Kemistin data-analysaattori
Toteutusdokumentti (2. iteraatio)

Taisto Gerdt
Timo Kääriäinen
Mika Miettinen
Pasi Rastas
Jari Tuominiemi

Ohjelmistotuotantoprojekti
Helsingin yliopisto
Tietojenkäsittelytieteen laitos
27. marraskuuta 2002

Versiohistoria:

Päiväys	Muutokset	Muutosten tekijä
26.11.2002	Alustava versio	Mika Miettinen
27.11.2002	Ryhmäkokouksessa 26.11 sovitut muutokset	Mika Miettinen

Sisältö

1 Yleistä **1**

1.1 Projektin aihepiiri 1

1.2 Ohjelmiston yleiskuvaus 1

1.3 Ohjelmiston toiminnan yleiskuvaus 2

1.4 Toteutuksen yleiskuvaus 2

2 Laskentamenetelmä **3**

2.1 Äitispektrin kerroinmatriisi 3

2.2 Tytärspektrin kerroinmatriisi 4

2.2.1 “Full scan”-menetelmä 4

2.2.2 “Daughter ion scan”-menetelmä 4

2.3 Lopullinen yhtälö 5

3 Tietosisältö **6**

3.1 Globaalit muuttujat 6

3.2 Fragmenttitietokanta 7

3.2.1 Fragmenttitietokannan tietosisältö 7

3.2.2 Fragmenttitietokannan käsittely 7

3.2.3 Fragmenttitietokannan ylläpitoon liittyvä käyttöliittymä 8

4 Arkkitehtuuri **8**

5 Ulkoiset liittymät **9**

5.1 Syötetiedosto 9

5.2 Syötetiedoston käsittely 12

5.3 Tulostiedosto 12

6 Käyttöliittymä **13**

7 Moduulien kuvaus **13**

7.1 Käyttöliittymä 13

7.2 Syntaksin tarkastus 15

7.3 Syöte- ja tulostiedostojen käsittely 15

7.4 Piikkien tunnistus 16

7.5 Tietokannan luku ja päivitys 16

7.6 Kerroinmatriisin luominen (äitispektri) 16

7.7 Kerroinmatriisin luominen (“Full scan” -fragmentointi) . . . 17

7.8 Kerroinmatriisin luominen (“Daughter ion scanning”) 18

7.9 Laskenta 19

7.10 Apufunktiot 19

A Syntaksit EBNF-muodossa **20**

A.1 Syötetiedostot 20

A.2 Fragmenttitietokannan tiedostot 21

B Esimerkki syötetiedostosta **21**

C Esimerkki fragmenttitietokannan sisällöstä **22**

Lähteet **23**

1 Yleistä

1.1 Projektin aihepiiri

Mikrobisolujen kasvua voidaan analysoida muun muassa mittaamalla solun tuottamia kemiallisia yhdisteitä. Erityisen tehokas menetelmä on niinkutsuttu isotooppijäljitys, jossa solulle syötetään hiili-13 -isotooppia sisältävää ravinnetta ja mitataan solun tuottamien yhdisteiden isotopomeerijakaumia, eli isotooppien jakaumia yhdisteen hiiliketjussa. Näiden isotopomeerijakaumien määrittämiseksi käytetään niinkutsuttua tandem-massaspektrometria, jossa voidaan mitata sekä koko molekyylin tuottama spektri että molekyylin palasten spektrit. Yhdistämällä eri spektrien tiedot isotopomeerijakaumasta saadaan tarkempi arvio kuin käyttämällä yhtä spektriä.

1.2 Ohjelmiston yleiskuvaus

Ohjelmiston pääasiallinen käyttötarkoitus on molekyylin positionaalisten isotopomeerien jakauman laskeminen massaspektrometrin tuottaman datan perusteella. Laskenta tapahtuu laitoksella kehitetyn menetelmän [Ran02, RoR02] mukaan.

Tämän lisäksi ryhmä toteutti fragmenttitietokannan aminohappomolekyylien sekä näistä syntyvien fragmenttien ominaisuuksien tallettamiseksi. Tällä tietokannalla on oleellinen osa pääohjelman suorittamassa isotopomeerijakauman laskemisessa. Fragmenttitietokanta on toteutettu kokoelmana tekstitiedostoja, ja sen ylläpito tapahtuu Matlab-ohjelmiston oman tekstieditorin avulla.

Kehitystyön pohjana käytettiin aiemmin laitoksella kehitettyä PIDC-nimistä Matlab-sovellusta [Ran01].

1.3 Ohjelmiston toiminnan yleiskuvaus

Pääpiirteissään ohjelmiston toiminta koostuu seuraavista askeleista:

1. Ohjelma lukee käyttäjän valitseman syötetiedoston.
2. Ohjelma nimeää syötetiedoston sisältämät piikistöt atomikaavoilla käyttäen apuna fragmenttitietokantaa.
3. Piikistöt nimettyään ohjelma laskee isotopomeerien jakauman.
4. Ohjelma tallentaa näin lasketun isotopomeerijakauman tekstitiedostoon.

1.4 Toteutuksen yleiskuvaus

Ohjelmisto on toteutettu kokonaan Matlabin avulla. Pääasiallinen toimivuus muodostuu joukosta Matlab-ohjelmointikielellä toteutettuja funktioita. Näiden lisäksi ohjelmiston käyttöliittymä on tehty Matlab-ohjelmiston käyttöliittymäkomponenttien avulla.

