Liet. matem. rink., **44**, *spec. nr.*, 2004, 774–778

Atsitiktinio amino rūgščių išsidėstymo sekoje matematiniai modeliai

Alvydas ŠPOKAS (BTI, VGTU), Albertas TIMINSKAS (BTI)
el. paštas: alvydas@ibt.lt, timis@ibt.lt

Įvadas

Unikalias baltymo biologines savybes lemia specifinis amino rūgščių išsidėstymas peptidinėje sekoje, suformuojant tam tikromis savybėmis pasižyminčius sekos fragmentus, kurie daro pagrindinę įtaką baltymų erdvinės struktūros susidarymui bei jų biologinei funkcijai. Taigi baltymo funkcija labiausiai priklausoma nuo baltymo pirminės amino rūgščių sekos. Remiantis šiuo teiginiu kuriamos peptidinių sekų analizės programos sėkmingai sprendžiančios baltymų aukštesnių struktūrinių būsenų nuspėjimą ar baltymų panašumų paiešką bei jų įvertinimą [1, 2].

Atsitiktinė amino rūgščių seka vadinsime tokia seką, jei bet kurioje jos fragmento (santykinai neilgo) bet kurio elementų perstatymas yra vienodai tikėtinas. Tačiau dėl amino rūgščių savybių suderinamumo baltymo sekoje jų išsidėstymas nėra atsitiktinis, o paremtas tam tikrais dėsningumais [3]. Siekiant įvertinti baltymų sekų išskirtinumą būtina rasti tinkamą atsitiktinių sekų modeliavimo mechanizmą.

Šiame darbe tirtas restrikcijos fermentų rinkinys paimtas iš Rebase duomenų bazės [4]. Rinkinys papildomai išvalytas nuo informacijos šiukšlių, kurias sudaro giminingos, labai panašios arba hibridinės sekos turinčios labai panašaus pobūdžio baltymų sekose slypinčią informaciją [5]. Galutiniame rinkinyje paliktos tik tos sekos, kurių tarpusavio panašumas neviršija bendra-baltyminio panašumo, būdingo bet kuriems baltymams.

Analizuojamą baltymų imtį sudaro 152 restrikcijos fermentų peptidinės sekos, kurių ilgiausia seka turi $l_{\max} = 490$, trumpiausia $l_{\min} = 200$ amino rūgščių. Kiekvienoje sekoje yra 20 skirtingų amino rūgščių sunumeruotų tam tikra tvarka, kurių kiekis svyruoja nuo 3 iki 8 nuošimčių, o bendras amino rūgščių skaičius imtyje 46735.

Tikimybinis metodas

Kiekvieną baltymo peptidinę seką galima nagrinėti kaip tos sekos tam tikro ilgio l fragmentų aibę [6], kur kiekvienas fragmentas

$$S = (s_1, s_2, \ldots, s_l), \quad \text{kai } s_i \in= \{1, 2, \ldots, 20\}. \tag{1}$$

Fragmentų variantų skaičius labai greitai auga didėjant l, kaip 20^l. Vienos amino rūgšties ilgio fragmentų (kai $l = 1$) susidarymas sprendžiamas ganėtinai paprastai,

kadangi iš anksto yra žinomi kiekvienos amino rūgšties pradiniai kiekiai baltyme, o žinant analizuojamo baltymo ilgį nesunkiai apskaičiuojamos amino rūgščių aptikimo bei fragmento susidarymo tikimybės

$$P(s_i) = \frac{n_i}{L}; \tag{2}$$

čia n_i – pradinis i amino rūgšties kiekis baltyme, L – baltymo ilgis (pradinis amino rūgščių kiekis).

Esant dviejų ar daugiau amino rūgščių ilgio fragmentui jo susidarymo tikimybė gali būti taip pat paprastai įvertinama padarius prielaidą, kad amino rūgščių pradinės tikimybės nekinta ir jos papuola į fragmentą nepriklausomai viena nuo kitos, t.y., vykdant fragmento sudarymą su amino rūgščių gražinimu į pradinę imtį. Tokiu būdu fragmento varianto S_l tikimybė randama iš fragmento elementų parinkimo tikimybių sandaugos:

$$P(S_f) = P(s_1, s_2, \ldots, s_l) = P(s_1) \times P(s_2) \times \cdots \times P(s_l). \tag{3}$$

