VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
MATEMATINĖS INFORMATIKOS KATEDRA

Eimantas Paspirgėlis

Bioinformatikos studijų programa

Matematinės informatikos šaka

Baltijos jūros vandens mikroorganizmų genominių sekų tyrimas

Study of Baltic Sea Microbial Genomic Sequences

Bakalauro baigiamasis darbas

Vadovas: lekt. Irus Grinis

Recenzentas: dr. Saulius Gražulis

Vilnius 2015

Turinys

Įvadas

Baltijos jūroje buvo atlikti tyrimai, kurių metu buvo paimtas Baltijos jūros vandens mėginys. Jame buvo rasta įvairių mikroorganizmų, penki iš jų (3 archėjos ir 2 bakterijos) buvo dalinai sekvenuotos ir gauti jų nepilni genomai. Identifikavus mikroorganizmus, bakalauro darbui buvo parinktas vienas iš minėtų genomų detalesnei analizei.

Bakalauro darbe bus analizuojamas bakterijos *Sulfurimonas denitrificans* genomas. Analizė atliekama su trimis skirtingais įrankiais *MyRast, Prokka* ir *GeneMark*. Tai anotacijos įrankiai, su kuriais *Sulfurimonas denitrificans* genomas bus anotuojamas. Gavus visų trijų įrankių rezultatus jie bus lyginami tarpusavyje tam, kad surasti ir išrinkti tiksliausias baltymų sekas, kurios buvo gautos naudojant anotacijos įrankius. Taip pat bus ieškomi fermentai ir galimi jų keliai (angl. pathway). Visas identifikuotas baltymų sekas bus galima pavaizduoti vadinamojoje „genomo naršyklėje“, kurią sugeneruos *Python* kalba parašytas skriptas.

Taigi, darbo galutinis tikslas yra sukurti genomo pavaizdavimą naršyklėje (angl. genome browser) įprastinėmis priemonėmis naudojant skripto kalbas ir *html*.

Darbo tikslui pasiekti keliami uždaviniai:

- panagrinėti, iš vartotojo pusės, bioinformatikos anotacijos įrankius: *MyRast*, *Prokka* ir *GeneMark*;
- gauti automatinę *Sulfurimonas denitrificans* genomo anotaciją naudojant aukščiau paminėtus įrankius;
- sukurti skriptus, kurie:
    - palygintų tarpusavyje visų trijų įrankių gautus rezultatus;
    - surastų fermentus ir galimus jų kelius;
    - pagal *MyRast* išvesties failą automatiškai generuotų *html* kodą, kuris vaizduoja genomą naršyklėje.

Teorinėje darbo dalyje bus apžvelgiama tiriamo organizmo trumpa charakteristika, naudojami įrankiai: *Python, MyRast, Prokka* ir *GeneMark*. Praktinėje dalyje bus apžvelgiami naudojamų įrankių gauti rezultatai, sukurti skriptai ir jų rezultatai, taip pat sugeneruotas *Sulfurimonas* atvaizdavimas naršyklėje.

1.Teorinė dalis

1.1. Tiriamo organizmo trumpa charakteristika

1.1.1. Baltijos jūroje atliktas tyrimas

Darbe [1] Baltijos jūros vandenyse buvo atlikti tyrimai, kurie pateikia tam tikrą informaciją apie paviršutiniuose vandenyse gyvenančius mikroorganizmus, jų populiaciją. Taip pat atlikti tyrimai [2], kuriuose tiriamas bakterijų pasiskirstymas horizontaliai ir vertikaliai.

Šiame tyrime buvo bandoma ištirti mikroorganizmus, gyvenančius ne paviršutiniuose vandenyse, o gilesniame, sūresniame vandenyje įsikūrusius prokariotus. Todėl Baltijos jūroje maždaug iš 30 metrų gylio buvo paimtas vandens mėginys. Atlikus tyrimus, mėginyje buvo rasta įvairiausių bakterijų ir archėjų. Penki iš jų (2 bakterijos ir 3 archėjos) buvo siunčiamos į Jungtinių Amerikos Valstijų „Bigelow Laboratory for Ocean Sciences“ laboratoriją, kad būtų sekvenuoti jų genomai. Atlikus sekvenavimą, genomai buvo parsiųsti atgal į Lietuvą ir čia tiriami.

1.1.2. Sekvenuotų genomų apžvalga

Šiame tyrime buvo išsiaiškinta, kad minėti prokariotai, manoma, yra tokie:

- *Candidatus nitrosopumilus maritimus* (archėja);
- *Halobaculum gomorrense* (archėja);
- *Halobaculum genties* (archėja);
- *Phycisphaera mikurensis* (bakterija);
- *Sulfurimonas denitrificans* (bakterija).

Candidatus nitrosopumilus maritimus [10] yra labai paplitusios archėjos, kurios gyvena jūros vandenyje. Tai pirmasis grupės narys *1a Crenarchaeota*, kuris turėtų būti išskirtas grynakraujėje (angl. pure) kultūroje. Genų sekos rodo, kad grupės *1a Crenarchaeota* galima rasti daugelyje jūrų pakrančių aplink planetą. Jis yra vienas iš mažiausių mikroorganizmų – 0,2 mikronų skersmens. Viena iš jo savybių yra amoniako oksidavimas į nitritus. *Candidatus nitrosopumilus* yra iš *Nitrosopumilaceae* genties.

Halobaculum gomorrense [4] yra archėja, rasta Negyvojoje jūroje. Tai vienintelė *Halobaculum* genties rūšis, kuri yra „strypo/lazdelės“ formos.

Halobaculum genties archėja – iš esamos informacijos pavyko nustatyti tik tą faktą, kad atitinkamas genomas yra iš *Halobalucum* genties.

Phycisphaera mikurensis [3] – rasta ant jūros dumblių. Priklauso *Planctomycetes* tipui (skyriui), *Planctomyces* grupei. *Phycisphaera mikurensis* suteikia įžvalgų į *Planctomyces* gyvavimo ciklą.

Sulfurimonas denitrificans [5] yra *Sulfurimono* genties bakterija. Ši bakterija naudoja, viename iš metabolizės kelių, sulfidą arba tiosulfatą, kaip elektrono donorą, o nitratą arba nitritus, kaip akceptorių. Šis mikroorganizmas buvo nustatytas hidroterminėse ventiliacijos srityse ir naftos telkiniuose, jis šiose aplinkose gali vaidinti svarbų vaidmenį sieros apytakoje.

Buvo nuspręsta analizuoti *Sulfurimonas denitrificans* bakteriją.

1.1.3. *Sulfurimonas denitrificans* charakteristika

Sulfurimonas yra bakterijų gentis iš klasės *Epsilonproteobacteria*. Šios grupės nariai naudoja sulfidą, tiosulfatą ir elementinę sierą, kaip elektronų donorus, o CO_2 kaip jų anglies šaltinį. Žinomi keturi šios genties nariai:

- *Sulfurimonas autotrophica;*
- *Sulfurimonas denitrificans;*
- *Sulfurimonas gotlandica;*
- *Sulfurimonas paralvinellae.*

Sulfurimonas denitrificans pavadinimas buvo pakeistas iš *Thiomicrospira denitrificans*. *Sulfurimonas denitrificans* klasifikacija (1 lentelė):

Mokslinė klasifikacija	
Karalystė:	Bacteria
Skyrius:	Proteobacteria
Klasė:	Epsilonproteobacteria
Būrys(Eilė):	Campylobacterales
Šeima:	Helicobacteraceae
Gentis:	Sulfurimonas

1 lentelė. *Sulfurimonas denitrificans* mokslinė klasifikacija

Sulfurimonas denitrificans iš pradžių buvo rastas ant jūrų pakrančių nuosėdų. Ši bakterija atlieka svarbų vaidmenį ekosistemoje, vykdo sieros junginių transformacijos ir azoto junginių ciklus. *Sulfurimonas denitrificans* transformuoja sierą per sieros oksidacijos procesą ir paverčia nitritus į diazoto dujas per denitrifikaciją. Ši bakterija sukėlė nemažai susidomėjimo mokslo bendruomenėje dėl savo unikalios medžiagų apykaitos, kuri gali oksiduoti sierą ir mažinti nitratų kiekį [7]. Žemiau pateiktame paveikslėlyje pavaizduota *Sulfurimonas denitrificans* bakterija (1 pav.).

1 pav. Sulfurimonas denitrificans bakterija

Sulfurimonas denitrificans [3] genomas turi pakankamai didelius matmenis (2.2 Mbp), tai suteikia didesnį metabolinį universalumą arba reagavimą į aplinką, nei daugumai kitų sekvenuotų klasės *Epsilonproteobacteria* bakterijų. Šios bakterijos genai turi pilną autotrofinį redukcinį Krebso ciklą. Taip pat turi didelį jutimo ir reguliavimo baltymus koduojančių genų arsenalą, kurie svarbūs, kad būtų galima užkirsti kelią oksidaciniam stresui.

1.2 Naudojami įrankiai

1.2.1. *Python*

Python yra sukurta *Guido van Rossumo* 1990 metais. Pirmiausiai ji buvo scenarijų kalba *AmoebaOS* operacinei sistemai. *Python* dažniausiai lyginama su *Tcl*, *Perl*, *Scheme*, *Java* ir *Ruby*. *Python* kuriamas kaip atviro kodo projektas.

Python yra daugiaparadigmė programavimo kalba – ji leidžia naudoti keletą programavimo stilių: objektinį, struktūrinį, funkcinį, aspektinį. *Python* naudoja dinaminį tipų tikrinimą.

Python kūrėjų tikslai buvo sukurti kalbą, kuri yra lengvai skaitoma, išraiškinga, paprasta (tinkama neprofesionaliems programuotojams). Nors pradžioje ji buvo kuriama kaip scenarijų kalba, dabar ji naudojama ir dideliems programiniams projektams. Ši kalba labai paplitusi *Linux* sistemose.

Python – galinga ir patogi programavimo kalba, sparčiai populiarėjanti pasaulyje. Bene akivaizdžiausias šios kalbos privalumas – didelis įtrauktų bibliotekų kiekis. Tai leidžia supaprastinti ir pagreitinti programų kūrimą, kurti universalias, pagal poreikius pritaikytas programas. *Python* labai intuityvi programavimo kalba.

Su kitomis programavimo kalbomis anksčiau susidūręs žmogus, iškart pastebės, kad tą pačią užduotį galima atlikti šimtu skirtingų būdų. Taip yra todėl, kad *Python* savyje talpina tiek kitų populiariausių programavimų kalbų sintaksių ypatybes, tiek tūkstančius įvairiausių vidinių funkcijų.

Python – interpretuojama programavimo kalba, t.y. kodas analizuojamas programos vykdymo metu. Apart didelio skaičiaus standartinių bibliotekų yra įtraukta ir daugybė vidinių duomenų struktūrų. Minėti aspektai programuotojui suteikia didelę programavimo laisvę, leidžia sutrumpinti kodą bei programos rašymo laiką.

1.2.2. *Rast* serveris

Prokariotų genomų sekų skaičius auga nuolat ir auga greičiau nei mokslininkai geba jas visas anotuoti [8]. Tam sukurta visiškai automatizuota paslauga, kuri anotuoja bakterijų ir archėjų genomus. Servisas nustato baltymų koduotę, *rRNR* ir *tRNR* genus, priskiria funkcijas genams, prognozuoja genų pasiskirstymą į genomo posistemius, naudoja šią informaciją atkurti metabolinį tinklą ir sukuria lengvai parsisiunčiamą išvesties failą. Paslauga anotuoja genomą per 12-24 val. nuo pateikimo. Svarbu paminėti, jog greitis *MyRast* nėra svarbiausias aspektas, svarbiausia yra tikslumas, išsamumas ir suderinamumas.

Rast serveris naudoja *FIGfams* baltymų šeimų rinkinį. *FIGfams* – baltymų rinkinys, kurio nariai yra „globaliai panašūs“ ir visi turi bendrą funkciją. *Gerlt* ir *Babbitt* (2001) pasiūlė „turinčius tą pačią funkciją homologus“ genams, kurie yra nuo pradžios iki pabaigos panašūs ir turi tą pačią funkciją. Baltymų šeimos nariai yra nuo pradžios iki pabaigos panašūs:

- baltymų sekos yra panašios, jeigu sutampa ne mažiau kaip 70%;
- panašumas yra įvertinamas naudojant *BLAST*.

FIGfams yra bandoma sudaryti baltymų rinkinius, kurie atlieka tokią pačią funkciją. Patikimiausi *FIGfams* yra rankiniu būdu sudaryti posistemiai. Yra du būdai naudoti baltymus tame pačiame *FIGfam*:

- sulyginti du labai panašius genomus ir nustatyti ryšį tarp genų jų regionuose;

- jei artumas chromosomų buvo išsaugotas tikrinant daugelyje genomų, tikima, kad baltymai turi tą pačią funkciją.

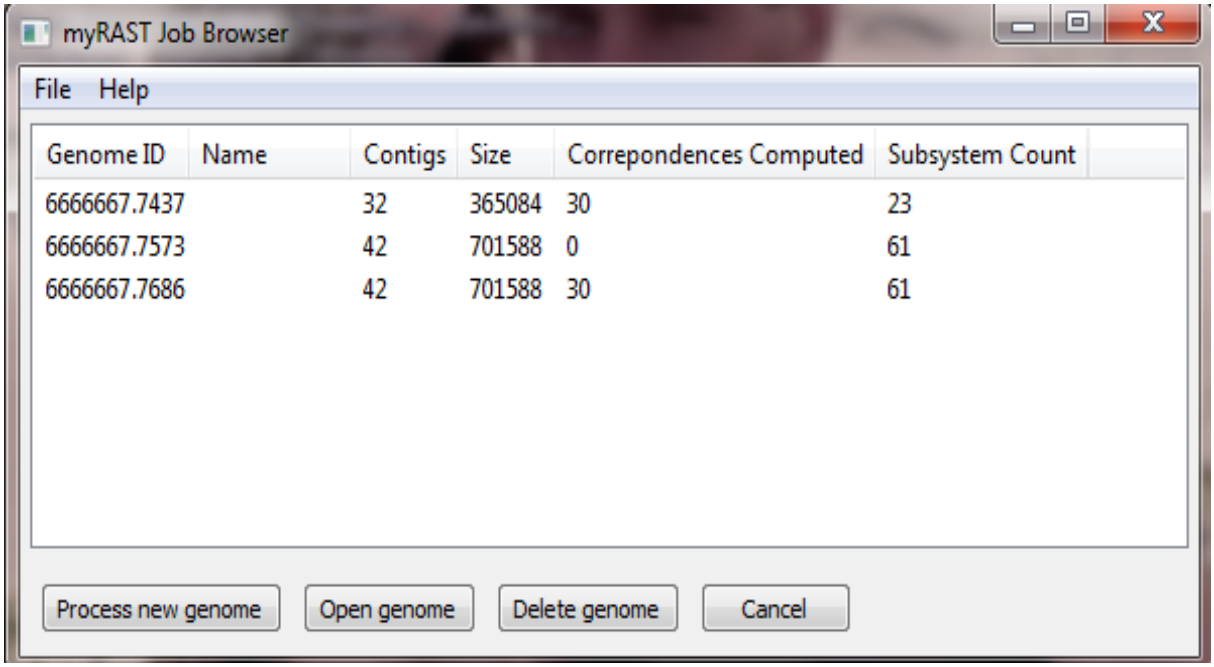
FIGfams yra naudojamas *RAST* funkcinei anotacijai. Jie naudojami, nes norima išvengti arba sumažinti galimai klaidingą funkcijų anotaciją. *FIGfams* nėra kuriami rankiniu būdu, jie gaunami automatiškai. Jei yra aptinkama klaidų *FIGfam* baltymų rinkinyje, korekcija daroma taisant posistemį arba kuriant jį naują. Šiuo metu yra daug *FIGfam* kolekcijų. Didžiausia apima apie 130,000 baltymų rinkinių iš kurių apie 50% yra rinkiniai, kuriuos sudaro tik dvi sekos.

