Kemistin data-analysaattori

Suunnitteludokumentti (1. iteraatio)

Taisto Gerdt
Timo Kääriäinen
Mika Miettinen
Pasi Rastas
Jari Tuominiemi

Ohjelmistotuotantoprojekti
Helsingin yliopisto
Tietojenkäsittelytieteen laitos
4. lokakuuta 2002

Versiohistoria:

Päiväys	Muutokset	Muutosten tekijä
1.10.2002	Alustava versio	KEDA
2.10.2002	Ryhmäkokouksessa 1.10 ehdotetut korjaukset	M.M. ja P.R.

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

1.5 Käytettävyysvaatimukset 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 5

2.3 Lopullinen yhtälö 6

3 Tietosuunnitelma 6

3.1 Fragmenttitietokanta 6

3.1.1 Fragmenttitietokannan tietomallinnus 6

3.1.2 Fragmenttitietokannan funktionaalinen mallinnus . . 7

3.1.3 Fragmenttitietokannan ylläpitoon liittyvä käyttöliittymä 7

4 Arkkitehtuurisuunnitelma 7

5 Ulkoiset liittymät 10

5.1 Syötetiedosto 10

5.2 Tulostiedosto 10

6 Käyttöliittymä **11**

7 Moduulisuunnitelma **11**

7.1 Käyttöliittymä 13

7.2 Syntaksin tarkastus 13

7.3 Syöte- ja tulostiedostojen käsittely 13

7.4 Piikkien tunnistus 13

7.5 Tietokannan luku ja päivitys 13

7.6 Kerroinmatriisin luominen (äitispektri) 14

7.7 Kerroinmatriisin luominen (“Full scan” -fragmentointi) . . . 14

7.8 Kerroinmatriisin luominen (“Daughter ion scanning”) 14

7.9 Laskenta 14

8 Ohjelmointityyli **15**

8.1 Lähdekoodin muotoilu 15

8.2 Kommentointi 16

A Esimerkki fragmenttitietokannan sisällöstä **17**

Lähteet **17**

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

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

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

Kehitystyön pohjana ryhmä käyttää 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 toteutetaan kokonaan Matlabin avulla. Pääasiallinen toimivuus muodostuu joukosta Matlab-ohjelmointikielellä toteutettuja funktioita. Näiden lisäksi ohjelmiston käyttöliittymä tehdään Matlab-ohjelmiston käyttöliittymäkomponenttien avulla.

Koska yhdessäkään laitoksen mikroluokassa ei ole Matlab-ohjelmiston Windows-versiota, tekee ryhmä kehitystyön Linux-ympäristössä. Matlabin eri versioiden ei kuitenkaan odoteta sisältävän suuria käyttöjärjestelmäriippuvuuksia, joten tämän ei pitäisi muodostaa ongelmaa. Toteutettavan ohjelmiston lopullisen version on kuitenkin toimittava Windows-ympäristössä ja tämä varmistetaan muilla keinoin, mahdollisesti asiakkaan avulla. Koska kehitystyö suoritetaan Linuxissa, on odotettavaa että lopullinen ohjelmisto toimii sekä Linux- että Windows-ympäristöissä. Näiltä osin ohjelmisto jopa ylittäisi alkuperäiset vaatimuksensa.

1.5 Käytettävyysvaatimukset

Ohjelman toiminta ei edellytä työasemalta erityisvaatimuksia. Mikä tahansa Windows 2000 -käyttöjärjestelmällä varustettu mikrotietokone, joka kykenee ajamaan Matlab-ohjelmistoa riittää. On mahdollista että ohjelma valmistuttuaan toimii myös muissa käyttöjärjestelmissä, joihin Matlab-ohjelmisto on saatavissa.

Ohjelman mahdollisista virhetoiminnoista käyttäjä saa välittömästi tiedon näytölle ja ohjeet virheen korjaamiseksi.

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):

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

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 ylläolevaan summaan, saadaan

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

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

$$\iota_D(t) = \sum_{{}^{+k}\{F\}\in\mathcal{F}(t)} \mathbb{P}({}^{+k}\{F\}) \cdot \frac{\mathbb{D}_E(F)}{\sum_{F_i\subset M}\mathbb{D}_E(F_i)}. \tag{1}$$

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

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

Sijoittamalla tämä yhtälöön 1 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 siten

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

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

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

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 2 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 Tietosuunnitelma

3.1 Fragmenttitietokanta

3.1.1 Fragmenttitietokannan tietomallinnus

Fragmenttitietokanta toteutetaan tiedostopohjaisena siten, että kutakin massaspektrometrissä käytettyä energiaa ja laitteistoa tai teknistä järjestelyä kohden on oma tekstitiedostonsa. Tiedostossa esitellään sarakkeittaen 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 fragmentin aiheuttaman piikin paikkaa suoraan tiedostostosta. Viimeisessä sarakkeessa esitetään ao. ionin suhteellinen osuus kyseisellä energialla. Liitteessä A on esimerkki fragmenttitietokannan sisällöstä 15 eV energialle.