Tačiau praktiškai amino rūgščių kiekiai baltyme yra baigtiniai, dėl to dviejų ir daugiau amino rūgščių fragmento išrinkimas turi būti vykdomas be grąžinimo ir atitinkamai kiek kitaip skaičiuojama susidariusio fragmento tikimybė. Kadangi pirmoji fragmento amino rūgštis yra fiksuojama (negražinama) tai dėl bendro amino rūgščių kiekio ir išrinktos amino rūgšties kiekio sumažėjimo automatiškai keičiasi neįtrauktų amino rūgščių tikimybės, tokiu atveju 2 lygtyje naudoti pažymėjimai keičia prasmę:

$$P'(s_i) = \frac{n'_i}{L'}; \tag{4}$$

čia n'_i – likutinis i amino rūgšties kiekis baltyme, L' – likutinis visų amino rūgščių kiekis.

Išrinkimo su grąžinimu atveju $P(s_i)$ apskaičiuojamas vieną kartą ir visuomet išlieka pastovus, o išrinkimo be grąžinimo atveju $P'(s_i)$ perskaičiuojamas po kiekvienos amino rūgšties išrinkimo. Pvz., išrinkus vieną amino rūgštį s_i likusių amino rūgščių išrinkimo tikimybės bus

$$P'(s_i) = \frac{n_i}{L-1}, \quad \text{jei} \quad s_i \neq s_i^{(-1)}; \quad P'(s_i) = \frac{n_i - 1}{L-1}, \quad \text{jei} \quad s_i = s_i^{(-1)}, \tag{5}$$

čia $s^{(-1)}$ – prieš tai išrinkta amino rūgštis.

Analogiškai apskaičiuojamos likutinės amino rūgščių sąlyginės tikimybės po antros amino rūgšties išrinkimo, po trečios, ir t.t. Faktiškai šiuo atveju vertinamo fragmento pasitaikymo tikimybė aprašoma daugiamačiu hipergeometriniu skirstiniu [7]. Tokiu būdu išspręstas atsitiktinės amino rūgščių sekos uždavinys bus absoliučiai tikslus, bet efektyvumo požiūriu dėl savo hierarchijos bei sąlyginių lygčių sistemų kiekio nėra patogus naudoti. Minėtam peptidinių sekų rinkiniui naudojant šiuolaikinius kompiuterius pavyko suskaičiuoti tikimybes tik 1, 2, 3, 4 ir 5 amino rūgščių ilgio fragmentams. Didesnių fragmentų skaičiavimai nebuvo atliekami dėl labai didelių laiko sąnaudų.

Supaprastintas matematinis modelis aprašomas 2, 3 lygtimis besąlygiškai yra greičiausiais, kadangi reikalauja elementarios kelių narių sandaugos. Tokio algoritmo

atlikimo trukmę tik nežymiai įtakoja fragmento ilgio didinimas. Neigiamo šio metodo savybė yra nepakankamas tikslumas susijęs su sistemine paklaida atsiradusia būtent dėl algoritmo supaprastinimo neperskaičiuojant likutinių amino rūgščių išrinkimo tikimybių. Didėjant analizuojamų fragmentų ilgiui sisteminė paklaida taip pat didėja, dėl ko šį metodą patogu naudoti tik preliminariems skaičiavimams.

Sekų generatorius

Kaip alternatyva matematiniam modeliui taip pat buvo nagrinėjamas bioinformatikoje gerai žinomas atsitiktinių peptidinių sekų generavimas [8] paremtas atsitiktinio perkėlimo (*angl.* random shuffle) algoritmu. Pagal šį algoritmą einamojoje ir naujai sugeneruotoje pozicijose esančios amino rūgštys sukeičiamos vietomis. Naujai generuojama pozicija gali būti bet kurioje sekos vietoje, tuo tarpu einamoji pozicija slenka iš eilės nuo sekos pradžios iki pabaigos. Procedūra atliekama kiekvienai sekos pozicijai (sumaišymo ciklas), dėl ko amino rūgštys perkeliamos į naują poziciją maždaug vieną kartą.