1.2.2.1 MyRast įrankis

Galima atsisiųsti *MyRast* [11] programą, kuri naudoja *Rast* serverį.

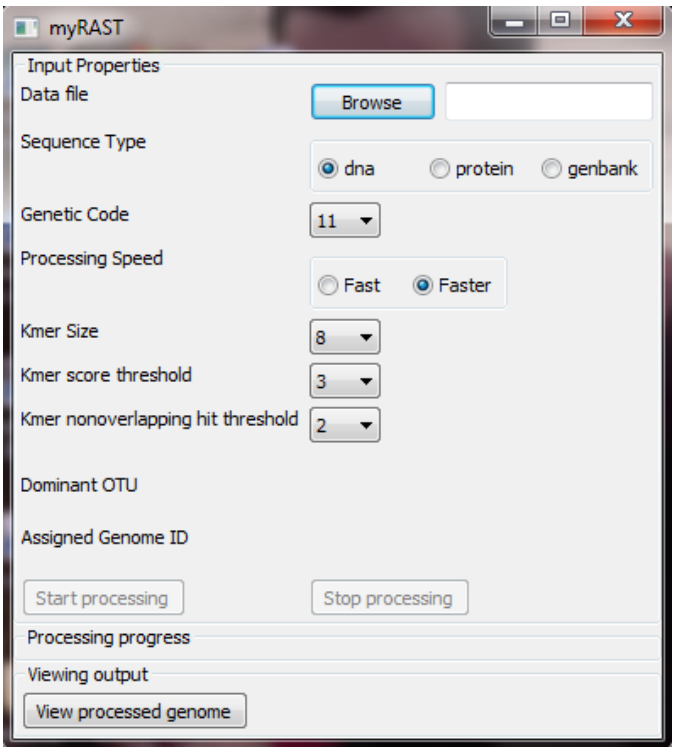
Yra keletas žingsnių darbui su *MyRast*:

1. Kai paleidžiamas *MyRast* matomas tuščias langas, nebent yra jau ankščiau anotuotų genomų, tuomet matomas jų sąrašas (3 pav.).



3 pav. Pradinis *MyRast* langas

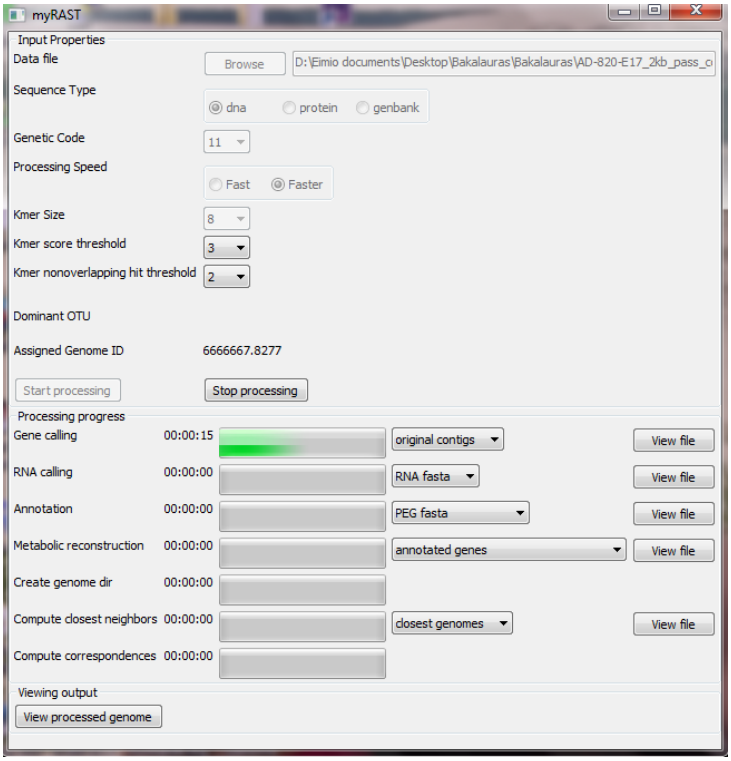
2. Pasirinkus „Process new genome“ pasiūloma pasirinkti failą, kuris turi būti anotuojamas (4 pav.).



4 pav. Failo ir nustatymų pasirinkimas

Galima įkelti failą *Genbank* formatu, nuoseklaus regiono (angl. contigs) failą *FASTA* formatu arba baltymų sekų failą *FASTA* formatu.

3. Kai viskas pasirinkta spaudžiama „Start processing“ (5 pav.).



5 pav. Vykdoma genomo anotacija

Kai pradedamas apdorojimas, matomas valdymo skydelis, kuriame rodomi anotacijos žingsniai.

4. Norint peržiūrėti genomą spaudžiama „View processed genome“ (6 pav.).

6 pav. Anotuoto genomo vizualus pavaizdavimas

Šis vaizdas rodo regioną naujai sekvenuoto genomo. Genai, kurie turi tą pačią funkciją, yra nudažyti ta pačia spalva. Genai yra vaizduojami kaip rodyklės. Užvedus ant bet kurios iš jų yra matoma informacija: ID, funkcija, nuoseklus regionas (angl. contigs), kur prasideda ir baigiasi, taip pat ilgis. Spaudžiant rodykles galima pereiti į vieną ar kitą pusę, per vieną geną, pusę ekrano, visą ekraną arba pereiti iš vieno nuoseklaus regiono į kitą. Yra trys pagrindinės funkcijos: keisti geno anotaciją, ištrinti arba įterpti naują geną.

1.2.3. *GeneMark*

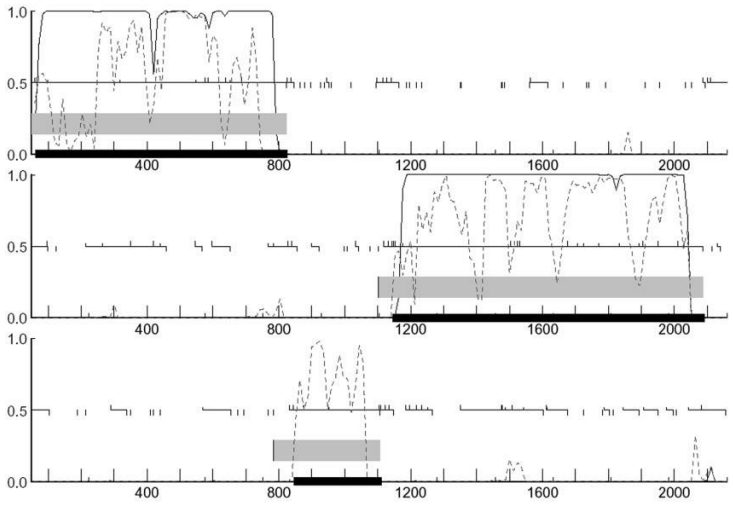
Genų identifikavimo uždavinys dažnai sprendžiamas tyrėjų, kurie susiduria su naujais ir gerai išnagrinėtais genomais, jiems šias užduotis patogiai ir patikimai padeda išspręsti *GeneMark* [6] – interneto programinė įranga (*http://opal.biology.gatech.edu/GeneMark/*). Svetainė suteikia sąsajas su *GeneMark* šeimos programomis, kurios skirtos genų prognozavimui: *prokariotų*, *eukariotų* ir virusų genomų. Šiuo metu serveris leidžia analizuoti apie 200 prokariotų ir daugiau kaip 10 eukariotų genomų, naudojant kiekvienai rūšiai būdingas programinės įrangos versijas. *GeneMark* svetainė dažnai atnaujinama, pateikiamos naujausios programinės įrangos versijos ir genų modeliai.

GeneMark ir *GeneMark.hmm* gali būti naudojami per *GeneMark* svetainę skirtą priokariotų *DNR* analizei. *DNR* analizė, bet kokių *prokariotų* rūšių yra palaikoma specialios versijos – *GeneMark.hmm*, naudojant euristinį modelį apskaičiuotą iš nukleotidų dažnio ir sekų

įvesties, kurių mažiausias ilgis gali būti 400 nt. *GeneMarkS* gali naudoti ilgesnes 1Mb sekas. Kaip dauguma įrankių, *GeneMark* svetainė naudoja panašių sąsajų programas. *GeneMark.hmm* leidžia įkelti failą su DNR sekomis arba įklijuoti jas į laukelį. Jei *FASTA* failo aprašymas prasideda simboliu „>“ tai ši eilutė naudojama, kaip pavadinimas. Visos kitos raidės, išskyrus A, C , G ir T, yra ignoruojamos ir konvertuojamos į N raides. Sąsaja reikalauja pasirinkti rūšies pavadinimą. Pasirinkimas *RBS* modelio yra neprivalomas. *GeneMark.hmm* praneša visus prognozuojamus genus formate, kuriame nurodoma geno kryptis, jo ribos, nukleotidų ilgis ir genų klasės. Klasė nurodo, kurį iš dviejų *Markovo* modelių naudoja *GeneMark.hmm*, tipinį ar netipinį. Galimybė, leidžianti generuoti *GeneMark* prognozes kartu su *GeneMark.hmm* analize suteikia svarbios papildomos informacijos. Šiuo atveju *GeneMark* yra nustatytas naudoti tuos pačius mokymo duomenis kaip ir *GeneMark.hmm*. Verta paminėti, kad *GeneMark.hmm* ir *GeneMark* papildo vienas kitą. Grafinės analizės išvestis yra prieinama *PDF* arba *PostScript* formatu. Šios išvesties fragmentas, iliustruoja tiek GeneMark.hmm tiek GeneMark prognozes (7 pav.).

Grafinis išvedimas aiškiai vaizduojamas naudojant keletą *Markovo* grandinės modelių, atstovaujančių skirtingų klasių genus. Kodavimo potencialo grafikas gaunamas naudojant tipinių genų modelius, parengtus pagal *GeneMarkS* – žymima juoda linija, o kodavimo potencialas gaunamas naudojant netipinį geno modelį – žymimas punktyrine linija.

Analizei DNR sekų, kurios nėra iš anksto apskaičiuotos specifiniam modeliui, galima naudoti programos versiją, kuri skirta sekoms didesnėms už 400 nt. Įrodyta, kad šis metodas naudingas nehomogeninių genomų analizei. Jei modeliai turi būti apskaičiuoti nežinomoms sekoms, kurios yra 1Mb ar ilgesnio ilgio, gali būti naudojama *GeneMarkS* programa. Ši programa turi daugiau skaičiavimo išteklių, taigi jos išvestis yra teikiama elektroniniu paštu.



7 pav. Diagrama vaizduojanti *GeneMark.hmm* ir *GeneMark* prognozes

1.2.4. *Prokka*

Genomo anotacija yra skirta identifikuoti ir ženklinti visus svarbius genomo sekos procesus. Čia pristatomas *Prokka* [9] – komandinės eilutės įrankis, kuris gali būti įdiegtas, bet kurioje *Unix* sistemoje. *Prokka* yra programinės įrangos rinkinys, kuris skirtas paprastai ir patikimai anotacijai. *Prokka* koordinuoja esamus programinės įrangos įrankius, kad pasiekti išsamų ir patikimą bakterijų genomų sekų anotaciją. Jei įmanoma, *Prokka* panaudos visus procesoriaus branduolius. Tipiškas bakterijos genomas, naudojant šiandieninį kompiuterį, būtų anotuotas maždaug per 10 minučių. Tai puikiai tinka kartotinei sekų modelių analizei.

Prokka vykdymo procesas:

1. **Įvedimas.** *Prokka* tikisi gauti genomo *DNR* sekas *FASTA* formatu. Sekos be tarpų būtų ideali įvestis. Šis sekos failas yra vienintelis privalomas parametras programinei įrangai.
2. **Anotacija.** *Prokka* remiasi išoriniais prognozavimo įrankiais, kad identifikuoti genomo funkcijų koordinates per nuoseklų regioną (angl. contigs). Šios priemonės išvardytos 2 lentelėje ir visos jos išskyrus *Prodigal* teikia koordinates ir aprašo atitinkamas funkcijas. Baltymus koduojantys genai yra anotuojami dviem etapais. *Prodigal* identifikuoja galimų genų koordinates, bet neaprašo tariamų genų. Tradicinis būdas prognozuoti, ką genas koduoja – jį palyginti su didelės duomenų bazės žinomomis sekomis. *Prokka* naudoja šį metodą, tik hierarchine tvarka, pradedant nuo mažų patikimų duomenų bazių, po to pereinant prie vidutinių ir galiausiai prie didelių duomenų bazių.

Įrankis	**Funkcijos**
Prodigal (Hyatt 2010)	koduojamos sekos(CDS)
RNAmmer (Lagesen 2007)	ribosomų RNR genai (rRNA)
Aragorn (Laslett 2004)	perduoda RNA ir tmRNA
SignalP (Petersen 2011)	signalo peptidai (N-term iš CDS)
Infernal (Kolbe 2011)	nekoduojamos RNR

2 lentelė. *Prokka* naudojami įrankiai

3. **Išvedimas.** Prokka sukuria dešimties formatų failus į nurodytą išvesties aplanką (3 lentelė).

