

Математички факултет

Београд, Студентски трг 16

**ОДРЕЂИВАЊЕ ПРАВИЛА ПРИДРУЖИВАЊА ЗА
АНАЛИЗУ НЕУРЕЂЕНИХ СЕКВЕНЦИ ПРОТЕИНА**

– семинарски рад из Истраживања података 2 –

Ментори:

проф. др Ненад Митић

Студент:

Анђела Дамњановић, 59/2019

Београд, јул 2023

Table of Contents

| | |
|--|----|
| 1. Увод..... | 3 |
| 1.1 Биолошки оквир..... | 3 |
| 1.2 Софтвер који се користи..... | 7 |
| 1.2.1 StatRepeats..... | 7 |
| 1.2.2 Jupyter notebook..... | 7 |
| 1.2.3 Програмски језик Python..... | 8 |
| 1.2.4 Програмски језик Java..... | 8 |
| 1.2.5 Програмски језик R..... | 8 |
| 1.3 Скуп података за рад..... | 8 |
| 2. Претпроцесирање података и налажење директних и инверзних некомплементарних секвенци..... | 10 |
| 3. Налажење позиција неуређених региона и тражење додатних карактеристика процесирањем текста..... | 16 |
| 3.1 Налажење позиција неуређених региона..... | 16 |
| 3.2 Налажење додатних карактеристика претпроцесирањем текста..... | 17 |
| 3.3 Прављење фајла за налажење правила придруживања..... | 20 |
| 3.4 Прављење фајлова који садрже трансакције..... | 21 |
| 4. Правила придруживања и априори алгоритам..... | 22 |
| 4.1 Априори алгоритам..... | 22 |
| 4.2 Налажење правила придруживања..... | 23 |
| 5. Анализа..... | 36 |
| 6. Литература..... | 37 |

1. Увод

1.1 Биолошки оквир

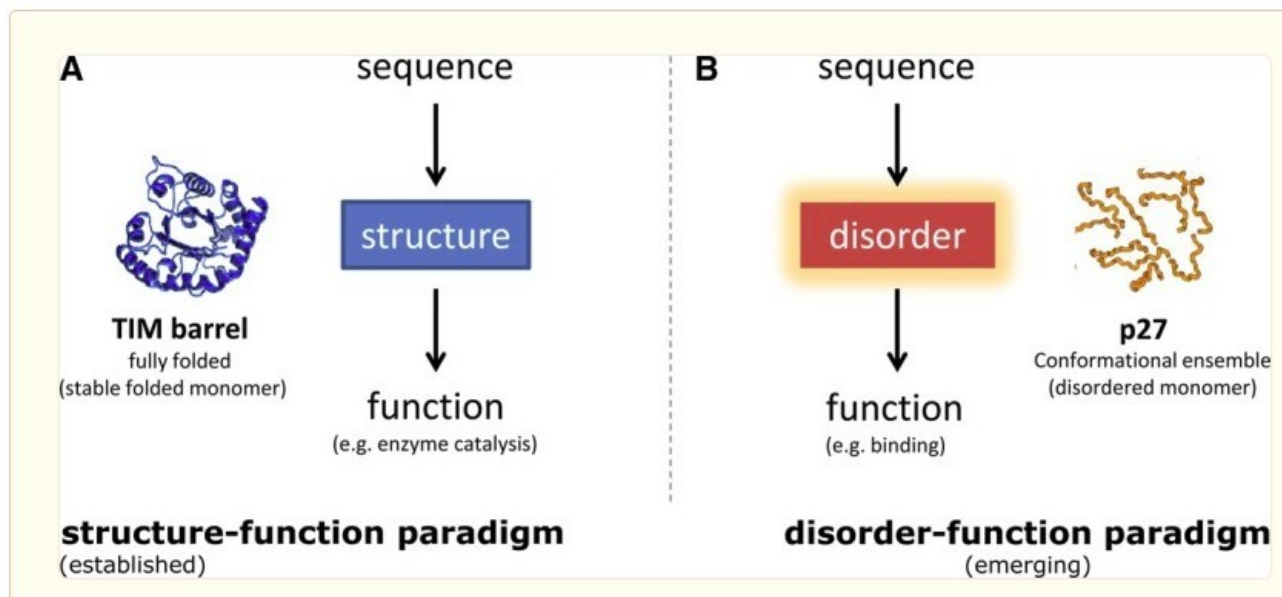
Проћеини (такође познати и под називом *беланчевине*) су природни полипептиди који су изграђени од *аминокиселина*. Они су крупни, комплексни молекули који имају много улога у људском телу: обављају већину посла у ћелијама, неопходни су да пружају структуру, регулацију и функцију телесним ткивима и органима, неки служе као антитела која бране организам, неки су ензими који потпомажу хемијске реакције у телу и стварање нових молекула, неки служе да преносе сигнале за координацију биолошких процеса, неки за себе везују атоме и мале молекуле из ћелије и преносе их са једног места у ћелији на друго, или чак кроз тело до других, удаљених ћелија.

Сваки протеин састоји се од стотина или хиљада мањих јединица (*аминокиселина*) поређаних у линеарне ланце и спојених међусобно пептидним везама између угљениковог атома и амино групе две аминокиселине. Секвенца аминокиселина у протеину дефинисана је у генима и садржана у генетичком коду. Генетички код одређују 20 „основних“ аминокиселина које могу ући у састав протеина. Свака аминокиселина је кодирана са 3 градивна блока ДНК (*нуклеотида*) који су одређени секвенцом гена.

Током 1960-их година прошлог века, Кристијан Анфинсен изнео је теорију да управо те секвенце аминокиселина одређују јединствен тродимензиони облик протеина, који даље одређује коју функцију дати протеин врши. Биохемијске студије које су Анфинсон и колеге спровели, заједно са јединственим увидима у молекуле који су добијени кристалографским проучавањем протеина као што су хемоглобин и бројни ензими, потврдио је претпоставку. Овај рад поставио је основу *секвенца-структура-функција* парадигме. И док се већина протеина и полипептидних сегмената "повинује" овом правилу, новија истраживања (спроведена у последњих неколико деценија) открила су да постоје полипептидни сегменти који одступају од терцијарне структуре. Уместо тога, они усвајају ансамбл различитих конформација, успевајући при томе да обављају своју функцију чак и у неуређеном/неструктурираном стању. Управо је таква њихова структура допринела томе да добију назив неуређене секвенце. Ове студије сада успостављају парадигму *секвенца-неуређеност-функција*, која каже да одређени полипептидни сегменти могу да врше своју функцију чак иако нису достигли дефинисану терцијарну структуру. Најскорија истраживања показала су

да преко 40% протеина свих еукариота садржи ове неуређене секвенце. Што је још важније, мутације у многим протеинима са неуређеним сегментима су умешане у људске болести, као што су неуродегенерација и рак.