Būtina paminėti, kad tokiu būdu sugeneruotos sekos nėra visiškai atsitiktinės, kadangi rezultatas dalinai priklauso nuo pradinio amino rūgščių išsidėstymo. Šią priklausomybė labai gerai iliustruoja pavyzdys. Tarkime, kad pradinę peptidinę seką sudaro tik trys amino rūgštys ABC. Pagal aprašytą algoritmą bus atliekami trys amino rūgščių sukeitimo etapai. Po pirmojo etapo galimi tik trys sekos variantai: ABC, BAC ir CBA. Po sekančio etapo gaunami 9, o po trečio – 27 sekų variantai, tarp kurių yra tik 6 skirtingi. Akivaizdu, kad 6 variantams iš 27 galimų susidaryti tikimybės nėra vienodos (ABC – 5/27, ACB – 4/27, BAC – 5/27, BCA – 5/27, CAB – 4/27, CBA – 4/27). Bendru atveju galutinių variantų tikimybės apytikriai gali būti išreikštos lygtimi:

$$P = \frac{n!}{n^n}, \tag{6}$$

kur n – maišomos sekos ilgis.

Galutinių variantų pasiskirstymą lemia pradinė seka, dėl ko aprašytu metodu gautos sekos nėra visiškai atsitiktinės, nors ilgesnėms sekoms ši paklaida yra nykstamai maža, o visiškai išnyksta po kelių maišymo iteracijų, kai kiekvienas sekantis maišymas atliekamas prieš tai sumaišytai sekai.

Pradinės baltymų sekos bei sugeneruotos atsitiktinės sekos analizuojamos tam tikrais metodais ir įvertinamas pradinių baltymo sekų išskirtinumas. Siekiant išvengti atsitiktinių sekų sudarymo paklaidų, tai pačiai sekai atliekama keletą sumaišymo ciklų, kiekvieną kartą seką analizuojant. Kiekvienas sekantis sumaišymo ciklas atliekamas ne su pradine baltymo seka, o su prieš tai vykdytame cikle sumaišyta seka. Tokiu būdu gauti analizės rezultatai artimi gautiems matematinių modelių pagalba, o tikslumas tiesiogiai priklauso nuo atliekamų sumaišymo ciklų skaičiaus.

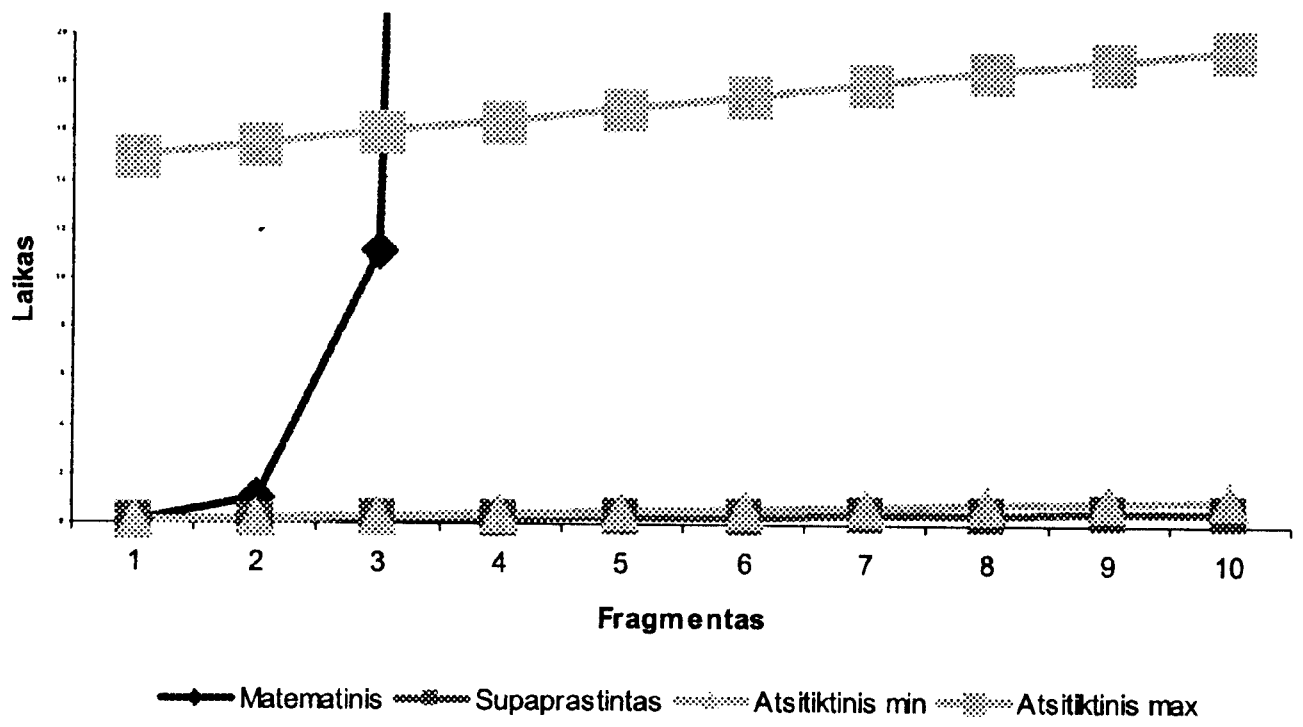
Vertinant atsitiktinių sekų generavimą susiduriama su dvilypiu efektyvumu, kai labai geras tikslumas pasiekiamas didelių skaičiavimų pagalba. Kaip jau minėta pagrindinė šio metodo neigiama savybę – atsitiktinės paklaidos, kurias maksimaliai sumažinti įmanoma tik daug kartų kartojant eksperimentą, tuo pačių naudojamas skaičiavimų laikas. Teigiama tokio metodo savybė yra tai, kad galima lanksčiai reguliuoti tikslumą skaičiavimo laiko sąnaudų sąskaita ir ribiniais atvejais priartėti prie vieno arba kito matematinių modelių.