Galūnė	Failo turinio aprašymas
.fna	*FASTA* failas su originaliais įvesties nuosekliais regionais (nukleotidai)
.faa	*FASTA* failas su išverstais kodavimo genais (baltymai)
.ffn	*FASTA* failas visų genomo funkcijų (nukleotidai)
.fsa	Nuoseklių regionų sekos pateikimui (nukleotidai)
.tbl	Funkcijų lentelė pateikimui
.sqn	Redaguotas failas pateikimui
.gbk	*Genbank* failas su sekomis ir komentarais
.gff	*GFF* v3 failas su sekomis ir komentarais
.log	*Prokka* prisijungimo failas
.txt	Santraukos anotacijos statistika

3 lentelė. *Prokka* išvedimo failų aprašymas

Prokka buvo sukurtas siekiant, kad tai būtų tikslus ir greitas įrankis.

2. Praktinė dalis

Praktinėje dalyje sprendžiamas pagrindinis uždavinys sukurti genomo naršyklę įprastinėmis priemonėmis naudojant skripto kalbas ir *html*.

Šio uždavinio sprendimas susideda iš kelių dalių:

- *Sulfurimonas denitrificans* bakterijos sekvenuoto genomo analizavimas bioinformatikos įrankiais:
    - *MyRast;*
    - *Prokka;*
    - *GeneMark.*
- įrankių išvesties sekų patikrinimas *Blast (Basic Local Alignment Search Tool)*;
- bioinformatikos įrankiais anotuotų genomų kryžminis palyginimas tarpusavyje;
- sukurtas skriptas fermentams rasti;
- sukurtas *Python* skriptas skirtas generuoti genomo naršyklę, t.y. *html* failą.

Toliau aprašysiu šiuos praktinėje dalyje atliktus uždavinius, kokie buvo gauti rezultatai ir kas buvo sukurta.

2.1. *MyRast* pritaikymas

Šio įrankio pagalba buvo anotuotas *Sulfurimonas denitrificans* (toliau: *Sulfurimonas*) genomas. Gavome anotuotą *FASTA* failą su analizei reikalinga informacija.

Turimas sekvenuotas bakterijos *Sulfurimonas* genomas. Šis genomas pateiktas FASTA formatu, kuriame genomo *DNR*. Ši sekvenuota *DNR* susideda iš daugybės *A, C, G, T* raidžių. Visas genomas yra suskirstytas į 42 dalis, kurių kiekviena dalis turi pavadinimą, ilgį ir identifikacinį numerį (8 pav.).

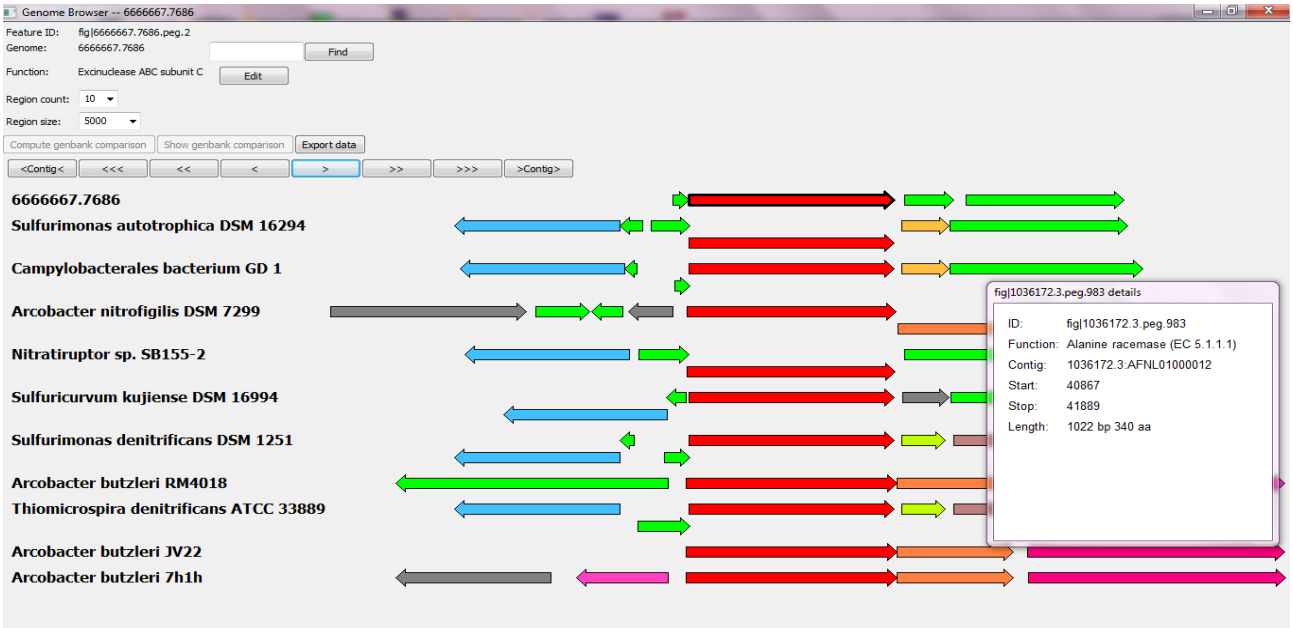
```
1  >AD-820-E17_NODE_1_length_88502_cov_101.2_ID_1
2  TTTTTGTTCTCAGTCATCCAGGAGGCAAACAAGAGTTTGATAATAATATAGAAAGTATTA
3  AAAGTGCTTTAAAATTCTACAAGCCGGAAGAATGTAGCGAAACTGTAATAAATGACATCA
4  AAGCATTTGTAACTCATACGCCCATAAAACCTGCGATAACAGCACAAAAAAAAGAACTTC
5  GATACATGGAAAGAGCTACAGGAGATTTCCCAATAAATATTCATGATGAAAAGCTAAATT
6  CTCTTGTAAAGAGCATCCAAAAAATCATAAAAGATGTAAATGAGCAAAAGTCAGCTACTT
7  TTACCCAAATAAAAAAGCAATAATGACACTAGAACAAACAATAAAACAACTGCCGCAAGG
8  TGCAGGAATATATCAATACTTTGATATTAATGGGCATCTTCTTTACATTGGGAAAGCAAA
9  AAATCTCTCAAACAGAGTTAAAAGTTATTTTAATTTTACACCTGAATTAAAACCCAATTC
10 TAATCTTTCAAACAGAATCACAAAAATGATTTTGCAAACTGCTTCACTCAGTTATATAGT
11 AGTAAATTCTGAACATGACGCCCTCATCTTAGAAAATTCTCTTATCAAACAGCTGGCTCC
```

8 pav. Fragmentas iš *Sulfurimono* sekvenuoto genomo

Dirbant su *MyRast* buvo parinkti tokie parametrai:

- sekos tipas – dnr;
- anotavimo greitis – greitas;
- genetinis kodas – 11;
- Kmer dydis – 8;
- Kmer rezultato riba (angl. score threshold) – 3;
- nesutampančių hitų riba – 2.

Paskutinis žingsnis yra įkelti savo failą su *DNR* sekomis ir spausti „*Start*“ mygtuką, kuris pradeda vykdyti genomo anotaciją. *Sulfurimono* genomo anotacija naudojant *MyRast* truko apie 1 valandą, nors genomas ir nėra didelis apie 710kb *FASTA* failas. Tai yra pakankamai greitai, nes atliekama ne tik genomo anotacija, bet ir pavaizduojama vizualiai. Vykdomą arba jau atliktą anotaciją galima peržiūrėti paspaudus „Open genome“ iš karto yra atidaromas vizualus pavaizdavimas anotuoto *Sulfurimono* genomo (9 pav.).



9 pav. *MyRast* vizualus genomo pavaizdavimas

Vizualiame genomo anotacijos pavaizdavime galima atlikti įvairias funkcijas, nagrinėti ir išsiaiškinti kiekvieną geną individualiai. Šiame atvaizdavime matome daug įvairių spalvų rodyklių, kiekviena iš jų vaizduoja skirtingą geną. Kryptis yra į kairę jei baltymo dydis eina nuo didesnės į mažesnę pusę pavyzdžiui, jei genas prasideda 1863 baze ir baigiasi 1342 baze ir atvirkščiai jei į dešinę pusę. Rodyklių spalvos skiriasi, kiekviena spalva žymi genų funkcines grupes: šviesiai žalia spalva žymi, jog tai yra fermentų grupė, mėlyna spalva žymimas nežinomas (angl. *hypothetical*) baltymas ir t.t. Taigi, jei domina konkreti geno funkcinė grupė, jas galima surasti pagal spalvas. Užvedus pelę ant kiekvienos rodyklės yra rodomas atskiras langas su to

geno informacija: ID, funkcija, nuoseklus regionas (angl. contig), pradžia, pabaiga ir baltymo ilgis. Pažymėjus, bet kurį geną ir paspaudus dešinį pelės mygtuką, galima sužinoti būtent to geno *DNR* ar baltymo seką, taip pat yra parinktis, kuri leidžia patikrinti konkretų geną *NCBI Blast*, jei manoma, jog tai gali būti klaidingai anotuotas genas. Taip pat yra galimybė ištrinti geną. Pasirinkus, bet kurį geną, *MyRast* lango viršuje atsiranda informacija apie šį geną, galima jį pataisyti, pavyzdžiui jeigu įvesta neteisinga geno funkcija, spaudžiame *„edit“* mygtuką ir pataisome funkciją į tinkamą. Taip pat yra paieška genome, jei mums reikalingas atitinkamas genas. Pavyzdžiui, įvedus 600 parodoma genomo dalis, kurioje yra 600-asis genas. Tai labai patogu ir naudinga analizuojant genomą, jei žinoma, koks konkretus genas yra reikalingas. Galima pasirinkti ir atvaizduojamą regionų skaičių bei jų dydį – patogu, jei vienu metu reikia dirbti su didesniu kiekiu genų.

Svarbiausia *MyRast* funkcija šiai analizei – duomenų eksportavimas. *MyRast* leidžia eksportuoti duomenis trimis formatais:

- skirtukais atskirtas tekstas;
- tik kableliais atskirtas tekstas;
- *FASTA* formatas.

Taip pat galima pasirinkti funkcijų tipus. Galimi variantai yra *peg* ir *rna*. *Peg* yra baltymų numeravimas, o *rna* tai *RNR* funkcijų rodymas. Tiriamu atveju abu funkcijų tipai mums reikalingi, todėl pažymimi abu. Kita galimybė yra pasirinkimas, kokios sekos turi būti įkeliamos į eksportuojamą failą. Gali būti įtrauktos tik ištransliuotos baltymų sekos arba ir DNR sekos. Kaip atrodo baltymų ir *DNR* sekų *FASTA* formato išvesties failas pavaizduojama 10 paveikslėlyje.

Kaip matyti paveikslėlyje, pirmiausiai yra išvedama *DNR* seka, o tik po to jau ištransliuota baltymo seka. Abiejų sekų pavadinimai tokie patys, tai parodo, jog tai ta pati seka. Kaip matome *DNR* seka yra akivaizdžiai ilgesnė nei jau identifikuota baltymo seka (10 pav.).

```
>fig|6666667.7686.peg.28 AD-820-E17_NODE_1_length_88502_cov_101.2_ID_1_26436_26786 C-type cytochrome, putative
ttgaggtatctttatccggttttatttttcataacttcattgcatgctgtagacatgaga
cctttactctttaatggaaactgcgtcacatgtcaccatacaagcagaacaatatctgca
ccctctattgtagagattaaagaaaattatttaagagcattccctcaaaaagaggctttt
gttgcgtacatgtcgacatgggttgctaaaccaaatatagagacttccataatgcatgat
gcaatagacaagtatgaagtgatgcctgacttaggatttgacataagtacttcaagagat
atctctgcttacatttacgaaacagattttagtcagattgaaccatactaa
>fig|6666667.7686.peg.28 AD-820-E17_NODE_1_length_88502_cov_101.2_ID_1_26436_26786 C-type cytochrome, putative
MRYLYPVLFFITSLHAVDMRPLLFNGNCVTCHHTSRTISAPSIVEIKENYLRAFPQKEAF
VAYMSTWVAKPNIETSIMHDAIDKYEVMPDLGFDISTSRDISAYIYETDFSQIEPY
>fig|6666667.7686.peg.29 AD-820-E17_NODE_1_length_88502_cov_101.2_ID_1_27225_26764 Phosphoribosyltransferase
atgtttatgaaatattatgcatatgaagattttaaacaagacacaaacaaactcttgaaa
cagataaaagagtttcagccggagatgatagtgggcattgctagaggtgggcttactttg
tctcatgcattggctgagggattaaacataagagatgttcaaacgctaagaactgaactt
tatgatgacacgcacaaaagggatgagataagcatacataacagatgtctattcggtgat
attaagagagttttggttgtggatgacatagctgacagcggtgatacgctcaaagctgtt
```

10 pav. *MyRast* išvesties failas, kuriame įtrauktos ir *DNR* sekos

Tolimesnei analizei reikalingos tik baltymų sekos. Todėl nereikia įtraukti *DNR* sekų į eksportuojamą failą. Tokiu būdu gaunamas *FASTA* formato failas tik su baltymų sekomis (11 pav.).

```
>fig|6666667.7686.peg.704 AD-820-E17_NODE_40_length_3792_cov_6.578_ID_79_2157_1591 hypothetical protein
MEDTFSNLATYGYIGLFLYSLGGGFVALIGAGVLSFLGKMDLTYSIAIAFFANALGDVLL
FYMARYHKSTMMDGIRKHRRKLALSHIMMKKYGSWIILIQKYIYGIKTLIPIAIGLTKYD
FKKFAILNVFSAGVWALTFGLGSYYSGNALVKFAEIIGDKPWIAPLVLVVLGGTLWFYLT
HATKKRTK
>fig|6666667.7686.peg.705 AD-820-E17_NODE_40_length_3792_cov_6.578_ID_79_3778_2165 Excinuclease ABC subunit B
MRLSSTANLLSYDDVIVIASVSANYGLGDPQEYENMVQSVAVGDVIAQKKLLLRLVEMGY
SRNDTYFDSGHIRVNGETLDIYPPYFEQEAIRIEFFGDEIEAIYTFDIIDNKRLEDHKQF
TIYATSQFSVSQEKMAVAIKRIEEELDERLAFFQAEGKLLEHQRLKQRVEFDLEMLQTTG
MCKGVENYSRLLTNKKPGEAPYTLLDYFELHHKDYLVIVDESHVSLPQYRDMYAGDRARK
EVLVEYGFRLPSALDNRPLKADEYINKAPHYLFVSATPSVYELEMSSVTAKQIIRPTGLL
DPIIEIKSSDNQVEDIHDEIKKITVNDERVLITVLTKKMAEALTKYLADLGIKVQYMHSD
IDTIERNQIIRALRLGEFDVLIGINLLREGLDLPEVSLVAILDADKEGFLRSETALVQTI
GRGARNSKGRVILYANKITGSMQRAIEKTTARREIQEAHNKKHNITPTTTKRSLDENLKL
EDYGDLYQKHKKMDKIPASERKAITKELMLRMKQAAKELNFEEAARLRDEIMKIKQL
```

11 pav. *FASTA* formato failas tik su baltymų sekomis

Atlikus bakterijos genomo *Sulfurimono denitrificans* anotaciją su *MyRast* buvo surasta 713 baltymų sekų. Anotuotame genome kiekvienai sekai yra priskiriama viena eilutė su jos anotacija, ji išskiriama pradžioje įterpiant varnelės ženklą.