Koska yhdessäkään laitoksen mikroluokassa ei ole Matlab-ohjelmiston Windows-versiota, teki ryhmä kehitystyön Linux-ympäristössä. Matlabin Windows- ja Linux-versiot ovat kuitenkin pieniä yksityiskohtia lukuunottamatta toiminnallisesti täysin identtisiä, joten tämä ei muodostanut suurta ongelmaa. Lisäksi yhdellä ryhmän jäsenellä (Timo Kääriäinen) on mahdollisuus käyttää Matlab 6 -ohjelmiston Windows-versiota, ja tällä tavoin ryhmä pystyi vakuuttumaan että ohjelmisto toimii myös Windows-ympäristössä.

Koska kehitystyö suoritettiin Linuxissa, toimii lopullinen ohjelmisto sekä Linux- että Windows-ympäristöissä. Ohjelma toimii mahdollisesti myös muissa käyttöjärjestelmissä, joihin Matlab-ohjelmisto on saatavissa.

2 Laskentamenetelmä

Artikkeleissa [Ran02] ja [RoR02] kuvatun laskentamenetelmän toteutus muodostaa suurimman osan tehtävästä ohjelmistosta. Karkeasti kuvattuna laskenta koostuu lineaarisen yhtälöryhmän johtamisesta syötetietojen ja fragmenttitietokannan avulla. Ratkaisemalla tämä yhtälöryhmä saadaan selvitetyksi molekyylin positionaalinen isotopomeerijakauma. Suurin osa laskentaa koostuukin kyseisen yhtälöryhmän kerroinmatriisin laskemisesta. Tässä luvussa selitetään pääkohdat kerroinmatriisien laskennasta.

2.1 Äitispektrin kerroinmatriisi

Äitispektri tarkoittaa massaspektrin sitä osaa, joka kuvaa tutkittavaa aminohappomolekyyliä ja sen massaisotopomeerejä. Kerroinmatriisin laskenta alkaa m -massaisen molekyylin $m+l$ -piikin suhteellisesta intensiteetistä (N on molekyylin M ylimääräisten neutronien maksimimäärä):

$$\iota_M(m+l) = \frac{h(m+l)}{\sum_{t=m}^{m+N} h(t)} = \mathbb{P}(^{+l}\{M\}). \quad (1)$$

Yhtälön oikeanpuolimmainen termi voidaan edelleen esittää isotopomeerien summana

$$\mathbb{P}(^{+l}\{M\}) = \sum_{i=0}^{l} \mathbb{P}(^{+l-i}\{\overline{C}\}) \cdot \mathbb{P}(^{+i}\{C\}).$$

Lausumalla termi $\mathbb{P}(^{+i}\{C\})$ positionaalisten isotopomeerien summana ja sijoittamalla lausekkeeseen 1, saadaan

$$\iota_M(m+l) = \sum_{\mathbf{w}\in\{0,1\}^n} {}^{\mathbf{w}}\alpha_M^l \mathbb{P}(^{\mathbf{w}}\{C\}).$$

Tämä yhtälö esittää yhtä ratkaistavavista lineaariyhtälöistä. Kerroinmatriisiksi tulee siten

$$\begin{bmatrix} {}^{0\ldots0}\alpha^0 & \ldots & {}^{1\ldots1}\alpha^0 \\ \vdots & \ddots & \vdots \\ {}^{0\ldots0}\alpha^N & \ldots & {}^{1\ldots1}\alpha^N \end{bmatrix}.$$

2.2 Tytärspektrin kerroinmatriisi

Tytärspektri tarkoittaa massaspektrin niitä piikkejä jotka vastaavat tutkittavan molekyylin fragmenttien eri isotopomeerejä. Tytärspektrin kerroinmatriisin laskentamenetelmä riippuu siitä, onko massaspektri tuotettu "Full scan"- vai "Daughter ion scan"-menetelmällä. Seuraavissa alaluvuissa käsitellään kerroinmatriisin laskentaa näille eri menetelmille.

2.2.1 "Full scan"-menetelmä

"Full scan"-moodissa kukin piikki $m/z = t$ sisältää kaikki isotopomeerit ${}^{+k}\{F\}$, joiden massa on t. Merkitään näiden isotopomeerien joukkoa $\mathcal{F}(t)$. Piikin suhteelliselle korkeudelle saadaan nyt lauseke

$$\iota_D(t) = \sum_{{}^{+k}\{F\} \in \mathcal{F}(t)} \mathbb{P}({}^{+k}\{F\}) \cdot \frac{\mathbb{D}_E(F)}{\sum_{F_i \subset M} \mathbb{D}_E(F_i)}. \tag{2}$$

Positionaalisten isotopomeerien avulla lausuttuna kerroin $\mathbb{P}({}^{+k}\{F\})$ saadaan edelleen muotoon

$$\mathbb{P}({}^{+k}\{F\}) = \sum_{\mathbf{w} \in \{0,1\}^n} {}^{\mathbf{w}}\alpha_M^k \mathbb{P}({}^{\mathbf{w}}\{C\}).$$

Sijoittamalla tämä yhtälöön 2 saadaan piikin suhteelliseksi korkeudeksi

$$\iota_D(t) = \sum_{\mathbf{w} \in \{0,1\}^n} \mathbb{P}({}^{\mathbf{w}}\{C\}) \cdot {}^{\mathbf{w}}\beta^m.$$

Lopulliseksi kerroinmatriisiksi tulee siis

$$\begin{bmatrix} {}^{0\ldots0}\beta^0 & \ldots & {}^{1\ldots1}\beta^0 \\ \vdots & \ddots & \vdots \\ {}^{0\ldots0}\beta^{m+N} & \ldots & {}^{1\ldots1}\beta^{m+N} \end{bmatrix}$$