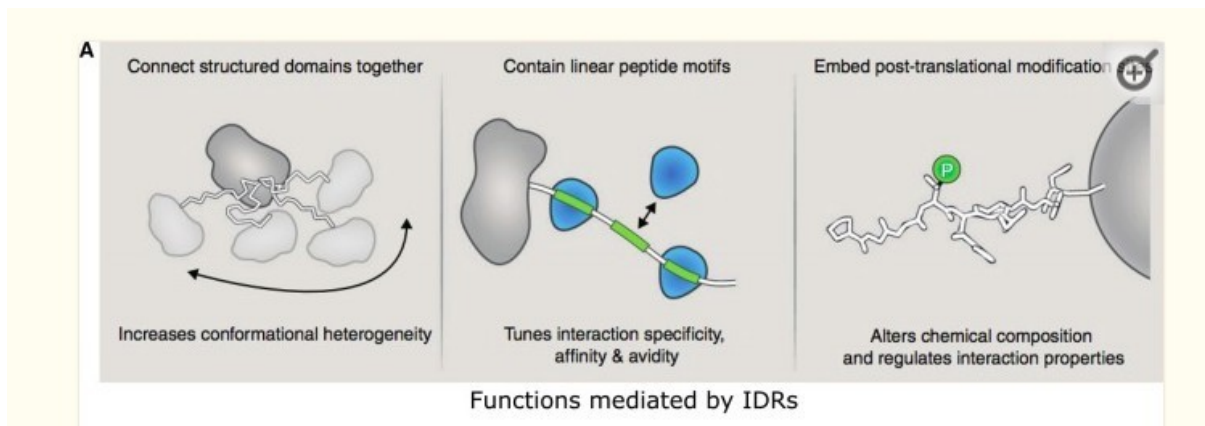
Следећа слика приказује горе наведене парадигме и одговарајући изглед протеина:



Слика 1 У делу под А приказана је парадигма секвенца-структура-функција, док је на слици В приказана парадигма секвенца-неуређеност-функција

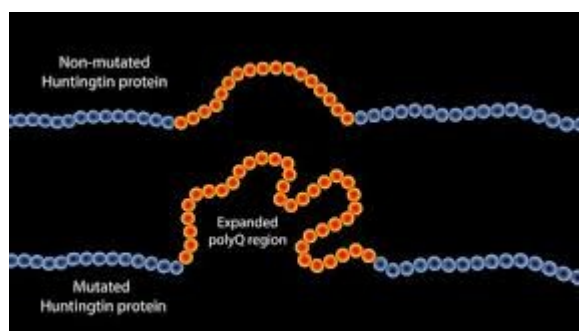
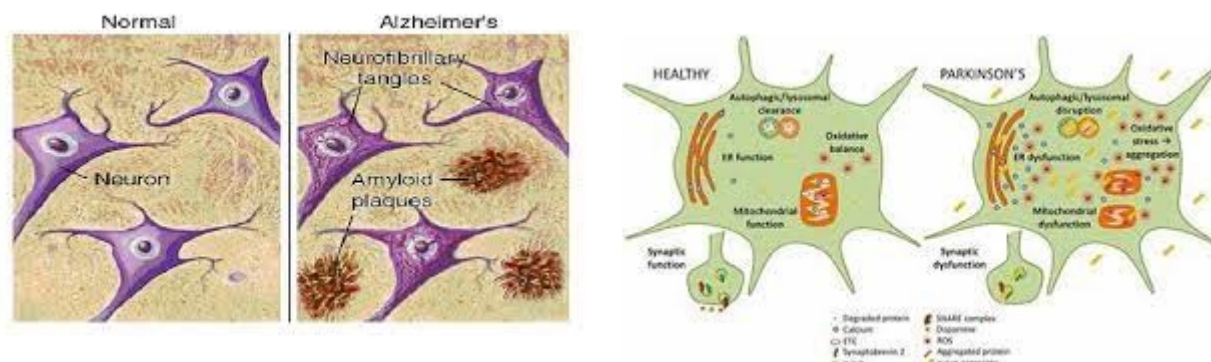
Неуређени региони (енг. Intrinsically Disordered Region – IDR) могу донети неке предности протеинима (Слика 2):

- Повезивање структурираних домена. На овај начин дозвољава се интеракција истог протеина са више различитих "комуникационих партнера" истовремено на веома функционалан начин.
- Олакшавање регулације функције протеина путем различитих посттранслационих модификација (ПТМ) остатака унутар IDR-а. Захваљујући својој конформационој флексибилности, неуређени региони служе као одлични супстрати за кодирање и декодирање информација путем посттранслационих модификација.
- Усвајање различитих конформација приликом везивања за различите партнере у интеракцији. Ова својства IDR-а чине их погодним за обављање сигналних и регулаторних функција. Заиста, анализе функција протеина са IDR на нивоу генома су откриле да су они обogaћени сигналним протеинима и протеинима који везују нуклеинску киселину као што су киназе, фактори транскрипције и фактори спајања.



Слика 2. На слици се могу видети интеракције протеина са више партнера истовремено (прве 2 сличице), као и регулација функције протеина (последња у низу сличица)

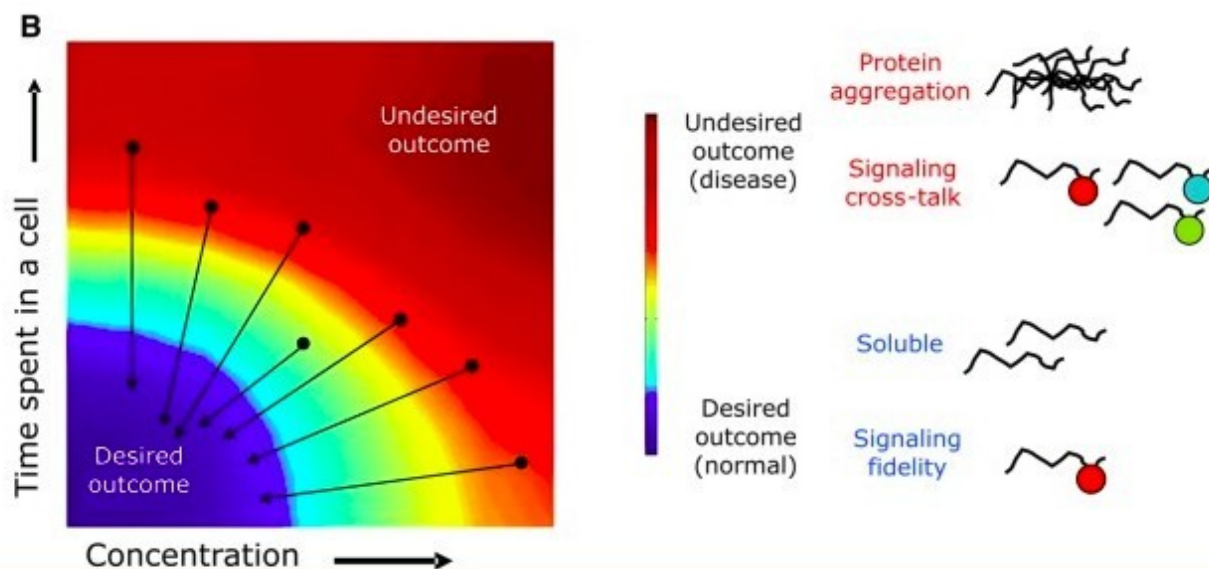
Међутим, док су многе студије показале како протеини са неуређеним регионима могу допринети повећању функционалне свестраности и ћелијској сложености, истраживања у последњих петнаестак година такође су открила важну улогу ових региона у многим људским болестима. Мутације које доводе до промене нивоа протеина са IDR могу довести до агрегације протеина, што доводи до болести као што је неуродегенерација, те не треба да нас изненађује што је објављено да се агрегати неуређених протеина налазе у веома високим концентрацијама у можданим наслагама пацијената са неуродегенеративним обољењима. Слично, мутације унутар IDR-а које повећавају склоност агрегацији, као што су оне које се виде у амилоид β -пептиду, α -синуклеину и Хантингтину, директно су повезане са болестима као што су Алцхајмерова, Паркинсонова и Хантингтонова болест, респективно.