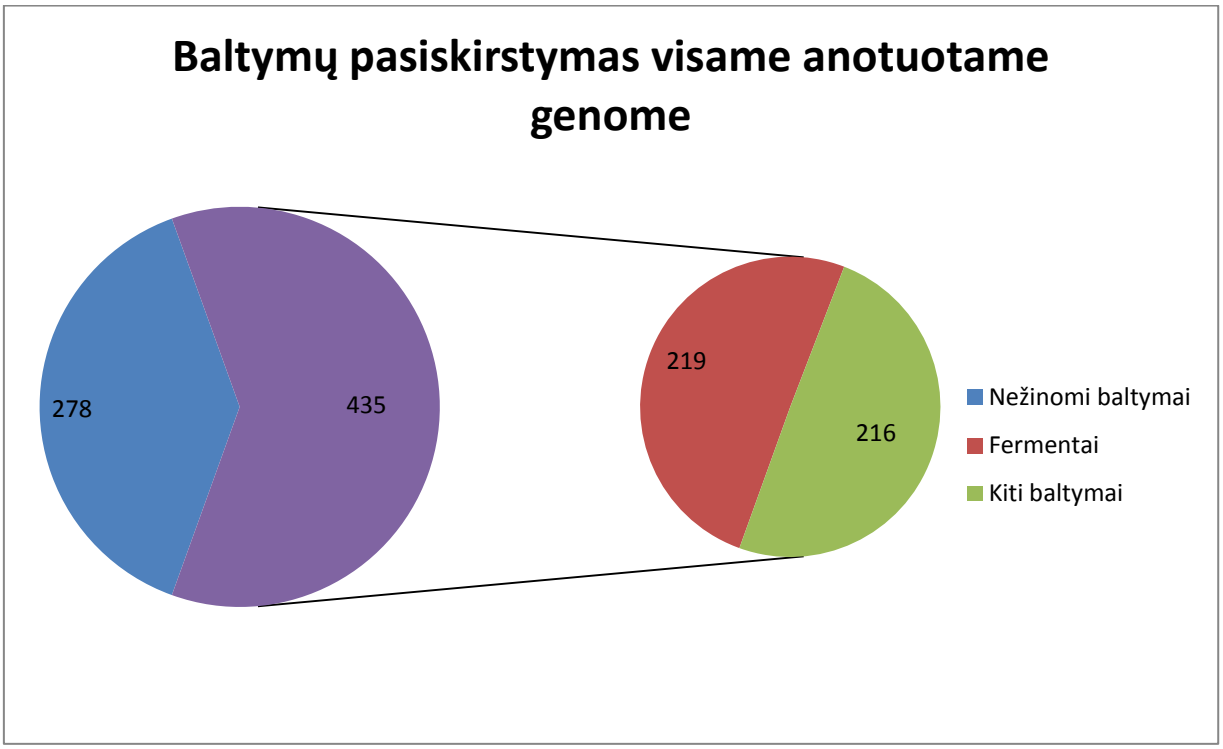
Toliau bus nagrinėjama, kas anotacijos eilutėje yra nurodoma. Pavyzdžiui, paimsime anotacijos eilutę iš 11 paveikslėlio:

- fig|66666667.7686 – genomo ID, kuris yra priskirtas *MyRast* programos;
- peg.705 – sunumeruojamos sekos pagal eilę;
- AD-820-E17 – sekvenavimo metu genomui suteiktas žymėjimas;
- NODE_40 – žymi iš kurios sekvenuoto genomo dalies buvo ištransliuota baltymo seka;
- length_3792 – dar netransliuoto genomo dalies ilgis;
- ID_79 – dar netransliuoto genomo ID;
- 3778_2165 – baltymo sekos pradžia ir pabaiga, taip pat galime apskaičiuoti transliuoto baltymo ilgį;
- Excinuclease ABC subunit B – baltymo sekos pavadinimas.

Dalis, nuo sekvenavimo metu genomui suteikto žymėjimo iki ID, yra paimama iš failo, kuris anotuojamas, tai nėra *MyRast* atrasta informacija. Kaip matome 11 paveikslėlyje 704 sekoje kartais *MyRast* nepavyksta surasti baltymo sekos ir ji yra pažymima, kaip „*hypothetical protein*“. Visus nesurastus baltymus, t.y. tuos, kurie pažymimi „*hypothetical protein“* skripto

pagalba, buvo iškirpti iš pagrindinio *FASTA* failo ir ieškomi *NCBI Blast protein* pasinaudojant *Blast* skriptu. Šis skriptas leidžia vienu metu nurodyti daug ieškomų sekų ir jas parsiunčia *html* formatu. Tos baltymų sekos, kurios *MyRast* pažymėtos kaip nežinomos yra palyginamos su rezultatais, gautais iš *Blast*. Apie 90% pirmuoju numeriu pažymėtų sekų yra „*hipothetical protein*“ ir jau antru numeriu einančios sekos dažniausiai įvardija koks tai galėtų būti konkretus baltymas. Galima daryti išvadą, jog *MyRast* įrankis renkasi labiausiai tikėtiną iš gaunamų baltymų, nors dažniausiai jau antru numeriu yra pateikiamas galimas konkretus atsakymas.

Apačioje pateiktame paveikslėlyje pavaizduota, kaip *MyRast* anotuotose 713 sekose pasiskirstė nežinomi baltymai, įvardinti baltymai ir fermentai (12 pav.).



12 pav. Baltymų pasiskirstymas anotuotame genome

Kaip matome, truputį daugiau nei trečdalis baltymų buvo neįvardinti. Nors tai atrodo didelis skaičius, tačiau reikėtų atsižvelgti ir į tai, jog *Sulfurimonas* yra labai mažai ištirta bakterija.

2.2. Prokka pritaikymas

Prokka – programinės įrangos įrankis, skirtas greitam anotavimui bakterijų, archėjų ir virusų genomų. Jis sukuria standartus atitinkančius išvesties failus. Šis įrankis yra valdomas naudojant komandinę eilutę. Pažymima, kad šis įrankis yra pakankamai sunkiai instaliuojamas, reikia nemažai priedų, kad būtų galima juo naudotis.

Naudotis šiuo įrankiu nėra taip paprasta kaip *MyRast*, tačiau išanalizavus vartotojo vadovą ir pasidomėjus apie šį įrankį, bei pabandžius atlikti kelias anotacijas viskas tampa suprantama. Šis įrankis negali sukurti vizualaus genomo pavaizdavimo, jis teikia tik įvairius išvesties failus, o jų yra nemažai. *Prokka* generuoja 10 išvesties failų. Įrankis turi daug komandinės eilutės opcijų.

Šiuo bioinformatikos įrankiu buvo anotuota *Sulfurimonas denitrificans* bakterija. Pirmiausiai komandinėje eilutėje reikėjo parašyti komandą, kuri įvykdytų bakterijos anotaciją:

```
prokka --outdir Prokka --prefix Sulfurimonas AD-820-E17.fa
```

Šia komanda nurodau direktoriją ***Prokka***, kurioje norima, kad būtų sukurti visi 10 išvesties failų su jų priešdėliais pavadinimu ***Sulfurimonas.*** Toliau nurodomas *FASTA* formato failas, kuris turi būti anotuotas, **AD-820-E17.fa.** Šis įrankis genomo anotaciją įvykdė greitai, maždaug per 10 minučių (13 pav.).

```
>PROKKA_00001 UvrABC system protein C
MTLEQTIKQLPQGAGIYQYFDINGHLLYIGKAKNLSNRVKSYFNFTPELKPNSNLSNRIT
KMILQTASLSYIVVNSEHDALILENSLIKQLAPKYNILLRDDKTYPYIYIDNSSEYPRFD
ITRKIIKSADITYFGPYSVGARDILDSIYEVCKLVQKKACLKSKKACLYYQIDKCLAPCE
FKVSHVRYKAELDLAQELIQNKKTLISKLTEKMSFYAEEMRFEEAGELRDRIERISRSEI
KSEIDFASNENYDIFVLHNSETRAVAVRIFMRNGKIISSSHDFIQLNDGYDEDEFYQRVL
LDFYAKEKPPIIAPILVMKKFSGLEIIAEHLSILFEKKALITAPSRGDKKHLIDLAVLNA
KELLKADKKQDLTKLFTEIKELLSLERIPNRVEIFDNSHMAGVATVGAMIVYENSQFDKK
SYRTYHLDAKDEYAQMRETLTRRVESFSKNSPPDLWIIDGGTTLLRLAVDILDSNGIFID
VIAISKEKIDAKSHRAKGKAHDILHTKDESFRLNPNDKRLQWIQNLRDEAHRSAIAFHKK
TKLKLDKASKLLNLHGISEAKIKKLLNHFGTFEALKELSEEEIASVLNIKDAQAIKNIYK
>PROKKA_00002 Aerotaxis receptor
MDKVIPVDEEYIYKGRVIISQTDVKGIITFANRKFYEVSGYALDELIGSSHSIMRHPNMP
KAVFEKIWETISGGQIWTGIIKNLRKDGRYYWVDIEILPIHNENNELTGYISARKPASRK
NVNETMALYKKMLATEQGDKNVNL
```

13 pav. *FASTA* formato failas generuotas su *Prokka*

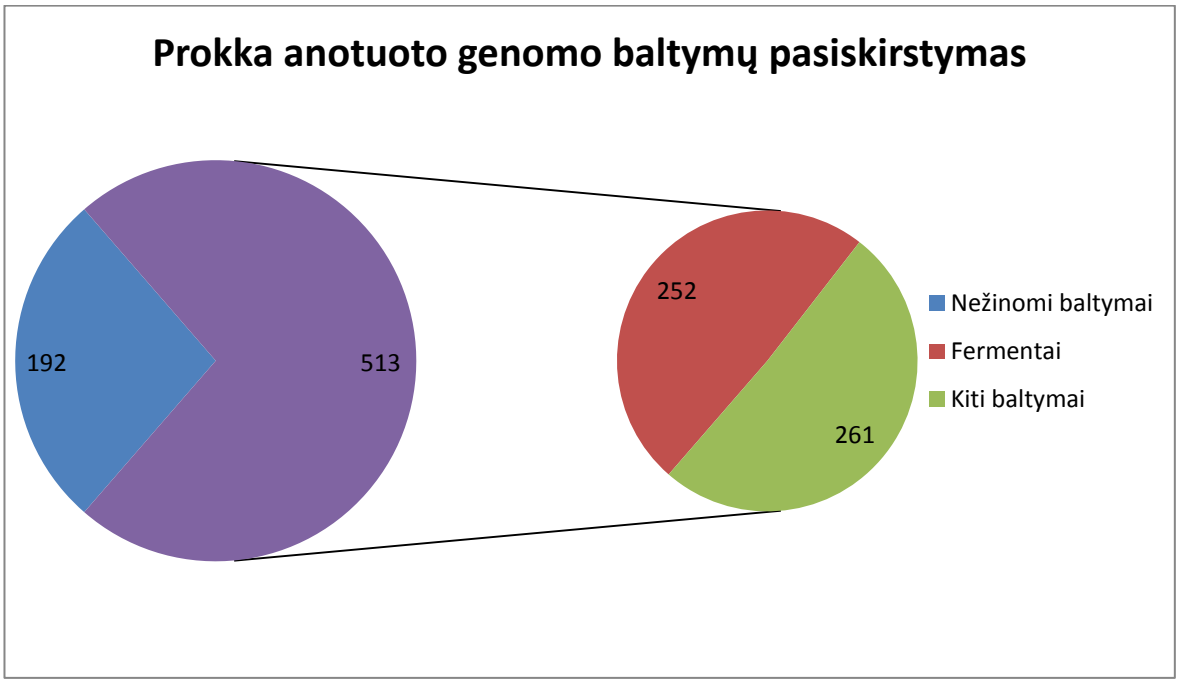
FASTA faile yra nurodomas tik genų eiliškumas ir surastas baltymas. Faile trūksta informacijos apie baltymo pradžią ir pabaigą genome, todėl nežinomas kiekvieno geno ilgis. Taip pat nėra nurodoma fermentų pavadinimų. Negalima teigti, jog *Prokka* neanotavo reikalingų duomenų, jie buvo anotuoti tik kituose formatuose. Visa reikalinga informacija yra *.gbk* faile t.y. standartinis *Genbank* failas, kur kiekvienai sekai aprašoma jos informacija (14 pav.).

Šio formato faile yra visa reikalinga informacija: DNR sekos pradžia ir pabaiga, jos ilgis, vieta genome, surasti fermentai, baltymo pavadinimas ir ištransliuotas pats baltymas.

```
CDS             4158..6530
                /gene="barA_1"
                /locus_tag="PROKKA_00004"
                /EC_number="2.7.13.3"
                /inference="ab initio prediction:Prodigal:2.6"
                /inference="similar to AA sequence:UniProtKB:P0AEC5"
                /codon_start=1
                /transl_table=11
                /product="Signal transduction histidine-protein kinase
                BarA"
                /translation="MQIGLKNRLRLISLFPILILFSLTSYFIYDFYMNVQENHIVFLT
                LTFVIWVISIILAILGYLLSNEISTNITKLEDVLTRVAEDNKDFSSGKEMDLHTPEGT
                ADAYNLLEKIIKKTREDKAFAQEASEAKSMFLANMSHEIRTPLNGIVGFTELLKDTGL
                GEEQLEFVEIIEKSSENLLEIINNILDLSKIESNKIEVEDVIFDPITEFESAVDVYAV
                RASEKHIDLGCFIDPALERPLKGDPTKIKEVIINLLSNAVKFTSNAGALNVEIRKIDS
                GIEGTTRISFEIQDSGIGITSEQKSRIFDAFSQADISITRKYGGTGLGLTISSRFVEL
                MGGKLNLRSKLGKGTTFYFTLDFEEAEEAVNSSKNIFSKLNALILESPYKTKKQESYL
                REYLDYFGVNYKMFKNLDEIEISQKYSDYDLVITDYEFISEELLHKYEEFPKPVILLA
                KSSFMKKIDSMTLNIYKTLYEPLNISKLQQILSNYYTEKLTMTKIKKVVKKKVEEKDL
                KFNANILVAEDNVINQKLIKRTLEDIGLSVSVASNGLEAFQKRKDENFDLIFMDIQMP
                YLDGIEATQEILDYEKDYNKPHVPIIALTANALKGDRAKFLEAGLDEYTTKPLIRSEI
                VSMLNNFLSDFVVSGSAAISDTKSKPIVPIKKEVASLPESKNDGKNYKADILLAKQSA
                FESKLYAKILTELEYSYEITSNIEELKNLTKEFTYKLIVLDDEFNGLELSQFSKDIKE
                SNVATGFKTHLILINSSQKEKILEYKPYVDEIIENVVNKDILQLVFKKFI"
```

14 pav. Ištrauka iš anotuoto *Genbank* formato failo

Naudojant *Prokka* įrankį buvo rastos 705 sekos (8 mažiau nei su *MyRast* įrankiu). Žemiau esančiame paveikslėlyje pavaizduota, kaip šiam įrankiui pavyko anotuoti baltymus (15 pav.).