3.1.2 Fragmenttitietokannan funktionaalinen mallinnus

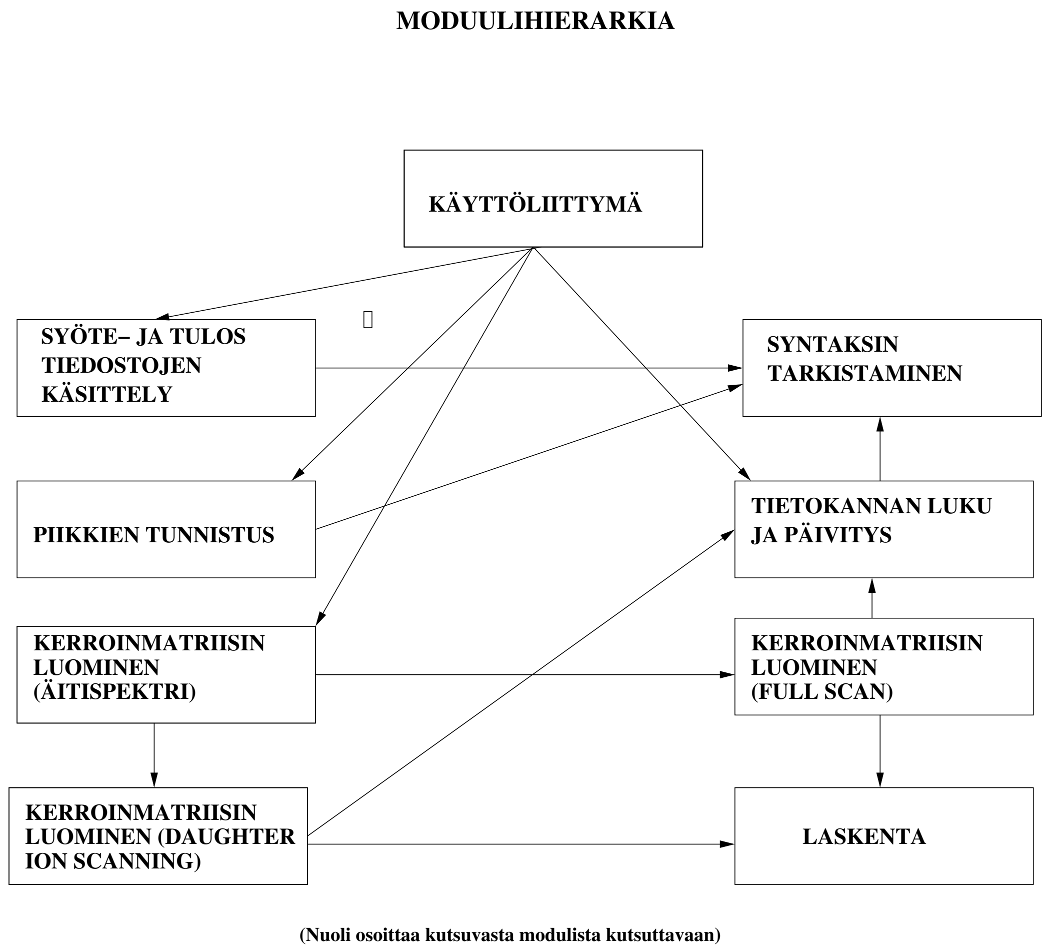
Ohjelmiston käynnistyksen yhteydessä järjestelmä valitsee syötetiedoston energia-arvoa vastaavan fragmenttitietokannan tai käyttäjä valitsee haluamansa järjestelmän tarjoamista vaihtoehdoista. Järjestelmä lukee tietokannan tarvittavat osat muuttujien arvoiksi, joita voidaan käyttää laskennassa. Muuttujien järkevyys tarkistetaan ennen laskentaa. Virheellistiä tai puuttuuvia arvoja löydettäessa käyttäjää informoidaan virheestä. Käyttäjä pystyy käyttämään laskennassa apuna myös osin tietokannasta poikkeavaa tietoa - kuten jättämään käsittelemättä tiettyä fragmenttia.

