perehtyä syvemmin algoritmeihin, joita oli pintapuolisesti käsitelty kurssilla Algortihms for Bioinformatics.

Ohjelman käyttö

Ohjelmaa käyttäessään käyttäjä valitsee tekstipohjaisesta käyttöliittymästä minkä sekvenssianalyysin hän haluaa suorittaa, antaa syötteen ja kertoo pisteytyksskeeman. Syötesekvenssit annetaan tekstitiedostona, josta käyttäjä antaa tiedoston nimen. Ohjelma olettaa, että sekvenssit ovat ensimmäisenä tekstitiedostossa ja erotettuna toisistaan rivinvaihdolla. Mikäli pisteytysskeema on monimutkainen, senkin voi antaa tekstitiedostona. Tämän jälkeen ohjelma suorittaa analyysin ja palauttaa tuloksen syötteenä ruudulle.

Ohjelman tehokkuus

Aikavaativuustavoite on kaikissa algoritmeissa $O(n^2)$. Koska algoritmit käyttävät syötteiden pituuden mukaista matriisia, joka täytetään algoritmia ajettaessa, ei pienempään päästä. Silti kaikki algoritmit pitäisi pystyä pitämään neliöllisinä. Tilavaativuus on samoista syistä $O(n^2)$.

Lähteet

Pääasialliset lähteenä tällä hetkellä "An Introduction to Bioinformatics Algorithms", Jones, Neil C. ja Pevzner, Pavel A. MIT Press, 2004.