15 pav. *Prokka* įrankiu anotuoto genomo baltymų pasiskirstymas

Kaip matoma paveiksle *Prokka* įrankis identifikavo daugiau baltymų nei *MyRast*. Mažiau nei trečdalis, liko neidentifikuoti. Visus nežinomus baltymus patikrinus *Blast*, buvo apie 90% baltymų sekų, kuriose pirmoje vietoje buvo rodomas *„hypothetical protein“*.

Prokka įrankis yra patogus dėl greito anotavimo ir didelio pasirinkimo išvesties failų, tačiau trūko informacijos *FASTA* formato faile. Taip pat įrankis vizualiai neatvaizduoja genomo.

2.3. GeneMark

GeneMark yra interneto programinės įrangos svetainė, kuri suteikia sąsajas su *GeneMark* šeimos programomis. Jos skirtos prognozuoti *prokariotų, eukariotų* ir virusų genomų genus. Programos išvesties failai atsiunčiami į elektroninį paštą. Buvo išbandytos šios *GeneMark* šeimos programos:

- *GeneMark;*
- *GeneMark.hmm;*
- *GeneMarkS.*

Naudojimasis *GeneMark* svetaine yra labai paprastas. **Pirmiausiai** reikia pasirinkti programą kuri bus naudojama. Pasirinkus *GeneMark* programą, atsiranda vienintelis pasirinkimas – *prokariotai*, paspaudus ant jo, atsiranda langas, leidžiantis pasirinkti kelis nustatymus. Pirmiausiai yra įkeliamas failas su sekomis tik *FASTA* formatu, sekančiame lange pasirenkama *prokarioto* rūšis, čia pavyko rasti reikalingą *Sulfurimonas denitrificans*. Kitame lange keli pasirinkimai, susiję su išvesties failais, galimi pasirinkimai baltymų ir nukleotidų failai, pasirenkami abu. Taip pat yra leidžiama pasirinkti kodavimo potencialų grafiką, tačiau jis neleidžiamas *multi-FASTA* failams, taigi jis netinkamas, nes turimas genomas yra šio formato. **Paskutiniame žingsnyje** yra prašoma įvesti elektroninį paštą, į kurį bus nusiųsti išvesties failai. Failuose skyrėsi tik sekos, vienos išverstos į baltymų sekas (16 pav.), kitos į genų nukleotidų.

```
>orf_1 (AD-820-E17_NODE_1_length_88502_cov_101.2_ID_1, 3 - 323) translated
FVLSHPGGKQEFDNNIESIKSALKFYKPEECSETVINDIKAFVTHTPIKPAITAQKKELRYMERATGDFP
INIHDEKLNSLVKSIQKIIKDVNEQKSATFTQIKKQ*
>orf_2 (AD-820-E17_NODE_1_length_88502_cov_101.2_ID_1, 323 - 2125) translated
MTLEQTIKQLPQGAGIYQYFDINGHLLYIGKAKNLSNRVKSYFNFTPELKPNSNLSNRITKMILQTASLS
YIVVNSEHDALILENSLIKQLAPKYNILLRDDKTYPYIYIDNSSEYPRFDITRKIIKSADITYFGPYSVG
ARDILDSIYEVCKLVQKKACLKSKKACLYYQIDKCLAPCEFKVSHVRYKAELDLAQELIQNKKTLISKLT
EKMSFYAEEMRFEEAGELRDRIERISRSEIKSEIDFASNENYDIFVLHNSETRAVAVRIFMRNGKIISSS
HDFIQLNDGYDEDEFYQRVLLDFYAKEKPPIIAPILVMKKFSGLEIIAEHLSILFEKKALITAPSRGDKK
HLIDLAVLNAKELLKADKKQDLTKLFTEIKELLSLERIPNRVEIFDNSHMAGVATVGAMIVYENSQFDKK
SYRTYHLDAKDEYAQMRETLTRRVESFSKNSPPDLWIIDGGTTLLRLAVDILDSNGIFIDVIAISKEKID
AKSHRAKGKAHDILHTKDESFRLNPNDKRLQWIQNLRDEAHRSAIAFHKKTKLKLDKASKLLNLHGISEA
KIKKLLNHFGTFEALKELSEEEIASVLNIKDAQAIKNIYK*
```

16 pav. *GeneMark* programos išvesties *FASTA* failas

Išvesties failas atrodo neinformatyviai, tik išverčiamos sekos į baltymų sekas. Eilutėje, kuri apibūdina baltymo seką, yra tik perkopijuojama informacija iš pradinio failo, pažymima sekos

pradžia ir pabaiga. Galiausiai pažymima, jog baltymo seka yra identifikuota. Apie tai, koks galėtų būti baltymas nėra pateikiama jokios informacijos.

Naudojantis *GeneMark.hmm* atsirado papildomas pasirinkimas išvesties failo formato genų prognozavimui, galimi pasirinkimai tarp *LST* ir *GFF* formatų. Naudojantis *GeneMarkS* prie formato pasirinkimo atsirado dar ir sekos tipo pasirinkimas: prokariotas, eukariotas, virusas, fagai, *EST/cDNA*, mes pasirinkome prokarioto seka. Abiem programom failo formatas parinktas *LST*. Visi kiti nustatymai tokie patys, kaip ir naudojantis *GeneMark* programa. Į elektroninį paštą failai atsiunčiami taip pat iš karto. Abu išvesties *FASTA* failai atrodo identiškai (17 pav.).

```
>gene_1|GeneMark.hmm|106_aa|+|3|323 >AD-820-E17_NODE_1_length_88502_cov_101.2_ID_1
FVLSHPGGKQEFDNNIESIKSALKFYKPEECSETVINDIKAFVTHTPIKPAITAQKKELR
YMERATGDFPINIHDEKLNSLVKSIQKIIKDVNEQKSATFTQIKKQ

>gene_2|GeneMark.hmm|600_aa|+|323|2125  >AD-820-E17_NODE_1_length_88502_cov_101.2_ID_1
MTLEQTIKQLPQGAGIYQYFDINGHLLYIGKAKNLSNRVKSYFNFTPELKPNSNLSNRIT
KMILQTASLSYIVVNSEHDALILENSLIKQLAPKYNILLRDDKTYPYIYIDNSSEYPRFD
ITRKIIKSADITYFGPYSVGARDILDSIYEVCKLVQKKACLKSKKACLYYQIDKCLAPCE
FKVSHVRYKAELDLAQELIQNKKTLISKLTEKMSFYAEEMRFEEAGELRDRIERISRSEI
KSEIDFASNENYDIFVLHNSETRAVAVRIFMRNGKIISSSHDFIQLNDGYDEDEFYQRVL
LDFYAKEKPPIIAPILVMKKFSGLEIIAEHLSILFEKKALITAPSRGDKKHLIDLAVLNA
KELLKADKKQDLTKLFTEIKELLSLERIPNRVEIFDNSHMAGVATVGAMIVYENSQFDKK
SYRTYHLDAKDEYAQMRETLTRRVESFSKNSPPDLWIIDGGTTLLRLAVDILDSNGIFID
VIAISKEKIDAKSHRAKGKAHDILHTKDESFRLNPNDKRLQWIQNLRDEAHRSAIAFHKK
TKLKLDKASKLLNLHGISEAKIKKLLNHFGTFEALKELSEEEIASVLNIKDAQAIKNIYK
```

17 pav. *GeneMark.hmm* ir *GeneMarkS* išverstos baltymų sekos

Išvesties faile matoma, jog anotacijos eilutėje prisidėjo skaičius rodantis baltymo sekos ilgį. Šios informacijos neteikė nei vienas įrankis. Pastebėta, kad rodoma sekos pradžia, pabaiga ir pradinio genomo dalies pavadinimas.

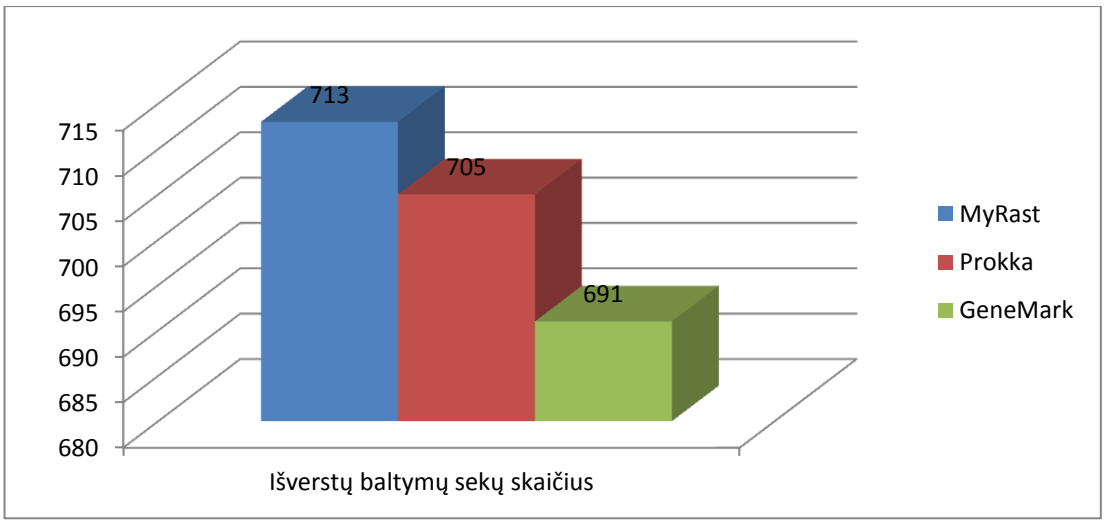
Išvesties failai tarp *GeneMark.hmm* ir *GeneMarkS* visiškai nesiskiria. Tačiau palyginus juos su *GeneMark* programos išvesties failu, atsiranda gan didelis skirtumas, skiriasi rastas sekų skaičius:

- GeneMark.hmm– 735 rastos sekos;
- GeneMarkS – 735 rastos sekos:
- GeneMark– 691 rastos sekos.

Dėl rastų skirtingų kiekių sekų, galima daryti išvadą, kad tai gali būti susiję su *GeneMark.hmm* ir *GeneMarkS* programose pasirinktu genų prognozavimo algoritmu *LST*, nes šie failai tarpusavyje nesiskiria.

2.4. Baltymų sekų palyginimas

Atlikus *Sulfurimonas denitrificans* genomo anotacijas su trimis skirtingais bioinformatikos įrankiais ir gavus 3 *FASTA* formato failus su identifikuotomis baltymų sekomis svarbu palyginti juos tarpusavyje. Šis palyginimas reikalingas, nes atlikus anotacijas gauti rezultatai skiriasi (18 pav.).



18 pav. Išverstų baltymų sekų skaičius su kiekvienu įrankiu

Rezultatai skiriasi nežymiai. Vadinasi, tas pats genomas su tomis pačiomis *DNR* sekomis yra identifikuojamas skirtingai. Todėl atliksime visų *FASTA* failų palyginimą. Iš viso buvo atlikti 3 palyginimai:

- *MyRast* su *Prokka*;
- *MyRast* su *GeneMark*;
- *Prokka* su *GeneMark*.

Parašytas *Python* skriptas, kuris atlieka baltymų sekų palyginimą ir išveda sekas, kurios yra nerandamos ieškomame faile t.y. baltymų sekų skirtumas tarp *FASTA* failų.

Trumpai apžvelgsime sukurto *Python* skripto atliekamus žingsnius:

- nuskaitomi failai;
- sukarpomas failas į atskiras sekas;
- kiekviena seka suskirstoma į mažesnes sekas po 20 raidžių;
- ieškoma mažesnių sekų kitame baltymų sekų faile;
- spausdinami rezultatai.

Pakomentuotas *Python* skriptas, kuris atlieka baltymų sekų lyginimą (19 pav.).