2.2.2 "Daughter ion scan"-menetelmä

"Daughter ion scan"-menetelmällä tuotetussa massaspektrissä on ensin valittu isotopomeeri ${}^{+l}\{M\}$, joka on sitten fragmentoitu. Tämä toistetaan sitten kaikille halutuille massaisotopomeereille. Isotopomeeristä ${}^{+l}\{M\}$ syntyneen piikin $m/z = t$ suhteelliseksi korkeudeksi saadaan

$$\iota_D(t,l) = \sum_{{}^{+k}\{F\} \in \mathcal{F}(t)} \mathbb{P}({}^{+k}\{F\} | {}^{+l}\{M\})) \cdot \frac{\mathbb{D}_E(F)}{\sum_{F_i \subset M} \mathbb{D}_E(F_i)}. \tag{3}$$

Ehdollinen todennäköisyys voidaan edelleen kirjoittaa muotoon

$$\mathbb{P}({}^{+k}\{F\} | {}^{+l}\{M\}) = \\ \sum_{i=0}^{k} \sum_{j=0}^{l-k} \mathbb{P}({}^{+i}\{C \cap F\} \cap {}^{+j}\{C \cap \overline{F}\}) \cdot \frac{\mathbb{P}({}^{+k-i}\{\overline{C} \cap F\} \cdot \mathbb{P}({}^{l-k-j}\{\overline{C} \cap \overline{F}\})}{\mathbb{P}({}^{+l}\{M\})}$$

Lausumalla tämä positionaalisten isotopomeerien avulla ja sijoittamalla yhtälöön 3 saadaan piikin suhteelliseksi korkeudeksi

$$\iota_D(t) = \sum_{\mathbf{w} \in \{0,1\}^n} \mathbb{P}({}^{\mathbf{w}}\{C\}) \cdot {}^{\mathbf{w}}\beta_F^{m,l}.$$

Lopulliseksi kerroinmatriisiksi tulee siten

$$\begin{bmatrix} {}^{0\ldots0}\beta^{0,l} & \ldots & {}^{1\ldots1}\beta^{0,l} \\ \vdots & \ddots & \vdots \\ {}^{0\ldots0}\beta^{m+N,l} & \ldots & {}^{1\ldots1}\beta^{m+N,l} \end{bmatrix}$$

2.3 Lopullinen yhtälö

Kokoamalla kaikki edellä saadut matriisit yhteen, saadaan yhtälö

$$S \cdot \mathbb{I}_M = \begin{bmatrix} A \\ B \\ B_0 \\ \vdots \\ B_N \end{bmatrix} \cdot \begin{bmatrix} \mathbb{P}({}^{0\ldots0}\{C\}) \\ \vdots \\ \mathbb{P}({}^{1\ldots1}\{C\}) \end{bmatrix} = \begin{bmatrix} \vec{\iota} \\ \vec{\iota}_{FS} \\ \vec{\iota}_{DIS}(0) \\ \vdots \\ \vec{\iota}_{DIS}(N) \end{bmatrix}$$

Molekyylin positionaalinen isotopomeerijakauma saadaan ratkaisemalla tämä yhtälö.

3 Tietosisältö

3.1 Globaalit muuttujat

Ohjelmiston keskeisin tietosisältö talletetaan globaaleihin muuttujiin. Nämä muuttujat esitetään seuraavassa.

EPSILON
: Toleranssivakio, jonka alle jäävät luvut käsitellään nollina.

AminoAcids
: Muuttuja sisältää kaikkien 20 aminohapon viralliset kolmikirjaimiset IUPAC-lyhenteet ja niiden massaspektrometrimittauksissa esiintyvien [M+H]-ionimuotojen massat. Lisäksi muuttuja sisältää kustakin aminohaposta massaspektrometrissä syntyvien fragmenttien tiedot.

Elements
: Sisältää aminohappomolekyyleissä esiintyvien atomien nimet, atomipainot ja eri isotooppien luonnolliset esiintymistiheydet. Esiintymistiheydet ovat taulukossa siten, että 1. alkio sisältää +0-isotoopin esiintymistiheyden, 2. alkio +1-isotoopin jne.

InputArray
: Taulukko, joka sisältää ohjelmalle annetun syötetiedoston sisällön.

InputComments
: Tämä taulukko sisältää syötetiedoston alussa esiintyneet kommentit, jotka liitetään sellaisinaan mukaan tulostiedostoon.

InputDatalines
: Taulukko sisältää syötetiedoston kaikki datarivit, jotka nekin liitetään mukaan tulostiedoston alkuun.

PeakList
: Sisältää syötiedostosta fragmenttitietokannan avulla tunnistettujen piikkien tiedot.

3.2 Fragmenttitietokanta

3.2.1 Fragmenttitietokannan tietosisältö

Fragmenttitietokanta on toteutettu tekstitiedostopohjaisena siten, että kutakin massaspektrometrissä käytettyä energiaa ja laitteistoa tai teknistä järjestelyä kohden on oma tiedostonsa. Tiedostossa esitellään sarakkeittain seuraavat tiedot:

- ioniin liittyvä äitimolekyylin nimi
- ionin asema (M-äitimolekyyli-ioni, D=tytärioni äitimolekyylille)
- ionin kemiallinen kaava
- ionissa mukana olevat hiiliatomit
- ionin kaavapaino
- ionin suhteellinen osuus (vain tytärioneille)

Äitimolekyylin kemiallinen kaava esitetään ionimuodossa siten, kuin se massaspektrissä esiintyy. Näin esimerkiksi kaavapainoltaan 89 oleva alaniini (C3H7NO2) esitetään tiedostossa muodossa C3H8NO2 (kaavapaino 90). Tiedostossa esiintyvä kaavapaino voitaisiin laskea myös suoraan kaavasta, mutta se on otettu mukaan tiedostoon helpottamaan tiedonhakua fragmenttitietokannasta. Viimeisessä sarakkeessa esitetään tytärionin piikin suhteellinen intensiteetti kyseisellä energialla. Jos rivi kuvaa äitimolekyyliä, on sarake tyhjä. Liitteessä A.2 on fragmenttitietokannan tiedoston EBNF-muotoinen syntaksi ja liitteessä C esimerkki fragmenttitietokannan sisällöstä 30 eV energialle.