Слика 3. Поређење нормалних протеина и њихових мутација. На сликама су редом приказане мутације које доводе до Алцхајмерове, Паркинсонове (слике горе), те Хантингтонове болести (доња слика)

Показало се да су неуређени региони обогаћени генима који учествују у ћелијској сигнализацији и протеинима повезаним са раком, као што су онкогени или гени супресори тумора. Као што је горе поменуто, овакви региони могу посредовати у интеракцијама ниског афинитета, па обиље протеина који поседују ове особине може да формира непожељне интеракције и да веже друге протеине у непродуктивне комплексе. На овај начин, они могу пореметити равнотежу у многим сигналним и регулаторним мрежама, што доводи до болести као што је рак.

Да би објаснили како су корисне и потенцијално штетне улоге протеина са IDR-ом једнако вероватне у ћелији, научници су истражили доступност таквих протеина унутар ње, како у смислу времена проведеног у ћелији, тако и у стабилном стању неуређених протеина и њихових транскрипата у многим организмима, од квасца до људи (Слика 4). Уочено је да су протеини са IDR-ом строже регулисани од оних са структурираним доменима у више фаза експресије гена, у распону од синтезе транскрипта до деградације протеина. На овај начин, количина и време присуства протеина са неуређеним регионима у ћелији су строго ограничени. Све док су ове мере у границама, исходи су позитивни, нпр. интеракције су поуздане, а протеини растворљиви. Међутим, ако се њихов животни век или количина у ћелији значајно промени, то може довести до нежељених исхода као што је накупљање протеина или унакрсно слање сигнала због нефункционалних интеракција. На основу ових резултата, научници су изнели претпоставку да је унутар ћелије координирана строга регулација протеина са неуређеним регионима у неколико фаза транскрипције и превођења, што осигурава да су они присутни кратко време и у малим количинама. Ова стратегија минимизује штетне ефекте неуређених протеина и истовремено дозвољава њихов витални допринос функционисању ћелије. Важна импликација овог запажања је да, поред мутација које утичу на протеине са неуређеним регионима и узрокују болест, мутације које утичу на гене који регулишу доступност ових протеина такође могу бити важна класа гена болести које би требало пажљиво истражити.



Слика 4 График који показује пожељни (плава боја у доњем левом углу) и непожељни (црвена боја у горњем десном углу) исход када је регулација протеина у питању

1.2 Софтвер који се користи

За потребе овог рада били су коришћени следећи алати:

- програм StatRepeats
- Jupyter notebook
- програмски језик Python
- програмски језик Java
- програмски језик R

1.2.1 StatRepeats

StatRepeats је један од програма који се налази у оквиру пакета RepeatsPlus. Овај пакет је скуп програма који обезбеђује статистичко филтрирање према дужини улазног низа и броју понављања, филтрирање маске мотива, као и велики број других опција и намењен је искључиво за академску употребу.

1.2.2 Jupyter notebook

Jupyter notebook је веб апликација за креирање и дељење рачунарских докумената. Нуди једноставно, модернизовано искуство усредсређено на документе.

1.2.3 Програмски језик Python

Python је језик опште намене, свестран и моћан, те је стога идеалан за рад јер је код писан у овом програмском језику концизан и лак за читање. Скоро сваки проблем на који су програмери током година рада наилазили, овај језик може да реши: од веб развоја преко машинског учења до истраживања података, па је управо из ових разлога Python главни избор за рад на овом пројекту.

1.2.4 Програмски језик Java

Java је програмски језик опште намене, конкурентан, строго типизиран, објектно оријентисани језик заснован на класама. Обично се компајлира у бајткод скуп инструкција и бинарни формат дефинисан у спецификацији Java виртуелне машине. Веома је лак за рад са фајловима, те ће у ту сврху и бити највише коришћен за израду рада.

1.2.5 Програмски језик R

R је софтверско окружење корисно за статистичко рачунарство и графику. Међутим, овај програмски језик има и пакете који се могу користити за истраживање података, као на пример пакет који налази правила придруживања, и који ће бити коришћен за израду овог рада.

1.3 Скуп података за рад

Скуп података који је коришћен за израду овог рада може се наћи у бази неуређених протеина DisProt (линк се може наћи у литератури). Приликом посете бази, могуће је одабрати жељену верзију података за преузимање, врсте протеина, као и које жељене аспекте. Након извршеног одабира, потребно је назначити и тип података за преузимање, као и формат. За потребе овог рада преузета је актуелна верзија (у време израде то је 2022_12), док су за скупове протеина и аспекте одабране све могућности. Подаци су преузети у JSON формату.

DisProt

Browse Ontology Release notes Download Help About Biocuration

Download

Release: 2022_12 (Current) Dataset: All Aspect: All

Type of the data: ☒ Regions ☐ Consensus Include: ☒ Ambiguous ☐ Obsolete

Select the format:

IDPontology

Ontology: 0.3.0 (Current)

The IDPO team...

Слика 5 Подешавања

Све информације које се налазе у овој бази података су проверене, тако да нема потребе за претпроцесирањем, већ је могуће одмах радити са готовим подацима. У самој бази налазе се информације за 2470 протеина. Неке од карактеристика протеина које се налазе у JSON фајловима су:

- фамилија протеина - носи информације о идентификатору фамилије протеина, имену, почетку и крају неуређеног региона
- секвенца протеина у облику ниске карактера,
- таксономија – у којој врсти је пронађен дати узорак,
- идентификатор у оквиру базе неуређених секвенци,
- идентификатор таксономије,
- број неуређених региона који се налазе у датом протеину, као и информације о самим регионима
- региони садрже информације о стању протеина, онтологији, крају и идентификатору региона, референце (текст са информацијама) и идентификатор термина
- неке занимљиве статистике као што је проценат протеина који садржи неуређене регионе.