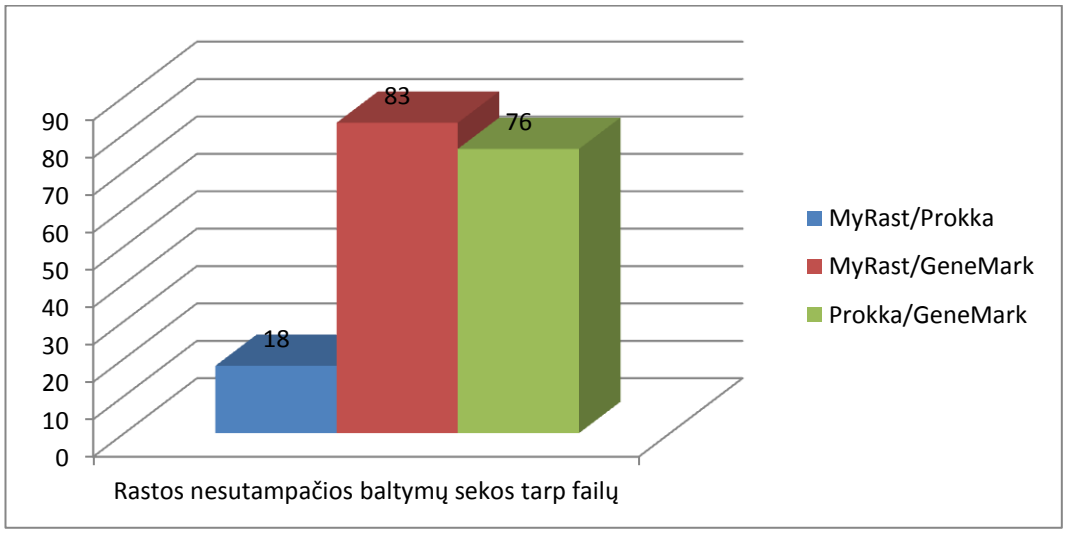
```
import re

countDisagreement = 0
searchSeq = open("MyRastGalutinis.fa", 'r')            # Ivesties seka kurios seku ieskosime
searchSeqRes = open("out.txt", 'w')                    # Rasome rezultatus sutvarkyto failo
for line in searchSeq:                                 # Pakeiciamos anotacijos eilutes i simboli '>', tai atliekama,
    if line[0] == '>':                                 # nes veliau bus galima faila sukarpyti pagal si simboli ir
    newline = line.replace(line, '>')                  # turesime tik baltymo sekas, be anotacijos eiluciu
        searchSeqRes.write(newline + '\n')
    else:
    searchSeqRes.write(line)
searchSeq.close()
searchSeqRes.close()

searchSequences = open("out.txt", "r")                 # Pakeistas failas imamas kaip ivestis
InSearchSeq = open("bacteria.faa", "r")                # Ivesties failas kuriame ieskosime seku
Result = open("testResult", "w")                       # Rezultatu rasymas
InSearch = ''.join([line.replace('\n', '') for line in InSearchSeq.readlines()]) # nuskaitomas visas tekstas ir
                                                       # istrinami naujos eilutes simboliai
Search = ''.join([line.replace('\n', '') for line in searchSequences.readlines()])
result = Search.split('>')                             # Sukarpomas tekstas pagal simboli '>'
m = len(result)
for i in range(1,m):                                   # Apdorojama po viena elementa
    UpperLetter = re.sub("[^A-Z]", "", result[i])      # Paliekamos tik didziosios raides
    SmallSeq = [UpperLetter[each:each+20]+'\n' for each in xrange(0,len(UpperLetter),20)]  # Seka sukarpoma po 20 simboliu
    for i in range(len(SmallSeq)):
    SubSmall = re.sub("[^A-Za-z]", "", SmallSeq[i])
    if SubSmall not in InSearch:                       # Ieskoma mazuju seku kitame faile
        countDisagreement +=1                          # Jei nerandama pridedamas vienetas
        if len(SmallSeq)*0.2 < countDisagreement:      # Jei nerastu mazuju seku skaicius didesnis uz visas vieno baltymo
            Result.write(str(UpperLetter) + '\n')      # mazasias sekas, tuomet seka rasoma i faila, kaip nesutampanti seka
    countDisagreement=0
Result.close()
lines = []
outfile = open('testResult1', 'w')                     # Istrinamos vienodos eilutes is galutinio failo
for line in file('testResult'):
    if line not in lines:
        lines.append(line)
        outfile.write(line)
outfile.close()
```

19 pav. *Python* skriptas, kuris atlieka baltymų sekų lyginimą

Atlikus palyginimą su visais trimis *FASTA* failais, gauti skirtingi rezultatai (20 pav.).



20 pav. Nesutampančių baltymų sekų kiekis tarp išvesties failų

Diagramoje matoma, kad rezultatai labai stipriai išsiskyrė lyginant su *GeneMark* įrankiu, tiek lyginant su *MyRast* 83 skirtingos sekos, tiek su *Prokka* 76 skirtingos sekos. Kadangi tokie dideli skirtumai tarp įrankių, didžiąją dalį tų sekų, kurios nurodytos kaip skirtingos, buvo patikrintos rankiniu būdu, kad įsitikinti, jog skriptas gerai atlieka savo darbą.

Daroma išvada, kad *GeneMark* programa nepatikimai anotuoja, bent jau šiam konkrečiam *Sulfurimono* genomui. Kadangi didelis neatitikimas tarp sekų, nors buvo leidžiamas 20% nesutapimas.

Taip pat rasti panašūs skirtumai lyginant *GeneMark* su *MyRast* ir *Prokka*. Pastarųjų dviejų įrankių lyginimas, davė realius rezultatus. 18 skirtingų baltymų sekų nėra didelis kiekis atsižvelgiant į tai, jog ir pats identifikuotų sekų skaičius skiriasi 8 sekomis. Skiriasi ir pačių įrankių metodai identifikuojant baltymus. Visos šios 18 sekų buvo patikrintos *Blast n*audojantis skriptu. Rezultatai buvo pasiskirstę tolygiai atsižvelgiant į pirmoje vietoje esančias prognozes:

- Trečdalis buvo nežinomi baltymai;
- trečdalis priklausė *Sulfurimonas* bakterijai;
- trečdalis buvo nurodyti kaip kitų organizmų baltymai.

Tolimesniame tyrime remiamasi tik tomis baltymų sekomis, kurios buvo vienodos lyginant *MyRast* ir *Prokka* įrankių išvestis.

2.5. Fermentų paieška

Fermentas – baltyminis katalizatorius, kuris paspartina organizme vykstančias chemines reakcijas tūkstančius kartų. Be fermentų šios reakcijos nevyktų arba vyktų labai lėtai ir organizmai negalėtų egzistuoti. Fermentai yra sudėtingos organinės medžiagos: vienkomponenčiai – sudaryti tik iš baltymų ir dvikomponenčiai – į kurių sudėtį, be baltymų, įeina ir kitos medžiagos, dažniausiai vitaminai. Šiuo metu ištirta daugiau kaip 3000 fermentų.

Fermentai yra svarbios medžiagos organizmui, todėl jas svarbu surasti ir pažymėti galimus fermento kelius (angl. pathway). Kurie baltymai turi fermentus atskirai ieškoti nereikės, tai jau atliko *MyRast* įrankis anotuodamas genomą. Prie kiekvienos sekos, kuri turi fermentus pažymėta, pavyzdžiui (EC 3.4.21.53). Toks žymėjimas nurodo tam tikrą fermentą pagal fermentų klasifikaciją.

Toliau darbe buvo ieškoma kiekvieno fermento galimi keliai (angl. pathway). Kadangi iš viso palygintame faile yra 212 fermentų, pirmiausiai buvo parašytas nedidelis skriptas. Kuris iš

failo iškerpa visas eilutes, kuriose yra fermentų pavadinimai. O iš tų eilučių paima jau tik pačių fermentų pavadinimus ir taip gaunamas sąrašas visų faile esančių fermentų.

Toliau, parsiunčiama visa informacija apie tuos fermentus. Tam yra naudojama *Kegg* duomenų bazė. *Kegg* tai yra rinkinys duomenų bazių susijusių su genomais, biologiniais keliais, ligomis, narkotikais ir cheminėmis medžiagomis. *Kegg* naudojama bioinformatikos tyrimams ir švietimui, įskaitant duomenų analizę genomikoje, metagenomikoje, metabolomikoje. Tam, kad parsisiųsti informaciją apie kiekvieną fermentą parašytas nedidelis skriptąs, kuris automatiškai parsiunčia visus *html* failus apie kiekvieną nurodytą fermentą (21 pav.).

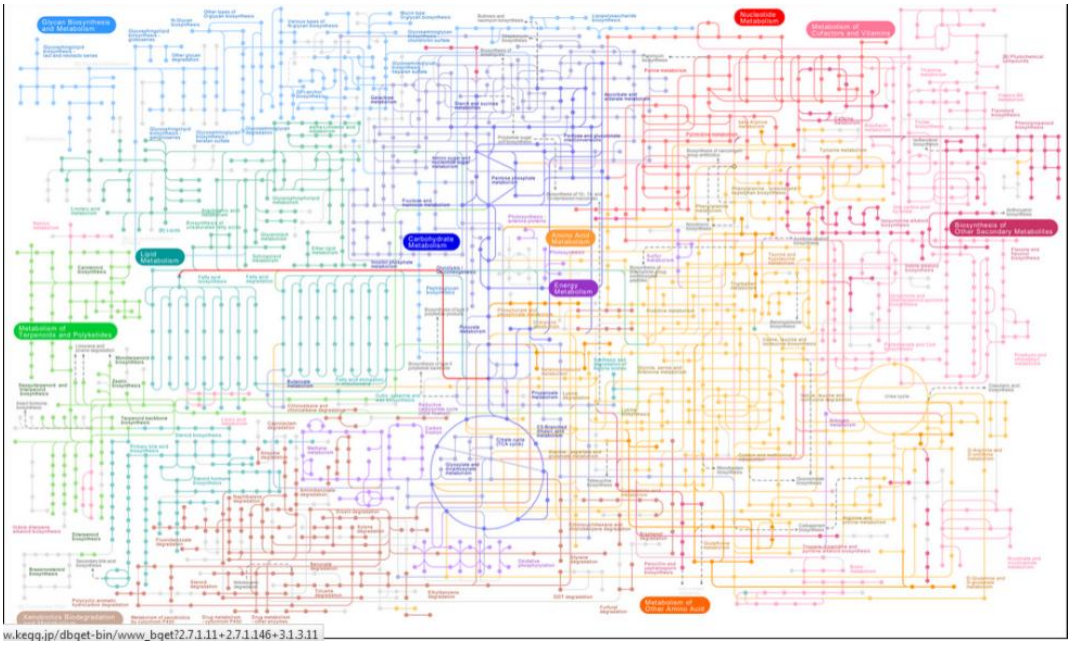
```
#!/usr/local/bin/python

import urllib

seq = open("RastKeggEC1", "r")
seq1 = seq.readlines();
for line in seq1:
    url = "http://www.kegg.jp/dbget-bin/www_bget?ec:" + line
    name = "ec" + line
    print line
    urllib.urlretrieve(url, filename=name)
```

21 pav. Skriptas skirtas parsiųsti *html* failus su informacija apie fermentus

Parsiųstame faile yra reikiama informacija apie kiekvieno fermento galimus kelius (angl. pathway). Fermento kelias nurodomas kaip internetinis adresas, į kurį nuėjus yra vizualiai pavaizduojamas tam fermentui priklausantis kelias. Taip atrodo fermentui *ec6.4.1.2* priklausantis metabolinis tinklas (22 pav.).



22 pav. Vaizduoja fermento ec6.4.1.2 metabolinį tinklą

Iš viso parsiųsta 212 *html* failų, iš kurių reikia paimti galimų kelių adresus, kurie parodo konkretaus fermento kelius. Tam buvo parašytas skriptas, kuris nusiskaito visų failų pavadinimus į sąrašą, po to ima po vieną failą, jį atidaro ir visą nuskaito, tuo pačiu ištrindamas naujos eilutės simbolius. Toliau suskaičiuojama kiek faile yra kelių ir tiek kartų iškerpami iš failo reikalingų kelių pavadinimai, suformuojamas adresas ir rezultatai rašomi į failą (23 pav.).

```
ec2.1.1.207
ec1.2.1.38
http://www.kegg.jp/kegg-bin/show_pathway?ec00330+1.2.1.38
http://www.kegg.jp/kegg-bin/show_pathway?ec01100+1.2.1.38
http://www.kegg.jp/kegg-bin/show_pathway?ec01110+1.2.1.38
http://www.kegg.jp/kegg-bin/show_pathway?ec01130+1.2.1.38
ec2.7.6.3
http://www.kegg.jp/kegg-bin/show_pathway?ec00790+2.7.6.3
http://www.kegg.jp/kegg-bin/show_pathway?ec01100+2.7.6.3
```

23 pav. Pavaizduota, kaip atrodo sąrašas su atrinktomis nuorodomis į fermentų kelius

Kaip matome pavaizduoti fermentai ir jiems priklausantys keliai. Nuėjus tais adresais bus matomas vizualus kelių pavaizdavimas. Kai kurie fermentai kelių neturi, po jais nieko nerašoma.

2.6. Genomo naršyklė

Bioinformatikoje genomo naršyklė tai grafinė sąsaja, pavaizduojanti informaciją iš biologinės duomenų bazės genomo duomenims. Genomo naršyklė leidžia tyrėjams vizualizuoti ir naršyti pasirinktą genomą su anotuotais duomenimis, genų prognozavimu, baltymais ir t.t. Anotuoti duomenys dažniausiai yra iš kelių skirtingų šaltinių.

Paskutinis tyrimo žingsnis yra panaudoti visus gautus rezultatus kuriant genomo naršyklę. Tai yra rezultatus, kurie buvo gauti palyginus *MyRast* ir *Prokka* baltymų sekas ir surastus fermentų galimus kelius. Iš viso palygintame faile yra 695 sutampančios baltymų sekos. Todėl norint sukurti genomo naršyklę su anotuojama informacija, šiam darbui atlikti buvo nuspręsta parašyti skriptą, kuris generuotų *html* failą su atvaizduojama genomo naršykle. Skriptas parašytas *Python* kalba. Šis skriptas yra universalus, jis gali būti naudojamas ir kitiems su *MyRast* įrankiu identifikuotiems genomams, tačiau jame turėtų būti tik baltymų sekos t.y. neįtrauktos DNR sekos į išvesties failą.

Toliau trumpai apžvelgimas parašytas skriptas. Išskiriami pagrindiniai žingsniai:

- nuskaitomi failai (baltymų sekų ir fermentų);
- į failą įrašomos pirmosios eilutės reikalingos *html* failui pradėti;
- nuskaitomi ir apskaičiuojami baltymų sekų ilgiai, id ir pavadinimai;

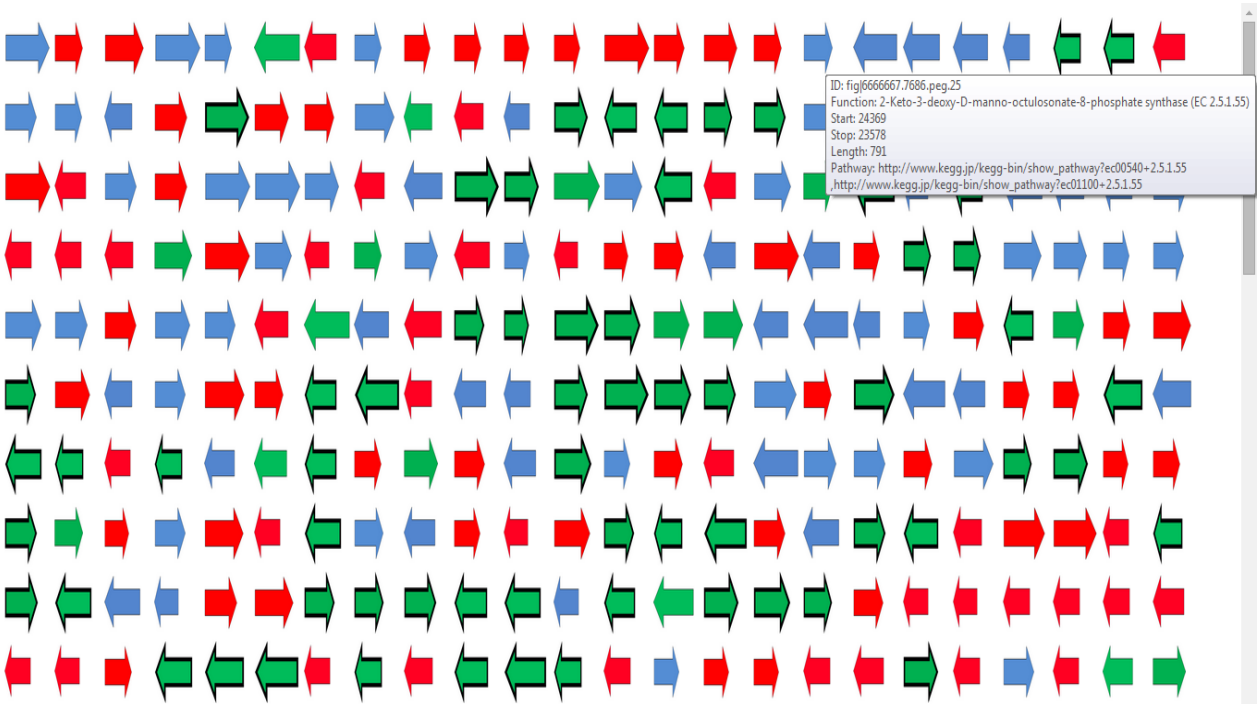
- surandami baltymai kuriems priskirti fermentai ir priskiriami fermentų keliai;
- patikrinama baltymų sekų kryptis;
- apskaičiuojamas rodyklės dydis, kiekvienam baltymui žymėti;
- generuojamas paprastas *html* failas su reikalinga informacija;
- tuo pačiu generuojamas atskiras *html* failas su baltymų sekomis ir jų anotacija.