3.2.2 Fragmenttitietokannan käsittely

Käyttöliittymässä tehtyjen valintojen perusteella ohjelmisto valitsee ja lukee muistiin fragmenttitietokannan sen tiedoston, joka vastaa määriteltyä energiamäärää ja laitetyyppiä. Funktio `parseFDB` tarkastaa tämän tiedoston syntaksin ja sijoittaa fragmenttien tiedot oikeille paikoilleen `AminoAcids`-taulukkoon (eli kunkin fragmentin tiedot menevät sen aminohapon kohdalle josta kyseinen fragmentti on syntynyt).

Laskennassa tarvitaan tästä tietokannasta vain tietyn äitimolekyylin ja siitä syntyneiden tytärionien tiedot. Tätä tarkoitusta varten on funktio `findFragments`, joka hakee muuttujasta `AminoAcids` halutun äitimolekyylin ja sen fragmenttien tiedot.

3.2.3 Fragmenttitietokannan ylläpitoon liittyvä käyttöliittymä

Fragmenttitietokannan ylläpitoon ei ole kehitetty varsinaista käyttöliittymää. Halutessaan päivittää tietokantaa käyttäjä painaa ohjelman pääikkunassa olevaa painonappia, joka avaa tietokannan oikean tiedoston Matlabin editoriin. Tätä editoria käyttämällä käyttäjä voi poistaa ja lisätä sekä muokata tietokannan tietoja.

4 Arkkitehtuuri

Kuten edellä mainittiin, koostuu ohjelma joukosta Matlab-ohjelmointikielellä toteutettuja funktioita. Käytetty ohjelmointikieli ei juurikaan anna mahdollisuuksia itsenäisten moduulien tekemiseen. Ohjelmiston voi kuitenkin katsoa loogisesti koostuvan seuraavista moduuleista:

- Käyttöliittymä
- Syntaksin tarkastus
- Syöte- ja tulostiedostojen käsittely
- Piikkien tunnistus
- Tietokannan luku ja päivitys

- Kerroinmatriisin luominen (äitispektri)
- Kerroinmatriisin luominen (full scan)
- Kerroinmatriisin luominen (daughter ion scan)
- Laskenta

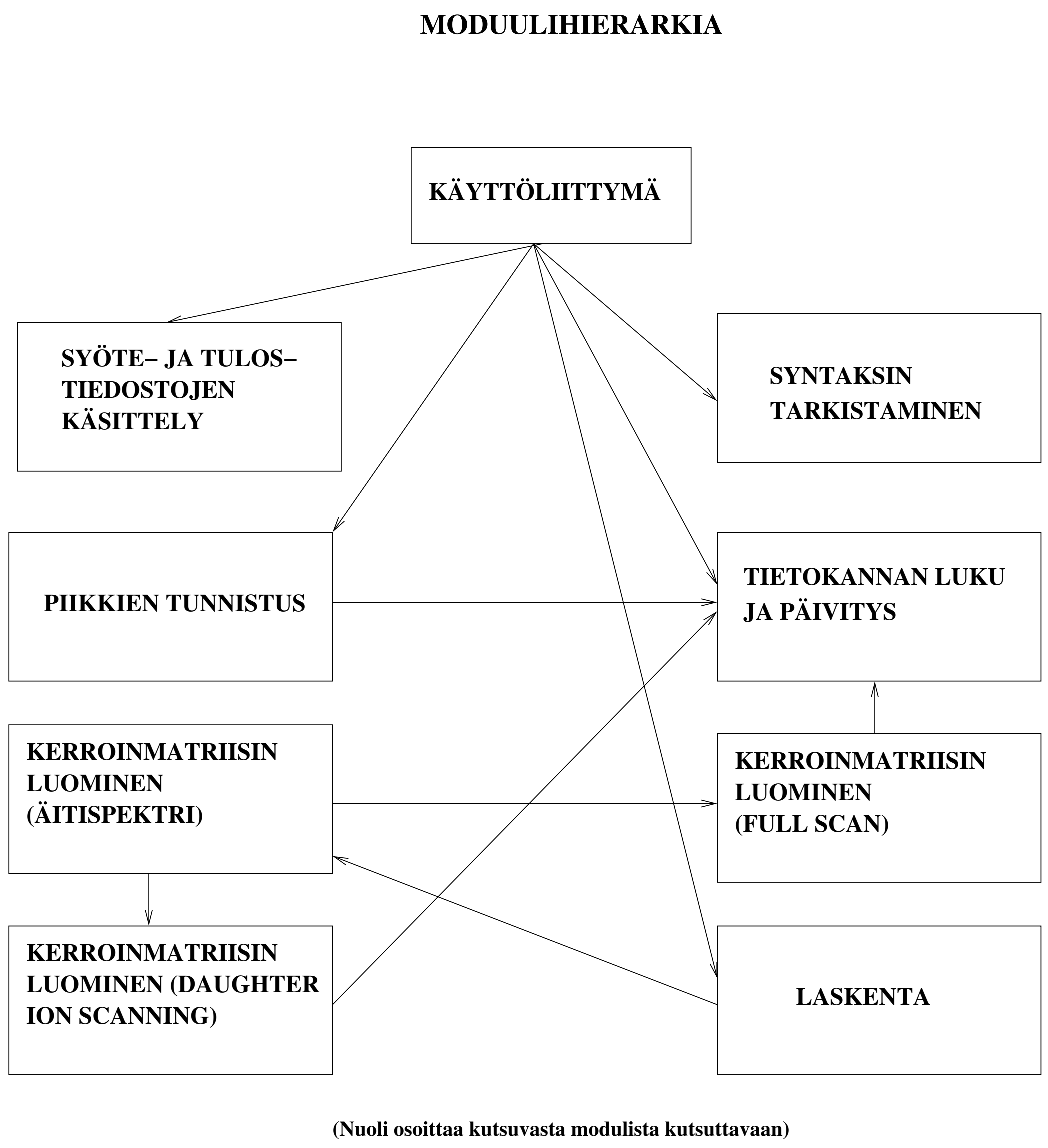
Näiden lisäksi ohjelmaan kuuluu neljä apufunktiota, joiden ei voi katsoa kuuluvan yhteenkään edellämainituista moduuleista, mutta eivät ole tehtäviltään kuitenkaan tarpeeksi yhtenäisiä muodostaakseen oman moduulinsa. Moduulit sekä niihin kuuluvat funktiot (ja edellämainitut apufunktiot) kuvataan luvussa 7. Ohjelmiston moduulihierarkia näkyy kuvassa 1 ja tietovuokaavio kuvassa 2.