2. Претпроцесирање података и налажење директних и инверзних некомплементарних секвенци

Да би за ове податке било могуће покренути програм StatRepeats, потребно их је прво пребацити у .fasta формат јер програм ради искључиво са овим форматом (у биоинформатици и биохемији, fasta формат је текстуални формат за представљање или нуклеотидних секвенци или аминокиселинских (протеинских) секвенци, у којима су нуклеотиди или аминокиселине представљени помоћу једнословних кодова. Састоји се од заглавља, које почиње знаком > којег прати идентификатор и даље опис протеина/ДНК секвенце и тела у коме се налази сама секвенца). За добијање одговарајућег .fasta фајла довољно је покренути програм toFasta.ipynb који ће обрадити прослеђени JSON фајл (назван podaci.json) и на основу њега направити датотеку preprocessed.fasta коју сада можемо проследити програму. Поменути програм користи библиотеку Bio, која је део колекција некомерцијалних Пајтон алата за рачунарску биологију и биоинформатику и садржи класе које представљају биолошке секвенце и а anotације секвенци. Библиотека такође може да чита и пише у различите формате датотека, па је за инсталацију те библиотеке потребно у терминалу откуцати наредбу `pip install Bio`, или, уколико покрећете програм из Јупитера, `!pip install Bio`. Добијени .fasta фајл садржи 26.488 линија, а део његовог садржаја може се видети на Слици 6.

```

home > andjela > Desktop > ipSeminarski > preprocessed.fasta
1  >DP000003 Human adenovirus C serotype 5
2  MASREEEQRETTPERGGAARRPPTMEDVSSPSPSPPPPRAPPKRMRRRIESEDEEDSS
3  QDALVPRTPSPRPSTSAADLAIPKKKKRPSKPERPPSPPEVIDSEEEEDVALQMVG
4  FSNPPVLKHKGGKRTVRRLNEDDPVARGMRTQEEEEEPSEAESEITVMNPLSVPIVSA
5  WEKGMEAAARALMDKYHVDNDLKANFKLLPDQVEALAAVCKTWLNEEHRGLQLTFTSKTFF
6  VTMMGRFLQAYLOSFAEVTYKHNEPTGCALWLHRCAEIEGELKCLHGSIMINKHEVIEMD
7  VTSENGQRALKEQSSKAKIVKNRWGRNVQISNTDARCCVHDAACPANQFSGKSCGMFFS
8  EGAKAQVAFKQIKAFMQALYPNAQTGHGHLMLPLRCECNSKPGHAPFLGRQLPKLTPFAL
9  SNAEDLDADLISDKSVLASVHHPALIVFOCCNPVYRNSRAQGGGPNCDFKISAPDLLNAL
10 VMVRLSWSENFTELPRMVVPEFKWSTKHQYRNVSLPAHSDARQNPFD
11 >DP000004 Homo sapiens
12 MKTQRDGHSLGRWSLVLLGLVMPALIAIAQVLSYKEAVLRAIDGINQRSSDANLYRLLD
13 LDRPTMDGDPDTPKPVSTVKETVCPRTTQSPEDCDFKKDGLVKRCMGTVTLNQARGS
14 FDISCDKNRKFALLGDFFRKSKEIKGKEFKRIVQRIKDFLRNLVPRTES
15 >DP000005 Escherichia phage lambda
16 MDAQTRRRERRAEQAQWKAANPLLVGVSAPVNRPILSLNRKPKSRVESALNPDLTVL
17 AEYHKQIESNLQRIERKNQRTWYSKPGERGITCSGRQKIKGKSIPLI
18 >DP000006 Equus caballus
19 MGDVEKGKKIFVQKCAQCHTVEKGGKHKTPNLHGLFGRKTGQAPGFTYTDANKNKGITW
20 KEETLMEYLENPKKYIPGKMFAGIKKTEREDLIAYLKKATNE
21 >DP000007 Homo sapiens
22 MPKRGGKGAEDGDELRTPEAKKSKTAAKKNDEAAGEGALYEDPPDQKTSPPSGKPA
23 TLKICSWNVVGLRAWIKKGLDWKKEAPDILCLQETKCSENKLPALQELPGLSHQYWS
24 APSDEKESVGVLLSRQCPLKVSYGIGDEEHDQEGRVIVAEFDSFVLVTAYVNPAGRLV
25 RLEYRQRWDEAFKFLKGLASRKPLVLCGDLNVAHEEIDLNPKNKKNAGFTPQERQGF
26 GELLQAVPLADSFRLHYLPNTYAYTFWTYMMNARSKNVGWRDLYFLLSHSLPALCDSKI
27 RSKALGSDHCPITLYLAL
28 >DP000008 Mus musculus
29 MLWQKSTAPEQAPAPRPYQGVVRKEPVKELLRRKRGHTSVGAAGPPTAVVLPHQPLATY
30 STVGPSCLDMEVSASTVTEETLCAGWLSQAPATLQPLAPWTPYTEYVSHEAVSCPYST
31 DMVYQVPCSYTVVGPSSVLTYASPLITNVTPTATPAVGPOLEGPEHQAPLTYFPWP
32 DBI STI DTSSI QYDDBDPTI SGBDEVAI DTSTDEPDI NMMDDBRRTSSI TTDKI L I FEE

```

Слика 6 Изглед добијеног фајла

Сада, када су обезбеђени подаци у оном формату који нам одговара, можемо да покренемо програм StatRepeats који ће нам пронаћи директне и инверзне некомплементарне секвенце. Пошто нема смисла узимати секвенце дужине 1, то овде неће бити рађено. За потребе овог рада, програм је покретан више пута, за различите минималне дужине секвенци (ради добијања увида у обим добијених података), па је због тога написан скрипт extractAllInfo.py чија је улога да покреће програм за различите улазне вредности, а који је потребно покренути у терминалу (пре тога је неопходно позиционирати се у терминалу у онај фолдер који садржи извршну верзију StatRepeats програма. Потребно је и добијени фајл преместити у поменути фолдер). Свака наредба која се налази у горепоменутом скрипту позива програм StatRepeats за своје параметре и као резултат извршавања програма креира фајл са називом (in)directNall.out (где N означава минималну дужину секвенце за коју је програм покренут, а ознака (in)direct говори да ли је програм покренут за директне или инверзне некомплементарне секвенце). Сваки новогенерисани фајл садржи следеће информације:

- ако у протеину не постоји секвенца дужа од минималне дужине, исписује само идентификатор датог протеина и информацију да су сва слова у секвенци из дозвољеног алфабета
- ако у протеину постоји/постоје секвенца/е дуже од минималне дужине, онда се, поред информација које се исписују и у случају да нема секвенци исписују и следећи подаци:

1. индекс почетка прве понављајуће секвенце,
2. индекс краја прве понављајуће секвенце,
3. индекс почетка друге понављајуће секвенце,
4. индекс краја друге понављајуће секвенце,
5. дужина поновљене секвенце, као и део секвенце који се понавља , и
6. укупан број понављајућих секвенци свих дужина које је програм нашао (Слика 7)

```

DP00069 Alphabet size is 20.
Input sequence has 0 letters that are not in expected alphabet.

DP00070 Alphabet size is 20.
Input sequence has 0 letters that are not in expected alphabet.

DP00071 Alphabet size is 20.
Input sequence has 0 letters that are not in expected alphabet.
DP00071,287,295,287,295,9,AQQQAQQQA,AQQQAQQQA
DP00071,234,242,234,242,9,GGGGGGGGG,GGGGGGGGG
DP00071,235,242,235,242,8,GGGGGGGGG,GGGGGGGGG
DP00071,234,241,234,241,8,GGGGGGGGG,GGGGGGGGG

Total for length 8 is 2.
Total for length 9 is 2.

DP00072 Alphabet size is 20.
Input sequence has 0 letters that are not in expected alphabet.
DP00072,10670,10677,10670,10677,8,EEYEEYEE,EEYEEYEE
DP00072,10254,10264,10254,10264,11,EKKPVPVPKKE,EKKPVPVPKKE
DP00072,5554,5562,5554,5562,9,ISVTDTVSI,ISVTDTVSI
DP00072,10255,10264,10888,10897,10,EKKPVPVPKKE,EKKPVPVPKK
DP00072,10889,10897,10889,10897,9,KKPVPVPKK,KKPVPVPKK
DP00072,10724,10731,10724,10731,8,PVPEEPVP,PVPEEPVP
DP00072,32641,32649,32641,32649,9,TVHPEPHVT,TVHPEPHVT
DP00072,21399,21406,21399,21406,8,VPGPPGPV,VPGPPGPV