3.1.3 Fragmenttitietokannan ylläpitoon liittyvä käyttöliittymä

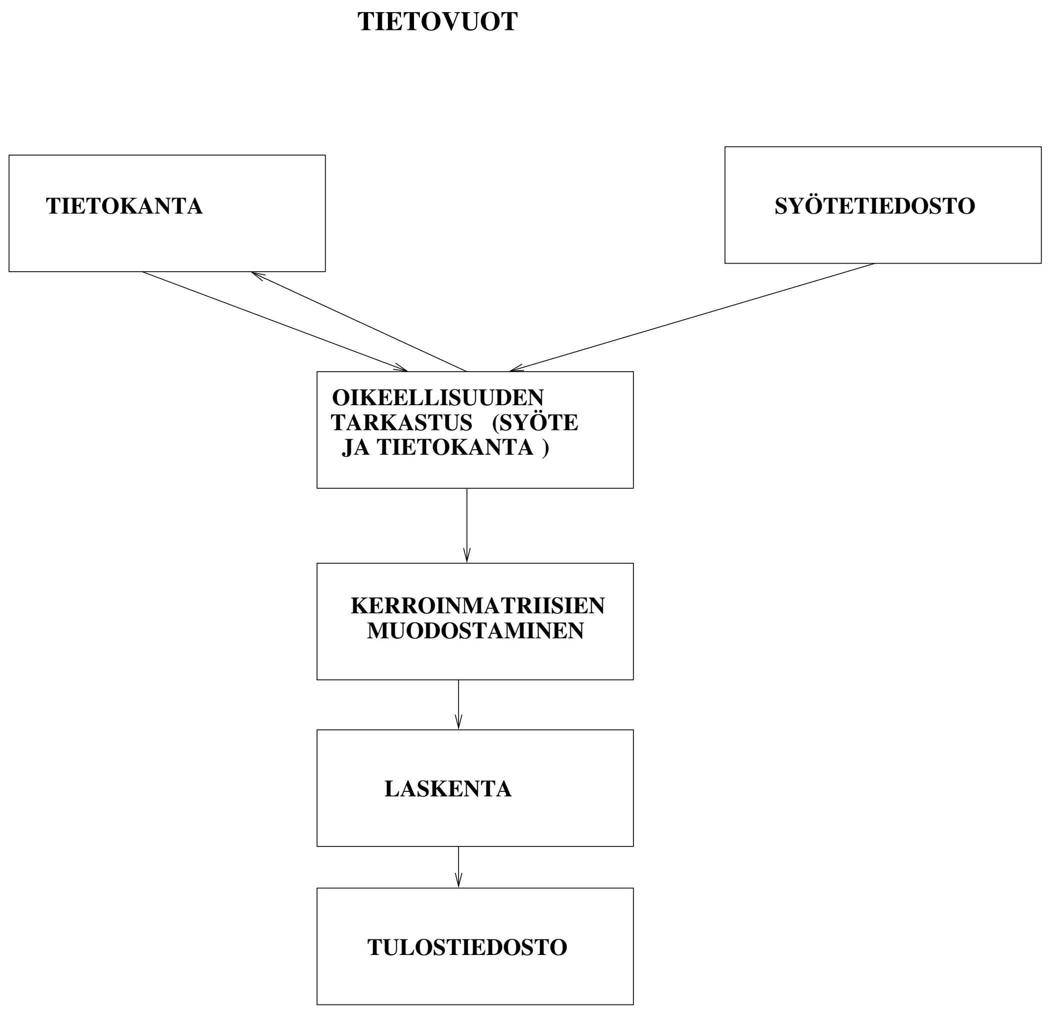
Käyttöliitymä toteutetaan Matlab-ohjelmistoon käyttöliittymän suunnittelutyökalulla. Se mahdollistaa tietyyn energiaan liittyvä äiti- ja tytärionien tietojen muuttamisen. Tietoja voidaan myös lisätä ja poistaa tiedostosta samoin kuin luoda uusia tiedostoja esim. eri massaspektrometrejä ja energioita varten.

4 Arkkitehtuurisuunnitelma

Ohjelmiston moduulihierarkia näkyy kuvassa 1 ja tietovuokaavio kuvassa 2.