5 Ulkoiset liittymät

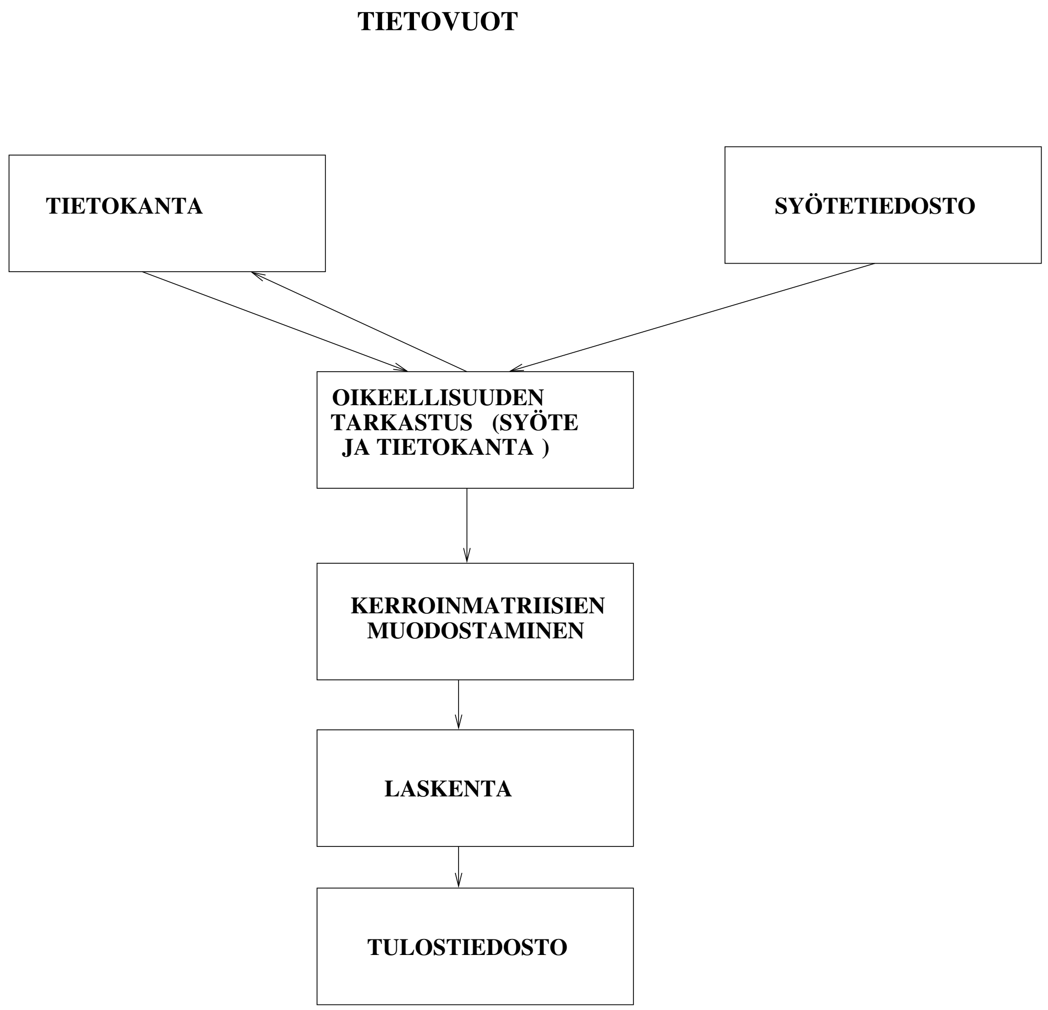
5.1 Syötetiedosto

Ohjelman lukema syötetiedosto sisältää massaspektrometrin mittauksista saadun datan. Käsittelyn helpottamiseksi tiedoston oletetaan sisältävän vain yhden äitimolekyylin tiedot. Lisäksi oletetaan, että tiedoston ensimmäinen datarivi sisältää perusmassaisen äitimolekyylin tiedot.

Syötetiedosto sisältää rivejä, jotka ovat joko kommenttirivejä tai kuvaavat yhtä massaspektrin piikkiä. Kommenttirivit alkavat prosenttimerkillä ja sisältävät tämän jälkeen mitä tahansa merkkejä jatkuen aina seuraavaan rivinvaihtomerkkiin saakka. Kukin datarivi sisältää kolme tabulaattorein (tai välilyönnein) erotettua arvoa: ensimmäisenä molekyylin massa (oikeammin massa/varaus -suhde) ja toisena sen intensiteetti. Kolmannelle sarakkeelle on kolme mahdollisuutta: se voi olla tyhjä, sisältää pienen kokonaisluvun tai sisältää M-kirjaimen. Jos sarake sisältää M-kirjaimen, kuvaa kyseinen piikki äitimolekyyliä. Jos sarake taas on tyhjä, on massaspektri tuotettu “Full scan”-menetelmällä ja piikki kuvaa tytärionia. Jos



Kuva 1: Ohjelmiston moduulihierarkia



Kuva 2: Ohjelmiston tietovuokaavio

massaspektri on tuotettu "Daughter ion scan"-menetelmällä, on sarakkeessa kokonaisluku. Tällöin kyseinen piikki kuvaa tytärionia ja luku kertoo mistä äitimolekyylin isotopomeeristä kyseinen fragmentti on peräisin.