Pirmiausiai iš bendro *MyRast* ir *Prokka* failo yra nuskaitomos baltymų sekos su anotacija apie juos ir iš atskiro failo nuskaitomi fermentai su jų keliais (angl. pathway). Vėliau įrašomos kelios eilutės į failą, kurios reikalingos *html* failo pradžiai. Iš nuskaitytų baltymų anotacijų yra nuskaitoma baltymo sekos pradžia ir pabaiga bei apskaičiuojamas jų ilgis. Taip pat nuskaitoma id ir pavadinimas. **Id** tai kiekvienos baltymo sekos anotacijoje pirmieji simboliai pavyzdžiui id = fig|6666667.7686.peg.1, sekančių sekų id skiriasi tik paskutiniu skaičiumi. **Pavadinimas** tai baltymo sekos pavadinimas, pavyzdžiui, *Excinuclease ABC subunit C*. Toliau, kiekvieno baltymo anotacijoje ieškoma ar jam yra priskirtų fermentų, jei taip iškerpamas to fermento pavadinimas. Atskirame faile ieškoma ar tam fermentui yra priskirtų kelių, jei taip jie paimami ir dedami į *html* failą kartu su ilgiu, pavadinimu ir id. Jei baltymo seka neturi jam priskirtų fermentų, tuomet darbas tęsiamas toliau. Kitame žingsnyje yra nustatoma, kokia baltymo sekos kryptis. Jei iš baltymo pradžios atėmus pabaigą gaunamas minusinis skaičius tai rodyklės, kuri simbolizuoja baltymą, kryptis bus į kairę pusę, jei pliusinis tuomet kryptis į dešinę pusę. Rodyklės yra paimamos kaip paveikslėliai rodančios viena arba kita kryptimi. Jos yra keturių spalvų: raudona – nežinomi baltymai, žalia – fermentai neturintys kelių, mėlyna – likusieji baltymai ir žalia su juodom linijom – fermentai turintys kelius. Kiekvienam baltymui yra apskaičiuojamas jo ilgis ir nustatomas jo rodyklės dydis. Turint visą reikalingą informaciją apie baltymą, galima kiekvienam generuoti *html* kodą, kuris nurodys visus duomenis apie jį. Taip pat kiekvienam baltymui yra suteikiama nuoroda į kitą *html* failą, kuriame yra sąrašas visų baltymų sekų su jų pavadinimais. Jeigu tai fermentas, po juo rašomos nuorodos į fermentų kelius. Šis sąrašas skirtas tam, kad vartotojas prireikus galėtų pamatyti kiekvieną baltymą individualiai.

Įvykdžius skriptą yra sugeneruojami du *html* failai. Vienas iš jų genomo naršyklė (24 pav.), kitas baltymų sekų sąrašas.

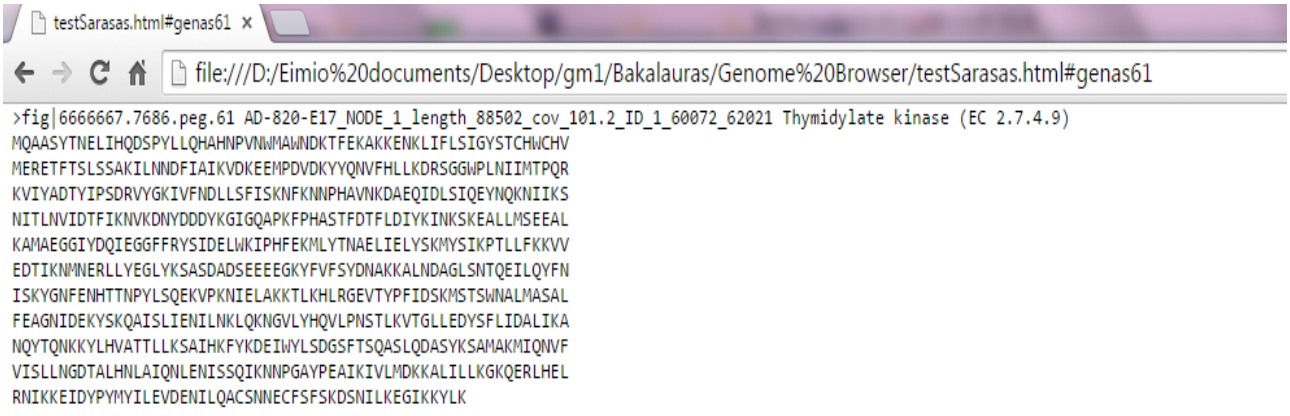
Kaip matyti paveiksle, daug skirtingo dydžio ir krypčių rodyklių. Kiekviena iš jų atvaizduoja tam tikrą *Sulfurimonas denitrificans* genomo baltymą. Iš viso yra 695 baltymų sekos ir pavaizduota tiek pat rodyklių. Užvedus pelę ant bet kurios rodyklės, kaip matyti 24 paveiksle, atsiranda laukelis, kuriame yra informacija apie būtent tą baltymą. Šiame laukelyje rodoma

informacija: ID, funkcija, baltymo sekos pradžia ir pabaiga, jos ilgis ir fermentų keliai, jei baltymas turi fermentų.



24 pav. *Sulfurimonas denitrificans* genomo naršyklės dalis

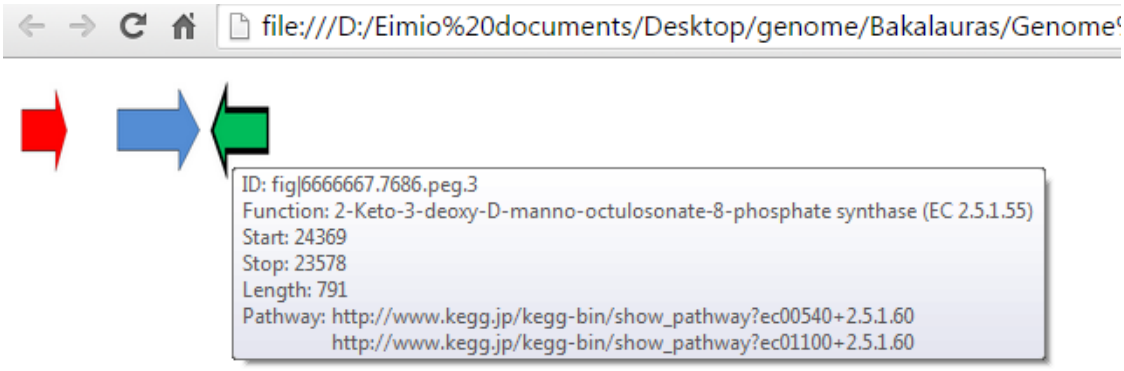
Paspaudus ant bet kurios rodyklės vartotojas yra nukeliamas į kitą *html* failą, kuriame yra pavaizduotas būtent tas baltymas ant kurio buvo paspausta (25 pav.).



25 pav. Naršyklės langas paspaudus ant rodyklės, rodoma baltymo seka

Apačioje pateikiamas pavyzdys, kaip atrodytų genomo naršyklė jeigu joje būtų tik trys baltymai. Visas genomo naršyklės funkcionalumas išliks. Užvedus pelę bus rodoma informacija apie baltymą, o paspaudus rodoma pati baltymo seka (26 pav.).

Šiuo pavyzdžiu parodoma, jog visiškai nesvarbu kiek baltymų sekų yra genome, ar tai būtų 3 sekos ar 3000 sekų, skriptas sugeneruoja *html* kodą, bet kokiam skaičiui baltymų, neprarandant genomo naršyklės funkcionalumo.



26 pav. Genomo naršyklė iš bet kokių trijų *Sulfurimonas denitrificans* baltymų

Toliau nagrinėjama, kaip atrodo sugeneruotas *html* kodas ir kokias funkcijas jis atlieka. Tai iš paprasčiausių elementų sudarytas kodas, kuris yra sugeneruojamas *Python* kalba parašyto skripto. Generavimo metu į *html* kodą yra sudedama visa reikalinga informacija susijusi su baltymo seka (27 pav.).

```
<td width = 50px><a href = "testSarasas.html#genas25"> <img src="arrowG1.
png"height=50% width=70%  title="ID: fig|6666667.7686.peg.25
Function: 2-Keto-3-deoxy-D-manno-octulosonate-8-phosphate synthase (EC
2.5.1.55)
Start: 24369
Stop: 23578
Length: 791
Pathway: http://www.kegg.jp/kegg-bin/show_pathway?ec00540+2.5.1.55
,http://www.kegg.jp/kegg-bin/show_pathway?ec01100+2.5.1.55
"></a></td>
<td width = 50px><a href = "testSarasas.html#genas26"> <img src="arrowR1.
png"height=50% width=72%  title="ID: fig|6666667.7686.peg.26
Function: FIG00852003: hypothetical protein
Start: 25316
Stop: 24468
Length: 848"></a></td>
```

27 pav. Ištrauka iš genomo naršyklės *html* kodo

Toliau analizuojama iš kokių elementų susideda *html* kodas. Pirmiausiai visas *html* kodas yra dedamas į lentelę, tam kad išlaikyti tvarkingą rodyklių pasiskirstymą. <*tr*> simboliai atskiria eilutes, kiekvienoje eilutėje yra dedamas tik tam tikras skaičius rodyklių, kad nepersidengtų viena rodyklė su kita. Kiekvienas baltymas prasideda simboliu <*td*> jis nurodo *html* lentelės vieną langelį. Toliau pačiame <*td*> nurodoma plotis tarp langelių, kad nebūtų, per dideli ar per

maži tarpai. Nuoroda į kitą failą atliekama pasinaudojant *href* elementu tam, kad būtų parodyta baltymo seka. <*img*> elementu yra nurodomas paveikslėlis, kuris bus dedamas į lentelės langelį, taip pat nustatomas paveikslėlio aukštis ir plotis, jie nustatomi apskaičiuojant pagal baltymo sekos ilgį. Taip pat elementas *title* tam, kad užvedus būtų rodoma informacija apie baltymą.

Šis *html* kodas yra sukurtas paprasčiausiomis priemonėmis, bet turintis reikalingą funkcionalumą. Genomo naršyklė puikiai vaizduojama *Google Chrome* naršyklės. Naudojant kitas naršykles gali būti iškraipomas vaizdas.

Išvados

Išanalizavus bakterijos *Sulfurimonas denitrificans* genomą, įvykdžius anotacijas su trimis skirtingais bioinformatikos įrankiais ir atlikus visų įrankių išvesčių palyginimus galima daryti išvadą, jog *MyRast* ir *Prokka* įrankiai anotaciją atliko patikimai, greitai ir suteikdami naudingos informacijos, t.y. buvo identifikuotas ir anotuotas *Sulfurimonas denitrificans* genomas. *GeneMark* įrankis dideliu sekų skirtumu išsiskyrė iš visų įrankių, todėl šio įrankio išvestis buvo nebenaudojama tolimesniam tyrimui. Taip pat, pasinaudojant *MyRast* anotacija, kuri identifikuoja ir fermentus, buvo surasti fermentų keliai ir panaudoti genomo naršyklės kūrime. Galima teigti, kad didžiausią naudą vartotojui suteikia įrankis *MyRast*, suteikdamas aiškią anotaciją apie genomą ir leisdamas tą pačią anotaciją matyti vizualiai, bei naršyti po ją.

Remiantis atliktomis genomo anotacijomis bei išanalizuota informacija apie genomą, buvo sukurtas *Python* skriptas, kuris generuoja *html* failą, t.y. genomo naršyklę. Šis skriptas yra tinkamas bet kokio genomo anotacijai atliktai su *MyRast* įrankiu. Sukurta genomo naršyklė vaizduoja baltymus kaip rodykles, kurios yra išskirtos keliomis spalvomis, identifikuodamos skirtingas baltymų grupes. Užvedus pelę ant baltymo rodoma informacija apie jį: ID, funkcija, baltymo pradžia, pabaiga ir jo ilgis ir nuoroda į fermentų kelius, jeigu tokie žinomi. Paspaudus ant baltymo yra rodoma baltymo seka. Genomo naršyklė yra naudingas įrankis analizuojant genomą, suteikia suvokimo, aiškumo ir galimybę nagrinėti kiekvieną baltymą individualiai.

Apibendrinant gautus rezultatus, galima daryti tokias išvadas:

- *MyRast* įrankis yra paprasčiausias, patogiausias ir aiškiausias iš visų trijų naudotų įrankių;
- visi įrankiai identifikavo ir anotavo genomą, tačiau *Prokka* ir *MyRast* suteikia daugiau ir tikslesnės reikalingos informacijos apie tiriamą genomą;
- sukurti skriptai, kurie:
    - palygina anotacijų įrankių išvesties failus ir leidžia gauti nesutampančias sekas, kaip rezultatą;
    - randa fermentus ir parsiunčia informaciją apie juos, taip pat leidžia tą informaciją apdoroti;
    - generuoja paprastą *html* kodą, kuris vaizduoja tiriamo *Sulfurimonas denitrificans* genomo naršyklę.

Literatūra

[1] Anders F Andersson, L. R. (2010). Pyrosequencing reveals contrasting seasonal dynamics of taxa within Baltic Sea bacterioplankton communities. *The ISME Journal* .

[2] Daniel PR Herlemann, M. L. (2011). Transitions in bacterial communities along the 2000km salinity gradient of the Baltic Sea. *The ISME Journal* .