```

Слика 7 На слици видимо део генерисаног садржаја. Из њега можемо видети да протеини са идентификаторима DP00069 и DP00070 немају понављајућу секвенцу дужу од оне којом је покренут програм, док DP00071 и DP00072 имају. Секвенце које задовољавају тај услов су затим наведене, као и њихова статистика, тј. број појављивања секвенце те дужине у протеину

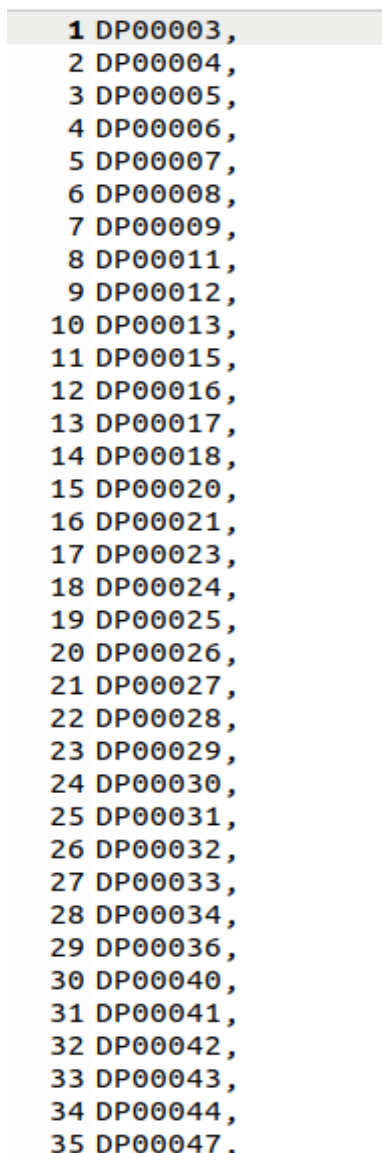
За каснију анализу могу нам бити корисне информације о броју секвенци које испуњавају услов, па преглед нудимо у оквиру табеле на Слици 8. Резултати су добијени покретањем програма countLines.py. Како је првих 7 линија сваког фајла посвећено информацијама, од добијеног броја линија одузет је овај број да би се добио број секвенци.

| Minimal sequence length | Number of direct sequences | Number of indirect sequences |
|-------------------------|----------------------------|------------------------------|
| 2 | 4579796 | 4703195 |
| 3 | 424599 | 414427 |
| 8 | 14872 | 12983 |
| 10 | 12712 | 11525 |
| 17 | 11118 | 10454 |
| 20 | 10854 | 10311 |
| 50 | 10194 | 9962 |
| 100 | 10006 | 9890 |

Табела 1 - приказује број директних и инверзних секвенци у зависности од минималне дужине секвенце

Међутим, иако овај фајл садржи све што нам је потребно за даљи рад, за даљу обраду било би доста једноставније када бисмо ове податке ипак чували у различитим датотекама. Да бисмо ово успели, довољно је покренути skript.py из терминала (и сада је потребно позиционирање у онај директоријум где се налази извршни StatRepeats програм). Једина разлика између овог и extractAllInfo.py програма је тај што је овај скрипт покренут са опцијом -load (in)directN која функционише тако што аутоматски генерише 3 фајла:

1. (in)directN.fasta.id у коме се у свакој линији налази по један идентификатор протеина из базе (пошто је програм покретан увек над истим подацима, сви генерисани .id фајлови имаће исти садржај- Слика 9)
2. (in)directN.fasta.load који садржи информације о (ин)директним секвенцама и то:
 - идентификатор протеина,
 - индекс почетка прве понављајуће секвенце,
 - индекс краја прве понављајуће секвенце,
 - индекс почетка друге понављајуће секвенце,
 - индекс краја друге понављајуће секвенце,
 - дужина поновљене секвенце, као и део секвенце који се понавља , и
 - укупан број понављајућих секвенци свих дужина које је програм нашао -Слика 10
3. (in)directN.fasta.stat у коме се, као што име и сугерише, налазе статистички подаци о сваком протеину из базе садржи податке о укупном броју секвенци одређене дужине, при чему се тај број креће од минимума који је задат па све до најдуже секвенце која се поклапа (Слика 11).



```
1 DP00003,  
2 DP00004,  
3 DP00005,  
4 DP00006,  
5 DP00007,  
6 DP00008,  
7 DP00009,  
8 DP00011,  
9 DP00012,  
10 DP00013,  
11 DP00015,  
12 DP00016,  
13 DP00017,  
14 DP00018,  
15 DP00020,  
16 DP00021,  
17 DP00023,  
18 DP00024,  
19 DP00025,  
20 DP00026,  
21 DP00027,  
22 DP00028,  
23 DP00029,  
24 DP00030,  
25 DP00031,  
26 DP00032,  
27 DP00033,  
28 DP00034,  
29 DP00036,  
30 DP00040,  
31 DP00041,  
32 DP00042,  
33 DP00043,  
34 DP00044,  
35 DP00047,
```