Syötetiedoston sisältämät lukuarvot ajatellaan loogisesti kokonaisluvuiksi, mutta myös liukuluvut hyväksytään; ne pyöristetään kuitenkin lähimpään kokonaislukuarvoon. Tälläinen ratkaisu on tehty koska lukuarvot voisivat olla esim. mittaustuloksia ja siten sisältää hieman epätarkkuutta.

Heti tiedoston alussa olevat kommenttirivit tulkitaan liittyvän itse mittaukseen ja ne liitetään sellaisenaan mukaan tulostiedostoon. Syötetiedoston syntaksi EBNF-notaatiolla esitettynä löytyy liitteestä A.1 ja esimerkki 30 eV:n energialla tuotetusta syötetiedostosta liitteestä B.

5.2 Syötetiedoston käsittely

Käyttöliittymässä valittu syötetiedosto luetaan muistiin ja funktio `parseInput` tarkastaa sen syntaksin. Tämä funktio myös huolehtii syötteen muuttamisesta sen lopulliseen, sisäiseen esitysmuotoon. Tiedoston alussa olevat kommenttirivit sijoitetaan `InputComments`-taulukkoon ja tiedoston datarivit sijoitetaan taulukkoon `InputDatalines`. Laskentaa varten datarivit muunnetaan myös struct-tietueiksi ja nämä tietueet talletetaan `InputArray`-taulukkoon.

5.3 Tulostiedosto

Ohjelman tuottama tulostiedosto sisältää laskennan tuloksena syntyneen isotopomeerijakauman puhtaana tekstitiedostona. Tulostiedosto sisältää aluksi tiedon tutkittavan molekyylin atomikaavasta. Tämän jälkeen tiedosto sisältää rivejä, joista kukin ilmoittaa jonkin positionaalisen isotopomeerin suhteellisen osuuden. Rivi sisältää ensin tiedon isotopomeeristä muodossa $[b_1 b_2 \ldots b_n]$ ja tämän jälkeen tabulaattorilla erotettuna kyseisen isotopomeerin suhteellisen osuuden. Jos ohjelma ei saanut isotopomeerijakaumaa täysin ratkaistua (kuten yleensä käy), jotkut rivit sisältävät yhden positionaalisen isotopomeerin sijasta summan useammista positionaalisista isotopomeereistä.

Näin muotoiltuna tulostiedosto on luettavissa yleisimmillä taulukkolaskentaohjelmilla.

6 Käyttöliittymä

Ohjelmiston käyttöliittymä on toteutettu Matlabin käyttöliittymäkomponenttien avulla ja se mahdollistaa seuraavat asiat:

- syöte- ja tulostiedostojen valitsemisen
- mittauksessa käytetyn energiamäärän (eV) määrittelemisen
- mittauksessa käytetyn laitteen valitsemisen
- automaattisesti tunnistettujen fragmenttien atomikaavojen muuttamisen käyttäjän toimesta

Energiamäärän ja laitteen valinta on osittain automatisoitu. Käynnistyksen yhteydessä ohjelmisto päättelee senhetkisen työhakemiston sisältämien tiedostojen perusteella käyttäjän valittavissa olevat laitteet ja energiamäärät. Ohjelmiston käyttöliittymä näkyy kuvassa 3.