[3] Fukunaga Y, K. M. (2009). Phycisphaera mikurensis gen. nov., sp. nov., isolated from a marine alga, and proposal of Phycisphaeraceae fam. nov., Phycisphaerales ord. nov. and Phycisphaerae classis nov. in the phylum Planctomycetes. *The Journal of general and applied microbiology* .

[4] Garrity, G. B. (2001). *Bergey's Manual of Systematic Bacteriology :Volume One : The Archaea and the Deeply Branching and Phototrophic Bacteria Editor.*

[5] Yuchen Han, M. P. (August 29, 2014). The Role of Hydrogen for Sulfurimonas denitrificans' Metabolism. *Plos* .

[6] John Besemer, M. B. (2005). GeneMark: web software for gene finding in prokaryotes, eukaryotes and viruses.

[7] *Microbe Wiki*. (n.d.). Nuskaityta iš https://microbewiki.kenyon.edu/index.php/Sulfurimonas_denitrificans

[8] Ramy K Aziz8, 9. D. (2008). The RAST Server: Rapid Annotations using Subsystems.

[9] Seemann, T. (2014). Prokka: rapid prokaryotic genome annotation. *Bioinformatics Advance Access* .

[10] Stefan Schouten, E. C. (2008). Intact Membrane Lipids of „Candidatus Nitrosopumilus maritimus," a Cultivated Representative of the Cosmopolitan Mesophilic Group I Crenarchaeota.

[11] Team, S. (n.d.). Genome Annotation by SEED Team.

Santrauka

Bakalauro darbo autorius Eimantas Paspirgėlis. Darbo tema „Baltijos jūros vandens mikroorganizmų genominių sekų tyrimas“. Bakalauro darbe analizuojama bakterijos *Sulfurimonas denitrificans* genomas su trimis skirtingais anotacijos įrankiais *MyRast*, *Prokka* ir *GeneMark*. Parašytas skriptas, kuris gavus visų trijų įrankių rezultatus lygina juos tarpusavyje tam, kad būtų surastos ir išrinktos tiksliausios baltymų sekos. Taip pat parašytas skriptas su kuriuo ieškoma fermentų ir galimų jų kelių (angl. pathway). Toliau, remiantis atliktomis genomo anotacijomis ir išanalizuota informacija apie genomą, buvo sukurtas *Python* skriptas, kuris generuoja *html* failą, t.y. genomo naršyklę. Šis skriptas yra tinkamas bet kokio genomo anotacijai atliktai su *MyRast* įrankiu. Genomo naršyklė yra naudingas įrankis analizuojant genomą, suteikia suvokimo, aiškumo ir galimybę nagrinėti kiekvieną baltymą individualiai.

Raktiniai žodžiai : Sulfurimonas, Sulfurimonas denitrificans, genomo naršyklė, MyRast, Prokka, GeneMark, fermentai, baltymai.

Summary

Bachelor's thesis author Eimantas Paspirgėlis. Topic is „Study of Baltic Sea Microbial Genomic Sequences“. Bachelor's thesis analyzes the bacteria Sulfurimonas denitrificans genome with three different annotation tools MyRast, Prokka and GeneMark. Created a script which on receipt all the results of the three tools compare them with each other to be detected and selected the most accurate protein sequences. Also created the script with wich was searched enzymes and their possible pathway. Further, on the basis of conducted genome annotation and analysis of information about the genome, it was developed Python script that generates an HTML file, ie genome browser. The script is suitable for any kind of annotating the genome conducted with MyRast tool. Genome Browser is a useful tool for analyzing the genome, gives understanding, clarity and the ability to examine each protein individually.

Keywords: Sulfurimonas, Sulfurimonas denitrificans, genome browser, MyRast, Prokka, GeneMark, enzyme, protein.

Priedai

1. Skriptas, kuris atlieka baltymų sekų palyginimą.

```
#!/usr/local/bin/python

# Eimantas Paspirgelis, Bioinformatika 4 kursas
# Programa atlieka baltymu seku palyginima
# Naudojimas: python Palyginti

import re

countDisagreement = 0
# Ivesties seka kurios seku ieskosime
searchSeq = open("MyRastGalutinis.fa", 'r')
# Rasome rezultatus sutvarkyto failo
searchSeqRes = open("out.txt", 'w')
# Pakeiciamos anotacijos eilutes i simboli '>', tai atliekama,
# nes veliau bus galima faila sukarpyti pagal si simboli ir
# turesime tik baltymo sekas, be anotacijos eiluciu
for line in searchSeq:
    if line[0] == '>':
    newline = line.replace(line, '>')
        searchSeqRes.write(newline + '\n')
    else:
    searchSeqRes.write(line)
searchSeq.close()
searchSeqRes.close()

# Pakeistas failas imamas kaip ivestis
searchSequences = open("out.txt", "r")
# Ivesties failas kuriame ieskosime seku
InSearchSeq = open("bacteria.faa", "r")
# Rezultatu rasymas
Result = open("testResult", "w")
# nuskaitomas visas tekstas ir istrinami naujos eilutes
simboliai
InSearch = ''.join([line.replace('\n', '') for line in
InSearchSeq.readlines()])
Search = ''.join([line.replace('\n', '') for line in
searchSequences.readlines()])
# Sukarpomas tekstas pagal simboli '>'
result = Search.split('>')
m = len(result)
# Apdorojama po viena elementa
for i in range(1,m):
    # Paliekamos tik didziosios raides
    UpperLetter = re.sub("[^A-Z]", "", result[i])
    # Seka sukarpoma po 20 simboliu
    SmallSeq = [UpperLetter[each:each+20]+'\n' for each in
xrange(0,len(UpperLetter),20)]
    for i in range(len(SmallSeq)):
```

```
    SubSmall = re.sub("[^A-Za-z]", "", SmallSeq[i])
    # Ieskoma mazuju seku kitame faile
    if SubSmall not in InSearch:
        # Jei nerandama pridedamas vienetas
        countDisagreement +=1
        # Jei nerastu mazuju seku skaicius didesnis uz visas
vieno baltymo
        # mazasias sekas, tuomet seka rasoma i faila, kaip
nesutampanti seka
        if len(SmallSeq)*0.2 < countDisagreement:
            Result.write(str(UpperLetter) + '\n')
    countDisagreement=0
Result.close()
lines = []
# Istrinamos vienodos eilutes is galutinio failo
outfile = open('testResult1', 'w')
for line in file('testResult'):
    if line not in lines:
        lines.append(line)
        outfile.write(line)
outfile.close()
```

2. Programa kuri iškerpa fermentų kelius ir suformuoja nuorodas.

```
#!/usr/local/bin/python
# -*- coding: utf-8 -*-

# Eimantas Paspirgelis, Bioinformatika 4 kursas
# Programa is daug failu isima fermentu kelius ir suformuoja
nuorodas
# Naudojimas: python EnzymeReference

import os
import re
from collections import Counter

# Nurodomas failas i kuri rasomi rezultatai
Result = open("keggResult", "w")
listNumber = 0
# Nurodomas aplankas kuriame yra visi html failai
path = 'test/'
# Visi failu pavadinimai nuskaitomi i sarasa
listing = os.listdir(path)
# Imama po viena faila
for infile in listing:
    # Atidaromas failas nuskaitymui
    fname = open(listing[listNumber],'r')
    # Nuskaitomas visas tekstas ir istrinami naujos eilutes
simboliai
    text = ''.join([line.replace('\n', '') for line in
fname.readlines()])
    # Suskaiciuojama kiek is viso faile yra keliu
```

```
    countPath = text.count("/kegg-bin/show_pathway")
    x = 0
    for i in range(countPath):
        # Sukarpomas failas pagal fermento kelio pradzia
        FirstSplit = text.split('/kegg-bin/show_pathway')
        # Nukerpama nereikalinga galune
        title = FirstSplit[i+1].split('"')
        # Suformuojama nuoroda
        Result.write("http://www.kegg.jp/kegg-bin/show_pathway"
+ (title[0]) + '\n')
    listNumber = listNumber+1
```

3. Programa, kuri generuoja html failą iš baltymų sekų(MyRast anotacija) ir fermentų galimų kelių.

```
#!/usr/local/bin/python
# -*- coding: utf-8 -*-

# Eimantas Paspirgelis, Bioinformatika 4 kursas
# Programa generuoja html faila is baltymu seku(MyRast
anotacija) ir fermentu keliu.
# Naudojimas: python Browser
# Nurodyti ivesties failus: sequenceFile - baltymu sekos,
enzymeFile - fermentu keliai.

import os
import re
from decimal import *

# baltymu seku ivesties failas
sequenceFile = open("RastGalutinis.fa", "r")
# rasomas html kodas su informacija apie baltymus
htmlFile = open("GenomoNarsykle.html", "w")
# rasomas baltymu seku sarasas
listHTML = open("Sarasas.html", "w")
# fermentu keliu ivesties failas
enzymeFile = open("FermentuKeliai", "r")
# nuskaitomas fermentu keliu failas
EcResRead = enzymeFile.readlines()
# rasomos html failo pradines eilutes
listHTML.write('<html>\n')
htmlFile.write('<html>\n<table>\n')
# nuskaitomas baltymu seku failas
sequenceRead = sequenceFile.readlines()
enzymeCount = 0
countTick = 0
countTick1 = 0
lineCount = 0
SizeProtein = 0
arrowLine = 24
max = 0
```

```
lastCurrentLine = 0
currentLine = 0
oneProteinEnzymes=[]
for i in sequenceRead:
    # skaiciuojamos viso failo eilutes
    lineCount +=1
    # atrenkamos tik anotacijos eilutes
    if i[0]=='>':
    # skaiciuojamos baltymu sekos
    countTick1 +=1
    # surandama baltymo pradzia
    number1 = i.split('ID_')[1].split('_',1)[1].split('
')[0].replace("_"," ").split(' ')[0]
    # surandama baltymo pabaiga
    number2 = i.split('ID_')[1].split('_',1)[1].split('
')[0].replace("_"," ").split(' ')[1]
    # surandamas baltymo pavadinimas
    name = i.split('ID_')[1].split(' ',1)[1]
    # apskaiciuojamas baltymo ilgis
    SizeProtein = int(number2) - int(number1)
    if SizeProtein < 0:
        SizeProtein = SizeProtein * (-1)
    if SizeProtein > max:
        max =SizeProtein
factor = max / 3
factor1 = factor/50
for i in sequenceRead:
    currentLine +=1
    if i[0]=='>':
    countTick +=1
    number1 = i.split('ID_')[1].split('_',1)[1].split('
')[0].replace("_"," ").split(' ')[0]
    number2 = i.split('ID_')[1].split('_',1)[1].split('
')[0].replace("_"," ").split(' ')[1]
    name = i.split('ID_')[1].split(' ',1)[1]
    # surandamas baltymo ID
    ID = i.split('>')[1].split(' AD')[0]
    # ieskoma ar tai fermentas
    if "(EC" in i:
        # iskerpamas fermento pavadinimas
        EC = i.split('(EC ')[1].split(')')[0]
        for g in EcResRead:
        if EC in g:
            # sudedami visi vieno fermento keliai
            oneProteinEnzymes.append(g)
    numResult = int(number2) - int(number1)
    # nustatoma rodykles kryptis ir rodykles spalva
    if numResult > 0:
        if 'hypothetical protein' in name:
            img = "arrowR.png"
        elif len(oneProteinEnzymes)!=0:
        img = "arrowGB.png"
        elif "(EC" in name:
```

```
            img =  "arrowG.png"
            else:
            img = "arrow.png"
        else:
            if 'hypothetical protein' in name:
                img = "arrowR1.png"
            elif len(oneProteinEnzymes)!=0:
            img = "arrowGB1.png"
            elif "(EC" in name:
            img =  "arrowG1.png"
            else:
            img = "arrow1.png"
        if numResult < 0:
            numResult = numResult * (-1)
            # skaiciuojama kada pradeti nauja rodykliu eilute
            if arrowLine - countTick == 23:
            htmlFile.write('<tr>\n')
            # skaiciuojamas rodykles dydis
        width = numResult / factor1 + 50
        if width > 100:
            width = 100
        if countTick <= arrowLine:
            # jei baltymas neturi fermento keliu rasoma si eilute
            if not oneProteinEnzymes:
                htmlFile.write('<td width = 50px><a href =
"Sarasas.html#genas'+str(countTick)+'"> <img src='+'"'+img+'"'+
'height=50% width='+str(width)+'%  title="ID: '+ID+'\nFunction:
'+name+'Start: '+str(number1)+'\nStop: '+str(number2)+'\nLength:
'+str(numResult)+'"></a></td>\n')
            # jei fermentas turi kelius rasoma si eilute
            else:
            htmlFile.write('<td width = 50px><a href =
"Sarasas.html#genas'+str(countTick)+'"> <img src='+'"'+img+'"'+
'height=50% width='+str(width)+'%  title="ID: '+ID+'\nFunction:
'+name+'Start: '+str(number1)+'\nStop: '+str(number2)+'\nLength:
'+str(numResult)+'\nPathway:
'+str(','.join(oneProteinEnzymes))+'"></a></td>\n')
            # uzbaigiama rodykliu eilute
            if countTick == arrowLine:
                htmlFile.write('</tr>\n')
                arrowLine += 24
        # generuojamas baltymu seku sarasas
        if lastCurrentLine == 0:
            lastCurrentLine = 0
        else:
            listHTML.write('<a id="genas'+str(countTick-
1)+'"><pre>\n')
            nulinam = 0
            # rasomos baltymu sekos
            for i in range(currentLine-lastCurrentLine):
                listHTML.write(sequenceRead[lastCurrentLine +
nulinam - 1])
                nulinam +=1
```

```
        # rasomi fermentu keliai
        if enzymeCount + 1 == countTick:
            listHTML.write('\nEnzyme pathway: '+'<a href = "' +
str('                '.join(lastProteinEnzymes)) +'">' + str('
'.join(lastProteinEnzymes)) + '</a>')
        if len(oneProteinEnzymes) != 0:
        lastProteinEnzymes = oneProteinEnzymes
        enzymeCount = countTick
    if countTick == countTick1:
        listHTML.write('<a
id="genas'+str(countTick)+'"><pre>\n')
            nulinam = 0
        for i in range(lineCount - currentLine +1):
            listHTML.write(sequenceRead[currentLine + nulinam -
1])
            nulinam +=1
    lastCurrentLine = currentLine
        # tarpai tarp baltymu seku

listHTML.write('\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n')
    listHTML.write('</pre></a>\n')
    oneProteinEnzymes = []
# baigiamas html failas
htmlFile.write('</tr>\n</table>\n</html>')
listHTML.write('</html>')
htmlFile.close()
```