Rezultatai

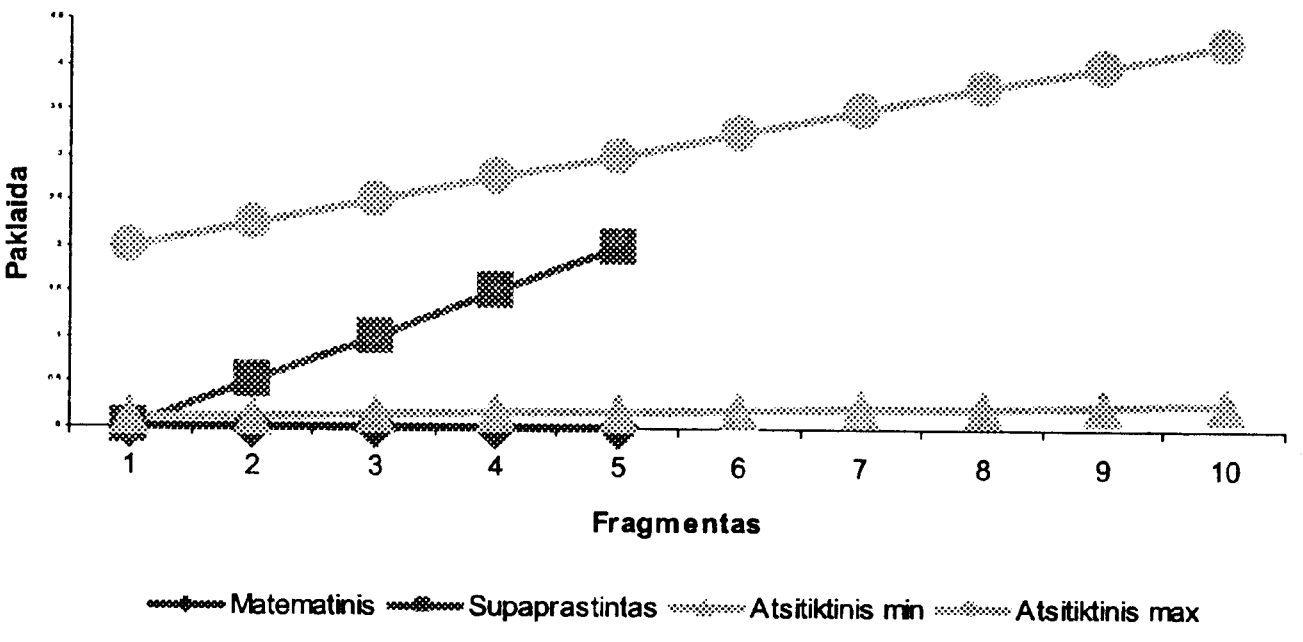
Eksperimentiškai pritaikant kiekvieną iš nagrinėtų atsitiktinių sekų gavimo būdų gauti laiko sąnaudų (1 pav.) bei paklaidos dydžių (2 pav.) įvertinimo rezultatai. Atsitiktinių sekų generavimo atveju laikui bei tikslumui didžiausią reikšmę turi atliekamų maišymo ciklų skaičius, tai ir rezultatuose pateikiami rėžiai atitinkantis mažiausią (vieną) ir didžiausią (šiame uždavinyje 1000) maišymo ciklų skaičių.