Слика 9 Изглед генерисаног .id фајла за директне секвенце дуже од 17

```
1 DP00017,197,214,199,216,18,APAPAPAPAPAPAPAP,APAPAPAPAPAPAPAP
2 DP00025,879,915,893,929,37,PTPTPEVPSEPETPTPTPEVPSEPETPTPTPEVP,PTPTPEVPSEPETPTPTPEVPSEPETPTPTPEVP
3 DP00025,879,901,907,929,23,PTPTPEVPSEPETPTPTPEVP,PTPTPEVPSEPETPTPTPEVP
4 DP00034,86,104,248,266,19,EGGSEGGGSEGGGSEGGG,EGGSEGGGSEGGGSEGGG
5 DP00034,244,262,249,267,19,GGGSEGGGSEGGGSEGGG,GGGSEGGGSEGGGSEGGG
6 DP00034,87,104,244,261,18,GGGSEGGGSEGGGSEGGG,GGGSEGGGSEGGGSEGGG
7 DP00065,1169,1253,1171,1255,85,DSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDS,DSDSDSDSDSDSDSDSDSDSDSDSDS
8 DP00065,1169,1251,1173,1255,83,DSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDS,DSDSDSDSDSDSDSDSDSDSDSDSDS
9 DP00065,1169,1249,1175,1255,81,DSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDS,DSDSDSDSDSDSDSDSDSDSDSDSDS
10 DP00065,1169,1247,1177,1255,79,DSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDS,DSDSDSDSDSDSDSDSDSDSDSDSDS
11 DP00065,1169,1245,1179,1255,77,DSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDS,DSDSDSDSDSDSDSDSDSDSDSDSDS
12 DP00065,1169,1243,1181,1255,75,DSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDS,DSDSDSDSDSDSDSDSDSDSDSDSDS
13 DP00065,1169,1241,1183,1255,73,DSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDS,DSDSDSDSDSDSDSDSDSDSDSDSDS
14 DP00065,1169,1239,1185,1255,71,DSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDS,DSDSDSDSDSDSDSDSDSDSDSDSDS
15 DP00065,1169,1237,1187,1255,69,DSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDS,DSDSDSDSDSDSDSDSDSDSDSDSDS
16 DP00065,1169,1235,1189,1255,67,DSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDS,DSDSDSDSDSDSDSDSDSDSDSDSDS
17 DP00065,1169,1233,1191,1255,65,DSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDS,DSDSDSDSDSDSDSDSDSDSDSDSDS
18 DP00065,1169,1231,1193,1255,63,DSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDS,DSDSDSDSDSDSDSDSDSDSDSDSDS
19 DP00065,1169,1229,1195,1255,61,DSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDS,DSDSDSDSDSDSDSDSDSDSDSDSDS
20 DP00065,1169,1227,1197,1255,59,DSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDS,DSDSDSDSDSDSDSDSDSDSDSDSDS
21 DP00065,1169,1225,1199,1255,57,DSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDS,DSDSDSDSDSDSDSDSDSDSDSDSDS
22 DP00065,1169,1223,1201,1255,55,DSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDS,DSDSDSDSDSDSDSDSDSDSDSDSDS
23 DP00065,1169,1221,1203,1255,53,DSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDS,DSDSDSDSDSDSDSDSDSDSDSDSDS
24 DP00065,1169,1219,1205,1255,51,DSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDS,DSDSDSDSDSDSDSDSDSDSDSDSDS
25 DP00065,1169,1217,1207,1255,49,DSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDS,DSDSDSDSDSDSDSDSDSDSDSDSDS
26 DP00065,1169,1215,1209,1255,47,DSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDS,DSDSDSDSDSDSDSDSDSDSDSDSDS
27 DP00065,1169,1213,1211,1255,45,DSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDS,DSDSDSDSDSDSDSDSDSDSDSDSDS
28 DP00065,1169,1211,1213,1255,43,DSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDS,DSDSDSDSDSDSDSDSDSDSDSDSDS
29 DP00065,1169,1209,1215,1255,41,DSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDS,DSDSDSDSDSDSDSDSDSDSDSDSDS
30 DP00065,1169,1207,1217,1255,39,DSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDS,DSDSDSDSDSDSDSDSDSDSDSDSDS
31 DP00065,1169,1205,1219,1255,37,DSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDS,DSDSDSDSDSDSDSDSDSDSDSDSDS
32 DP00065,1169,1203,1221,1255,35,DSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDS,DSDSDSDSDSDSDSDSDSDSDSDSDS
33 DP00065,1169,1201,1223,1255,33,DSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDS,DSDSDSDSDSDSDSDSDSDSDSDSDS
34 DP00065,1169,1199,1225,1255,31,DSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDSDS,DSDSDSDSDSDSDSDSDSDSDSDSDS
```

Слика 10 Изглед генерисаног .load фајла за директне секвенце дуже од 17

```
Total for length 17 is 5.
Total for length 19 is 4.
Total for length 21 is 4.
Total for length 23 is 4.
Total for length 25 is 4.
Total for length 27 is 4.
Total for length 29 is 30.
Total for length 31 is 1.
Total for length 33 is 1.
Total for length 35 is 1.
Total for length 37 is 1.
Total for length 39 is 1.
Total for length 41 is 1.
Total for length 43 is 1.
Total for length 44 is 1.
Total for length 45 is 1.
Total for length 47 is 1.
Total for length 49 is 1.
Total for length 51 is 1.
Total for length 53 is 1.
Total for length 55 is 1.
Total for length 57 is 1.
Total for length 59 is 1.
Total for length 61 is 1.
Total for length 63 is 1.
Total for length 65 is 1.
Total for length 67 is 1.
Total for length 69 is 1.
Total for length 71 is 1.
Total for length 73 is 1.
Total for length 75 is 1.
Total for length 77 is 1.
Total for length 79 is 1.
```

Слика 11 Изглед генерисаног .stat фајла за директне секвенце дуже од 17

Треба напоменути да је и овде, као и у претходном случају, N минимална дужина секвенце, а (in)direct у називу фајла индикатор да ли се ради о фајлу који садржи директне или инверзне секвенце.

Напомена: може се чинити да је овакав приступ непрактичан јер се издвајају дупле информације, међутим аутор овакав избор оправдава следећим чињеницама: када је обрада неуређених секвенци у питању, довољне су нам информације које се налазе у .load фајловима, међутим, уколико желимо да обрађујемо статистику везану за сваки протеин, то не можемо лако урадити из .stat фајла јер он не садржи идентификатор протеина за који су информације везане, док фајл који садржи свеобухватне информације то све садржи.

3. Налажење позиција неуређених региона и тражење додатних карактеристика процесирањем текста

3.1 Налажење позиција неуређених региона

Као што је назначено у уводном делу овог рада, део информација које је потребно обрадити обухвата информације о неуређеним регионима у оквиру протеина. Сваки протеин може имати један или више неуређених региона. Како су нам информације о почецима и крајевима неуређених региона од кључне важности, у овом поглављу бавићемо се управо издвајањем ових региона из датих протеина. Информације које ћемо издвојити из сваког протеина су следеће:

- позиција почетка неуређеног региона
- позиција краја неуређеног региона
- идентификатор региона и
- део секвенце који је обухваћен овим регионом.

Да бисмо ово урадили потребно је покренути програм `extractRegions.ipynb` који пролази кроз преузети JSON фајл, извлачи информације од интереса и уписује их у фајл `regioni.txt`. Како на сајту базе DisProt постоји могућност да се преузме и .fasta фајл који садржи информације о свим неуређеним регионима, и он је преузет са циљем да се провере