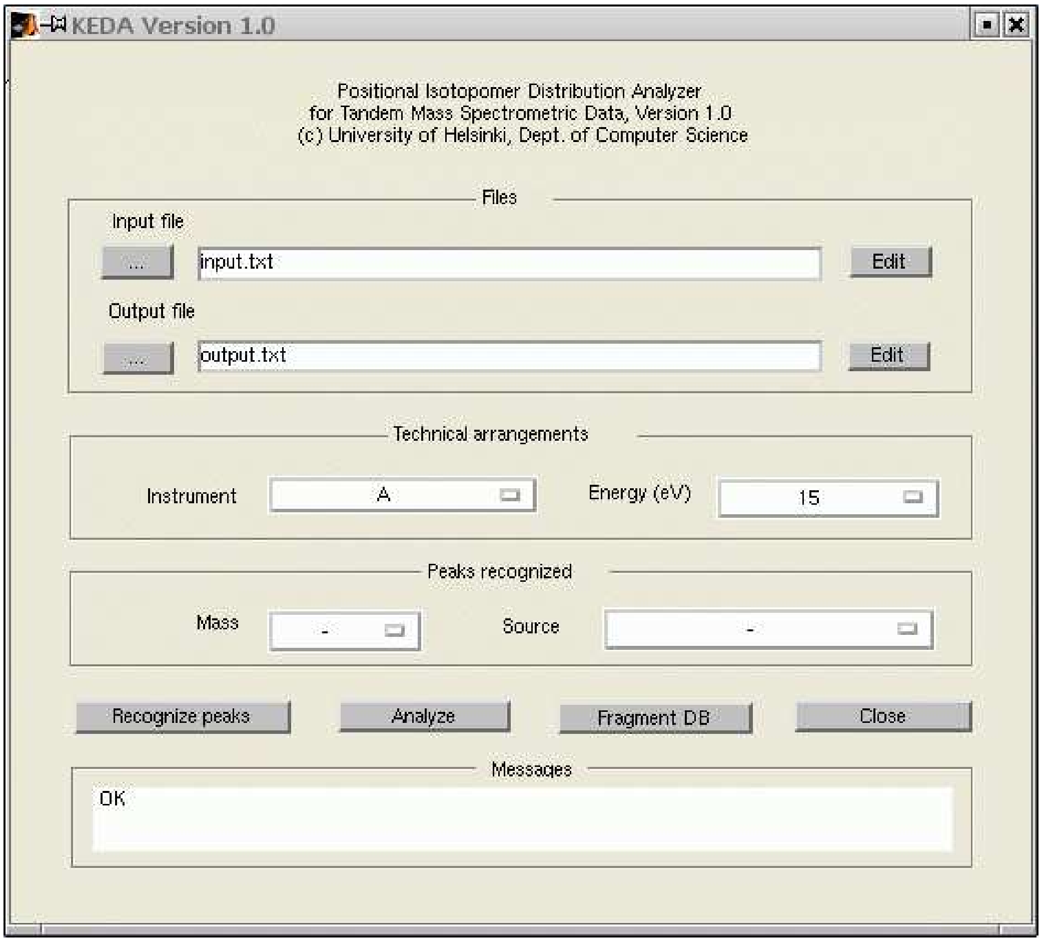
7 Moduulien kuvaus

Kustakin moduulista on kuvattu tehtävä, moduulit, joita kyseisestä moduulista kutsutaan sekä moduuliin kuuluvat funktiot.

7.1 Käyttöliittymä

Tehtävä

Toimii pääohjelmamoduulina ja ohjaa muita moduuleita.



Kuva 3: Ohjelmiston käyttöliittymä

Viittaukset

Piikkien tunnistus, syöte- ja tulostiedostojen käsittely, tietokannan luku ja päivitys, syntaksin tarkastus, laskenta

Funktiot

`KEDA`

`initializeDevAndEnergy`

`initializeGlobals`

7.2 Syntaksin tarkastus

Tehtävä

Tarkastaa, että syötetiedostossa on kaikki tarvittavat kentät ja että ne sisältävät oikeanlaista tietoa. Jos tietokantaa päivitetään, tarkastaa että uusi tieto on oikeanmuotoista.

Viittaukset

Ei ole.

Funktiot

`parseCarbons`

`parseFormula`

`parseInput`

`parseFDB`

7.3 Syöte- ja tulostiedostojen käsittely

Tehtävä

Hoitaa syötetiedoston ja fragmenttitietokannan lukemisen ja tulostiedoston kirjoittamisen.

Viittaukset

Ei ole.

Funktiot

`readFile`
`writeFile`

7.4 Piikkien tunnistus

Tehtävä

Tutkii, onko tietokannassa molekyylejä tai fragmentteja jotka sopivat annettuun dataan. Antaa ilmoituksen jos on tunnistamattomia piikkejä. Käyttäjällä on tällöin mahdollisuus täydentää tietokantaa.

Viittaukset

Tietokannan luku ja päivitys.

Funktiot

`recognizePeaks`

7.5 Tietokannan luku ja päivitys

Tehtävä

Etsii tietokannasta haluttua tietoa.

Viittaukset

Ei ole.

Funktiot

`findFragments`

7.6 Kerroinmatriisin luominen (äitispektri)

Tehtävä

Laskee äitispektrin perusteella kerroinmatriisin α-kertoimet. Lähettää sen jälkeen syötteen toiselle tai kummallekin fragmentteja tutkivalle moduulille. Kerroinmatriisi lasketaan luvussa 2.1 kuvatulla menetelmällä.

Viittaukset

Kerroinmatriisin luominen (kummatkin fragmentointitavat).

Funktiot

`createMotherMatrices`
`calculateIsotopomerProbability`
`calculateMultinomialProbability`
`changeCarbonsInFrIntoBinForm`
`changeIntIntoRelativeInt`
`countLabelsInCPartOfFr`
`countLabelsInCarbons`
`createAtomCombMatrix`
`createMotherMatrices`
`createTripleMatrix`
`getNonFrPartOfMol`
`getSuitableFragments`

7.7 Kerroinmatriisin luominen ("Full scan" -fragmentointi)

Tehtävä

Selvittää kunkin tytärpiikin (massa m) kohdalla, mistä erilaisista fragmenteista kyseinen piikki voi koostua. Laskee sitten kutakin hiiliyhdistelmää $\mathbf{w}$ vastaavan kertoimen ${}^{\mathbf{w}}\beta^{m}$ luvussa 2.2.1 kuvatulla tavalla.

Viittaukset

Tietokannan luku ja päivitys.

Funktiot

`createFsMatrices`
`calculateIsotopomerProbability`

```
calculateMultinomialProbability
changeCarbonsInFrIntoBinForm
changeIntIntoRelativeInt
countLabelsInCPartOfFr
countLabelsInCarbons
createAtomCombMatrix
createFsMatrices
createTripleMatrix
getNonFrPartOfMol
getSuitableFragments
```

7.8 Kerroinmatriisin luominen ("Daughter ion scanning")

Tehtävä

Muodostaa oman kerroinmatriisin kutakin äitispektrin piikkiä (leimoja l) kohti. Selvittää kyseisestä äitimolekyylistä syntyneitä tytärpiikkejä (m) vastaavat mahdolliset fragmentit ja laskee hiiliyhdistelmää $\mathbf{w}$ vastaavan kertoimen ${}^{\mathbf{w}}\beta^{m,l}$ luvun 2.2.2 esittämällä tavalla.

Viittaukset

Tietokannan luku ja päivitys.