Kaip jau buvo minėta absoliučiai tikslus yra matematinis modelis (išrinkimas be grąžinimo), kadangi atsikartoja su nuline paklaida. Tačiau dėl staigiai didėjančių laiko sąnaudų (1 pav.) šis metodą naudoti esant vidutiniams fragmento ilgiams neefektyvu, o esant dideliems fragmentų ilgiams (nuo 5–6) fiziškai neįmanoma.

Supaprastinto matematinio modelio (išrinkimas su grąžinimu) atsikartojamumas taip pat pasižymi stabilumu, tačiau rezultatai yra veikiami sisteminės paklaidos, kuri didėja kartu su analizuojamo fragmento ilgiu. Siekiant įvertinti sisteminę paklaida gauti rezultatai buvo lyginami su matematinio modelio rezultatais (2 pav.). Taip pat pastebėta, kad sisteminės paklaidos dydis priklauso nuo analizei pasirinktų sekų rinkinio, todėl sunku numatyti sistemines paklaidas kitiems baltymų rinkiniams su kitokiu amino rūgščių balansu arba sekų ilgiu.



1 pav. Skirtingų algoritmų atliekamų skaičiavimų santykinės laiko sąnaudos.



2 pav. Skirtingų algoritmų atliekamų skaičiavimų santykinės paklaidos.

Atsitiktinių sekų generavimo atveju paklaida buvo vertinama atliekant 10 to paties taško skaičiavimų (uždavinio kartojimų) keičiant sekų sumaišymo ciklų skaičių. Didžiausios paklaidos gaunamos atliekant tik vieną sekų sumaišymo ciklą, o artimos nuliui paklaidos gaunamos atliekant daug atsitiktinių sekų generavimo ciklų. Grafike parodyti rezultatai atitinka 1 (Atsitiktinis max) ir 1000 (Atsitiktinis min) atsitiktinių sekų generavimo ciklų.

Apibendrinimas

Kiekvieno iš trijų naudotų algoritmų trukmės sąnaudos gali būti išreikštos lygtimis (7), pagal kurias įmanoma apytikriai palyginti nagrinėtų skaičiavimo algoritmų trukmes:

$$t_M \approx T^f; \quad t_{M'} \approx T \times f; \quad t_A \approx T \times n; \tag{7}$$

kur f – fragmento ilgis, n – kartojimų skaičius.

Paklaidos taip pat gali būti išreikštos panašiomis lygtimis (8), apytikriai nusakančios šių algoritmų paklaidas

$$\Delta_M = 0; \quad \Delta_{M'} \approx \Delta \times f; \quad \Delta_A \approx \frac{\Delta}{n}. \tag{8}$$

Pagal šias lygtis įmanoma pasirinkti konkrečiam fragmento ilgiui tinkamiausią skaičiavimo algoritmą atsižvelgiant į reikiamą tikslumą ir laiko sąnaudas. Kalbant apie atsitiktinių sekų generavimo algoritmą paklaida ir trukmė tiesiogiai priklausomi nuo generavimų skaičiaus, todėl naudojant šį algoritmą įmanoma parinkti gautų rezultatų tikslumą skaičiavimo trukmės sąskaita.

Literatūra

1. *Bioinformatics*: *Sequence and Genome Analysis*, CSHL Press (2000).
2. *Bioinformatics*: *Sequence, Structure, and Databanks*, Oxford Press (2000).
3. S.H. White, R.E. Jacobs, The evolution of proteins from random amino acid sequences, I. Evidence from the lengthwise distribution of amino acids in modern protein sequences, *J. Mol. Evol.*, **36**(1), 79–95 (1993).
4. `http://rebase.neb.com/rebase/rebase.html`.
5. V. Pingoud, E. Kubareva, G. Stengel, P. Friedhoff, J.M. Bujnicki, C. Urbanke, A. Sudina, A. Pingoud, Evolutionary relationship between different subgroups of restriction endonucleases, *J. Biol. Chem.*, **277**(16), 14306–14 (2002).
6. Д. Гасфилд, Строки, деревья и последовательности в алгоритмах, *Информатика и вычислительная биология* (2003).
7. J. Kruopis, *Matematinė statistika*, Mokslas, Vilnius (1993).
8. `http://www.cbio.psu.edu/sms/index.html`.

SUMMARY

A. Špokas, A. Timinskas. Mathematical models of random arrangement of amino acids in the sequence

Mathematical methods which estimate the arrangement of amino acids in the random sequence where analyzed here. The random sequence of amino acids can be used as a reading point for protein's similarities. It is important to choose rapid and accurate model for imitation of composition of random sequences.

Keywords: random sequence, amino acids.