резултати написаног програма. За проверу кардиналности региона у .fasta фајлу, написан је програм checkCount.java, док се кардиналност скупа региона који смо ми добили може видети у поменутом extractRegions.ipynb фајлу, а може се погледати и број линија region1.txt фајла. У свим наведеним случајевима кардиналност скупова је иста - 10.544 региона. Изглед добијеног фајла приложен је на Слици 12.

```
DP00003r002 294-334 EHVIEMDVTSNGORALKEQSSKAKIVKNRWGRNVQISNT
DP00003r004 454-464 VYRNSRAQGGG
DP00004r001 134-170 LLGDFFRKSKEKIGKEFKRIVQRIKDFLRNLVPRTE
DP00004r002 134-170 LLGDFFRKSKEKIGKEFKRIVQRIKDFLRNLVPRTE
DP00004r004 134-170 LLGDFFRKSKEKIGKEFKRIVQRIKDFLRNLVPRTE
DP00004r005 150-162 FKRIQRIKDFLR
DP00004r006 150-162 FKRIQRIKDFLR
DP00005r001 1-107 MDAQTRRRERRAEKQAQWKAANPLLVGVSAPVNRPIILSLNRKPKSRVESALNPIDLTVLAEYHKQIESNLQRIERKNQRTWYSPGERGITCSGRQKIKGKSJ
DP00005r004 1-107 MDAQTRRRERRAEKQAQWKAANPLLVGVSAPVNRPIILSLNRKPKSRVESALNPIDLTVLAEYHKQIESNLQRIERKNQRTWYSPGERGITCSGRQKIKGKSJ
DP00005r005 1-107 MDAQTRRRERRAEKQAQWKAANPLLVGVSAPVNRPIILSLNRKPKSRVESALNPIDLTVLAEYHKQIESNLQRIERKNQRTWYSPGERGITCSGRQKIKGKSJ
DP00005r006 1-107 MDAQTRRRERRAEKQAQWKAANPLLVGVSAPVNRPIILSLNRKPKSRVESALNPIDLTVLAEYHKQIESNLQRIERKNQRTWYSPGERGITCSGRQKIKGKSJ
DP00005r007 1-107 MDAQTRRRERRAEKQAQWKAANPLLVGVSAPVNRPIILSLNRKPKSRVESALNPIDLTVLAEYHKQIESNLQRIERKNQRTWYSPGERGITCSGRQKIKGKSJ
DP00005r008 1-107 MDAQTRRRERRAEKQAQWKAANPLLVGVSAPVNRPIILSLNRKPKSRVESALNPIDLTVLAEYHKQIESNLQRIERKNQRTWYSPGERGITCSGRQKIKGKSJ
DP00005r009 1-107 MDAQTRRRERRAEKQAQWKAANPLLVGVSAPVNRPIILSLNRKPKSRVESALNPIDLTVLAEYHKQIESNLQRIERKNQRTWYSPGERGITCSGRQKIKGKSJ
DP00005r010 1-107 MDAQTRRRERRAEKQAQWKAANPLLVGVSAPVNRPIILSLNRKPKSRVESALNPIDLTVLAEYHKQIESNLQRIERKNQRTWYSPGERGITCSGRQKIKGKSJ
DP00005r011 1-107 MDAQTRRRERRAEKQAQWKAANPLLVGVSAPVNRPIILSLNRKPKSRVESALNPIDLTVLAEYHKQIESNLQRIERKNQRTWYSPGERGITCSGRQKIKGKSJ
DP00005r012 1-22 MDAQTRRRERRAEKQAQWKAAN
DP00005r013 34-47 NRPILSLNRKPKSR
DP00005r014 34-47 NRPILSLNRKPKSR
DP00005r015 1-22 MDAQTRRRERRAEKQAQWKAAN
DP00005r016 1-22 MDAQTRRRERRAEKQAQWKAAN
DP00005r017 1-107 MDAQTRRRERRAEKQAQWKAANPLLVGVSAPVNRPIILSLNRKPKSRVESALNPIDLTVLAEYHKQIESNLQRIERKNQRTWYSPGERGITCSGRQKIKGKSJ
DP00005r018 1-36 MDAQTRRRERRAEKQAQWKAANPLLVGVSAPVNRPI
DP00006r011 1-104 MGDVEKGKKIFVQKCAQCHTVEKGGKHKTGPNLHGLFGRKTGQAPGFTYTDANKNKGITWKEETLMEYLENPKKYIPGTMIFAGIKKKTEREDLIAYLKKATN
DP00006r012 1-104 MGDVEKGKKIFVQKCAQCHTVEKGGKHKTGPNLHGLFGRKTGQAPGFTYTDANKNKGITWKEETLMEYLENPKKYIPGTMIFAGIKKKTEREDLIAYLKKATN
DP00006r013 2-105 GDVEKGKKIFVQKCAQCHTVEKGGKHKTGPNLHGLFGRKTGQAPGFTYTDANKNKGITWKEETLMEYLENPKKYIPGTMIFAGIKKKTEREDLIAYLKKATNE
DP00006r014 2-105 GDVEKGKKIFVQKCAQCHTVEKGGKHKTGPNLHGLFGRKTGQAPGFTYTDANKNKGITWKEETLMEYLENPKKYIPGTMIFAGIKKKTEREDLIAYLKKATNE
DP00007r002 1-42 MPKRGGKGAVAEDGDELRTPEAKKSKTAACKNDKEAAGEGP
DP00007r006 1-36 MPKRGGKGAVAEDGDELRTPEAKKSKTAACKNDKEAAGEGP
DP00007r007 32-43 KNDKEAAGEGPA
DP00007r008 2-40 PKRGKKGGAVAEDGDELRTPEAKKSKTAACKNDKEAAGEGP
```

Слика 12 Део извучених неуређених региона

Напомена: када бисмо посматрали само позиције почетка и краја неуређених региона, имали бисмо привид да постоје дупликати којих се треба отарасити. Међутим, како сви ови региони имају свој јединствен идентификатор, сви добијени резултати су задржани.

3.2 Налажење додатних карактеристика претпроцесирањем текста

Сада бисмо желели да нађемо потенцијалне карактеристике које могу бити од значаја за наше резултате. Пошто није унапред познато које ће особине дати резултате, у овом делу биће издвојене неке карактеристике које се чине значајно. Наравно, да ли су оне заиста

значајне или не, показате нам тек анализа издвојених правила придруживања. Неке од могућих значајних карактеристика су:

- таксономија врсте код којих постоји дати протеин
- индикатор пресека неуређеног региона са понављајућом секвенцом
- индикатор да ли неуређени регион садржи понављајућу секвенцу
- ниво неуређености протеина (по препоруци ментора, остављене су оригиналне вредности које припадају скупу $[0,1]$)
- текст који описује дати протеин

Пошто су текстуални подаци о протеинима веома обимни, овом приликом неће бити коришћене целокупне информације које се могу наћи у преузетом фајлу, већ ће бити издвојене кључне речи и оне ће бити посматране као један атрибут протеина. Да бисмо ово урадили неопходно је да прво извучемо поменуте текстуалне податке. То се може урадити покретањем програма `extractText.ipynb`. Жељене информације биће извучене у фајл `extractText.txt` да бисмо могли даље да га обрађујемо.