Kuva 1: Ohjelmiston moduulihierarkia



Kuva 2: Ohjelmiston tietovuokaavio

5 Ulkoiset liittymät

5.1 Syötetiedosto

Ohjelman lukema syötetiedosto sisältää massaspektrometrin mittauksista saadun datan. Käsittelyn helpottamiseksi tiedoston oletetaan sisältävän vain yhden äitimolekyylin tiedot. Syötetiedosto sisältää rivejä, joista kukin kuvaa yhtä massaspektrin piikkiä. Kullakin rivillä on kolme tabulaattorein 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 massaspektri on tuotettu "Daughter ion scan"-menetelmällä, on sarakeessa kokonaisluku. Tällöin kyseinen piikki kuvaa tytärionia ja luku kertoo mistä äitimolekyylin isotopomeeristä kyseinen fragmentti on peräisin. Syötetiedoston muoto EBNF-notaatiolla esitettynä on siis (`NL` on rivinvaihto, `TAB` tabulaattori, `INTEGER` ei-negatiivinen kokonaisluku ja `EMPTY` tyhjä merkkijono)

```
input         ::= mass-spectrum
mass-spectrum ::= peak NL (peak NL)*
peak          ::= mass TAB intensity TAB mother
mass          ::= INTEGER
intensity     ::= INTEGER
mother        ::= INTEGER | EMPTY | 'M'
```

5.2 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 $[b_1 b_2 ... b_n]$ 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

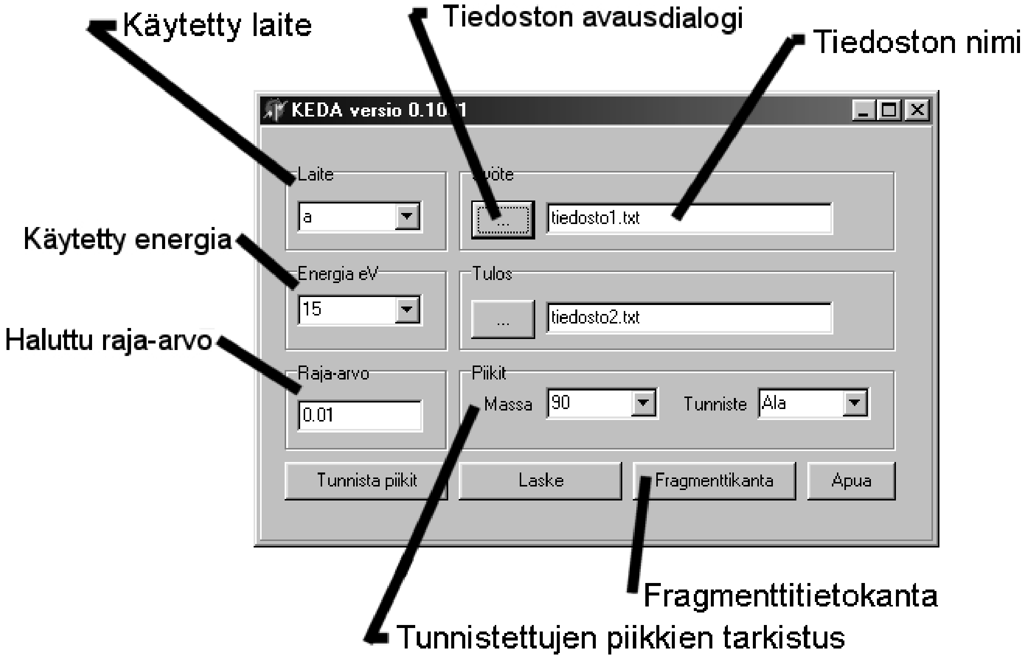
Ohjelmiston käyttöliittymä toteutetaan Matlabin käyttöliittymäkomponenttien avulla ja siitä pyritään tekemään mahdollisimman selkeä ja helppokäyttöinen. Käyttöliittymä mahdollistaa seuraavat asiat:

- syöte- ja tulostiedostojen valitsemisen
- mittauksessa käytetyn energiamäärän (eV) määrittelemisen
- mittauksessa käytetyn laitteen valitsemisen
- automaattisesti tunnistettujen fragmenttien atomikaavojen muuttamisen käyttäjän toimesta
- antaa käyttäjän määritellä raja-arvon, jonka alle jäävät syötetiedoston piikit jätetään laskennassa huomiotta.

Suunnitelma ohjelmiston käyttöliittymäksi on kuvassa 3.

7 Moduulisuunnitelma

Kustakin moduulista on kuvattu tehtävä ja moduulit, joita kyseisestä moduulista kutsutaan. Tarkempaan kuvaukseen (mm. sisäiset tietorakenteet) ei tässä ole mahdollisuuksia.



Kuva 3: Ohjelmiston suunniteltu käyttöliittymä

7.1 Käyttöliittymä

Tehtävä Toimii pääohjelmamoduulina ja ohjaa muita moduuleita.