Funktiot

```
createDisMatrices
calculateIsotopomerProbability
calculateMultinomialProbability
changeCarbonsInFrIntoBinForm
changeIntIntoRelativeInt
countLabelsInCPartOfFr
countLabelsInCarbons
createAtomCombMatrix
createDisMatrices
createTripleMatrix
```

`getNonFrPartOfMol`
`getSuitableFragments`

7.9 Laskenta

Tehtävä

Yhdistää kaikki muodostetut kerroinmatriisit ja niitä vastaavat tulosmatriisit. Ratkaisee näin saadusta yhtälöryhmästä hiilen positionaalisten isotopomeerien $\mathbf{w}$ todennäköisyydet $\mathbb{P}(^{\mathbf{w}}\{C\})$. Ratkaisua varten muodostettava yhtälöryhmä esitetään tarkemmin luvussa 2.3.

Viittaukset

Kerroinmatriisin luominen (äitispektri).

Funktiot

`main`
`calculate`

7.10 Apufunktiot

Tehtävä

Joukko apufunktioita, joita käytetään syntaksin tarkastuksessa ja kerroinmatriisien laskennassa.

Funktiot

`isDigit`
`isNumber`
`findAminoAcid`
`getIsotopeAbundances`

Päätemerkki	Merkitys
EMPTY	Tyhjä merkkijono (λ)
NL	Rivinvaihtomerkki
CHARACTER	Mielivaltainen merkki, joka ei ole rivinvaihto
INTEGER	Ei-negatiivinen kokonaisluku
FLOAT	Liukuluku
TAB	Tabulaattori
SPACE	Välilyönti
IUPAC-ABBREVIATION	Aminohapon kolmikirjaiminen IUPAC-lyhenne
FORMULA	Laillinen atomikaava ([KED02])
CARBON-CHAIN	Laillinen hiiliketju ([KED02])

Taulukko 1: Kielioppien päätemerkit

A Syntaksit EBNF-muodossa

Tässä luvussa esitellään syötetiedostojen ja fragmenttitietokannan tiedostojen syntaksit EBNF-notaatiolla esitettynä. Välikkeet on kirjoitettu pienellä, päätemerkit (tai sellaisiksi tulkittavat) kokonaan isoilla. Lisäksi tiedostoissa sellaisinaan esiintyvät literaalit esitetään tässä sitaattimerkeissä ('). Kieliopeissa käytettävät päätemerkit ja niiden merkitykset käyvät ilmi taulukosta 1

A.1 Syötetiedostot

```
input         ::= (input-line NL)+
input-line    ::= comment | peak | EMPTY
comment       ::= ws* '%' CHARACTER*
peak          ::= ws* mass ws+ intensity ws+ mother (comment | EMPTY)
mass          ::= INTEGER
intensity     ::= INTEGER
mother        ::= INTEGER | EMPTY | 'M' | 'm'
ws            ::= TAB | SPACE
```

A.2 Fragmenttitietokannan tiedostot

Seuraavassa on fragmenttietokannan tiedostojen EBNF-muotoinen syntaksi.

```
fdb           ::= (fdb-line NL)+
fdb-line      ::= comment | fragment | EMPTY
comment       ::= ws* '%' CHARACTER*
fragment      ::= ws* (mother | daughter) (comment | EMPTY)
mother        ::= mother-name ws+
                  role ws+
                  formula ws+
                  carbons ws+
                  mass
daughter      ::= mother-name ws+
                  role ws+
                  formula ws+
                  carbons ws+
                  mass ws+
                  rel-intensity
mother-name   ::= IUPAC-ABBREVIATION
role          ::= 'M' | 'D' | 'm' | 'd'
formula       ::= FORMULA
carbons       ::= CARBON-CHAIN
mass          ::= INTEGER
rel-intensity ::= FLOAT
ws            ::= TAB | SPACE
```

B Esimerkki syötetiedostosta

Tämä liite sisältää esimerkin siitä, miltä “daughter ion scanning”-menetelmällä tuotettu syötetiedosto näyttää.

```
% Measurements using Mass spectrometer A with Energy 30 eV
```

```
% were performed in Timbuktu 2002.

90 743899136       M
91 113999872       M
92 6172928         M
44 123844          1
45 174496          1
27 8794            1
28 13993           1
29 14401           1
44 50044           2
45 36667           2
46 21638           2
47 13342           2
```

C Esimerkki fragmenttitietokannan sisällöstä

Tässä liitteessä on lyhyt esimerkki siitä, miltä fragmenttitietokannan sisältö näyttää.

```
% Columns:
%    1. name of mother
%    2. role of ion (M=mother, D=daughter)
%    3. chemical formula for the ion
%    4. carbon atoms included in fragment
%    5. mass weight for the ion
%    6. relative abundance of the daughter ion
%       (intensity of peak) in MS spectrum.

Gly M C2H6NO2   C1C2      76
Gly D CH4N      C2        30  100
Gly D CH5N      C2        31  36
Gly D CH6NO     C2        48  2.1
```

```
Ala M C3H8NO2   C1C2C3    90
Ala D C2H3      C3C2      27  6.0
Ala D CH3N      C2        29  5.6
Ala D C2H6N     C3C2      44  100
Ala D C2H7N     C3C2      45  35
```

Lähteet

KED02 Ohjelmistotuotantoprojektiryhmä KEDA, Kemistin data-analysaattori, Ylläpitodokumentti. *Ohjelmistotuotantoprojekti, syksy 2002, Helsingin yliopisto, Tietojenkäsittelytieteen laitos*.

Ran01 Rantanen, A., PIDC (Positional Isotopomer Distribution Calculator), http://www.cs.helsinki.fi/research/icomic/PIDC/PIDC.zip

Ran02 Rantanen, A. et al., Computing Positional Isotopomer Distributions from Tandem Mass Spectrometric Data. *Metabolic Engineering*, hyväksytty.

RoR02 Rousu, J., Rantanen, A., Improved Computation of ^{13}C-Isotopomer Distributions from Tandem-MS Data.