Viitatut moduulit Piikkien tunnistus, syöte- ja tulostiedostojen käsittely, tietokannan luku ja päivitys, kerroinmatriisin luominen (äitispektri).

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.

Viitatut moduulit Ei ole.

7.3 Syöte- ja tulostiedostojen käsittely

Tehtävä Hoitaa syötetiedoston lukemisen ja tulostiedoston kirjoittamisen.

Viitatut moduulit Syntaksin tarkastus.

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.

Viitatut moduulit Tietokannan luku ja päivitys.

7.5 Tietokannan luku ja päivitys

Tehtävä Etsii tietokannasta haluttua tietoa ja lisää sinne tarvittaessa uusia molekyylejä ja fragmentteja.

Viitatut moduulit Syntaksin tarkastus

7.6 Kerroinmatriisin luominen (äitispektri)

Tehtävä Laskee äitispektrin perusteella kerroinmatriisin α-kertoimet. Lähettää sen jälkeen syötteen toiselle tai kummallekin fragmentteja tutkivalle moduulille.

Viitatut moduulit Kerroinmatriisin luominen (kummatkin fragmentointitavat).

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 ${}^{\mathbf{w}}\beta^{m}$.

Viitatut moduulit Tietokannan luku ja päivitys, laskenta.

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 ${}^{\mathbf{w}}\beta^{m,l}$.

Viitatut moduulit Tietokannan luku ja päivitys, laskenta.

7.9 Laskenta

Tehtävä Yhdistää kaikki muodostetut kerroinmatriisit ja niitä vastaavat tulosmatriisit. Ratkaisee näin saadusta yhtälöryhmästä hiilen positionaalisten isotopomeerien $\mathbf{w}$ todennäköisyydet $\mathbb{P}({}^{\mathbf{w}}\{C\})$.

Viitatut moduulit Ei ole.

8 Ohjelmointityyli

8.1 Lähdekoodin muotoilu

Yleisohjeena on että kukin funktio tekee vain yhden loogisen asiakokonaisuuden. Ohjelmiston ylläpidon helpottamiseksi pyritään kaikista funktioista edelleen tekemään mahdollisimman selkeitä ja yksinkertaisia. Lisäksi lähdekoodin muotoilussa noudatetaan seuraavia ohjeita:

- tunnukset ja kommentit englanniksi
- tunnusten nimeämistapa on Java APIn mukainen
- globaalien muuttujien nimet pitkiä ja kuvaavia; paikallisten muuttujien nimet mieluiten lyhyitä (lisäksi globaalit muuttujat erotetaan muista isolla alkukirjaimella)
- sisennykset neljä välilyöntiä
- rivin pituus (sisennykset mukaanlukien) mieluiten alle 80 merkkiä
- funktion pituus mieluiten alle 40 riviä

Näiden lisäksi käytetään ohjelmarakenteiden muotoilussa seuraavia konventioita

```
% for-silmukka
for loop
    % statements
end

% if-else-rakenne
if condition1
    % statements1
elseif condition2
```

```
    % statements2
else
    % statements3
end

% while-silmukka
while condition
    % statements
end

% switch-rakenne
switch switch_expression1
    case case_1
        % statements_1
    ...
    case case_n
        % statements_n
    otherwise
        % statements
end
```

8.2 Kommentointi

Kommentoinnissa on tärkeää että kunkin funktion alkuun (*heti* funktion esittelyrivin jälkeen) kirjoitetaan lyhyt kommentti kyseisen funktion merkityksestä. Tällöin Matlabin help-toiminto osaa näyttää kyseisen kommentin automaattisesti opastuksena käyttäjälle. Muussa kommentoinnissa pitäisi pyrkiä selittämään *mitä* tehdään, eikä sitä *miten* jokin asia tehdään. Toteutustavan tulisi näkyä jo lähdekoodin tasolla. Pidemmät kommentit on myös syytä kirjoittaa omille riveilleen ennen kommentoitavaa ohjelmakohtaa.

A Esimerkki fragmenttitietokannan sisällöstä

Tässä liitteessä on lyhyt esimerkki siitä, miltä fragmenttitietokannan sisältö näyttää. Fragmenttien syntymistodennäköisyydet ovat keksittyjä.

```
% 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  0.50
Gly D CH6NO     C2        48  0.50

Ala M C3H8NO2   C1C2C3    90
Ala D C2H3      C3C2      27  0.00
Ala D CH3N      C2        29  0.00
Ala D C2H6N     C3C2      44  1.00
```

Lähteet

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.