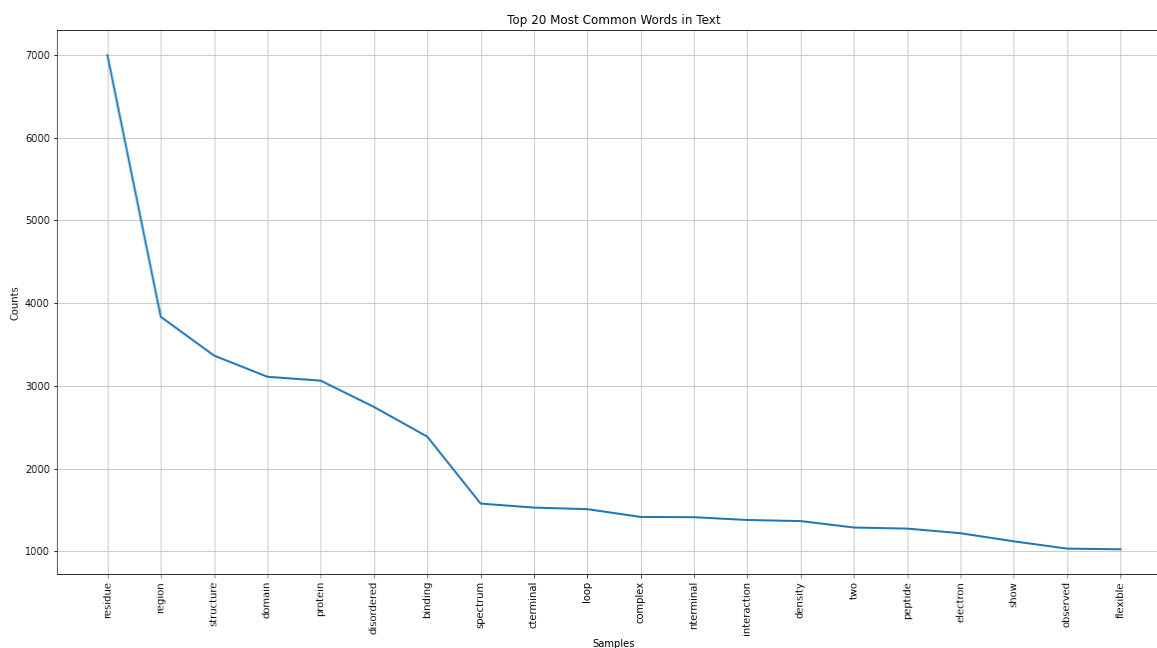
Сада, након успешног издвајања текстуалних целина, желимо да из добијеног текста извучемо само оне најзначајније речи. Међутим, услед обимности резултујућег документа, људском оку је немогуће да утврди које су то речи које се најчешће понављају. Зато ће овај задатак да одради програм уместо нас. Но, иако се летимичним погледом не може установити које речи су од значаја, ипак је могуће уочити које речи дефинитивно нису од значаја, па је први корак филтрирање речи тако да се из текста избаце сви знакови интерпункције, сви бројеви (као и све речи које садрже бројеве) и знаци за нови ред.

Погледом на резултат гореописаног процеса, видимо да смо постигли жељени ефекат. Али претпроцесирању и даље није крај јер у тексту и даље остају једнословне речи, грчка слова и стоп речи, које се ту не смеју наћи уколико желимо да извршимо анализу правилно. Стога је следећи корак управо даље пречишћавање документа од ових речи¹. Након новог одстрањивања, остаје нам текст из кога се напослетку могу извући кључне речи.

Пошто се у енглеском, као и у сваком језику, може десити да се једна реч нађе у више облика у тексту (нпр. иста именица се може наћи у једнини и у множини, глаголи се могу наћи у више времена...) не желимо да сваки облик речи посматрамо засебно већ желимо да посматрамо све облике речи које имају исти корен као једну (нпр. ако се у тексту јавља и

¹ Као скуп стоп речи коришћен је скуп стоп речи за енглески језик који се може наћи у оквиру библиотеке `nlTK.corpus.stopwords`. Међутим, како овај скуп не садржи све речи које ми желимо да одстранимо, додате су и нове стоп речи за наш конкретан документ

једнина и множина исте именице, желимо сва та појављивања да посматрамо као појављивање једне именице, а не две. Аналогно важи за глаголе). Испоставља се да пакет `nlTK` има решење и за тај проблем. Прво је потребно текст поделити на листу речи које га чине, а онда се речи из те листе лематизују, тј. налазе се корени сваке речи. Некада синтагме помажу да се стекне бољи увид у оно о чему се говори, те ћемо прво издвојити синтагме које се најчешће јављају. Биће издвојене најучесталије двочлане, трочлане и четворочлане синтагме. Након тога, издвајамо и најчешће употребљиване речи на основу добијених лематизованих речи. На следећој слици може се видети списак 20 највише понављаних речи, заједно са њиховом фреквенцијом.

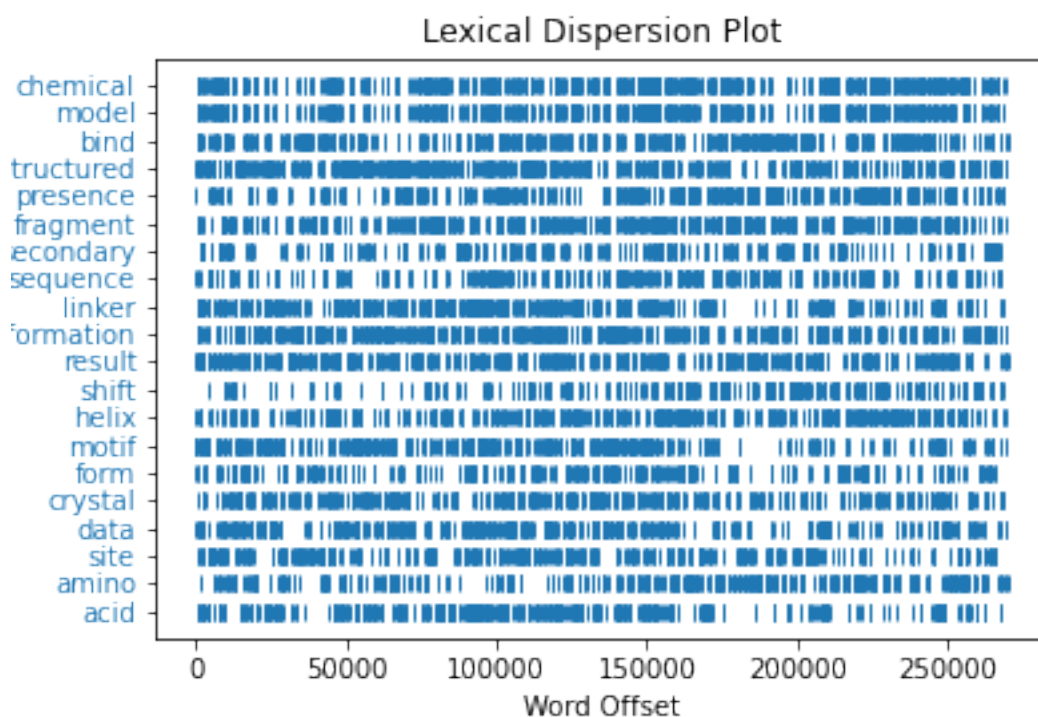


Слика 13 – график који показује фреквенцију појављивања 20 најчешћих речи у тексту. На у оси налазе се речи, док се на х оси налази њихова фрекванција. Као што можемо видети, све речи су веома честе

Уколико желимо да видимо и тачно када се која кључна реч појавила у тексту, добијени графици су експортирани у сликовном облику. На Слици 14 приказано је како изгледа граф лексичке дисперзије.

Поступак објашњен у последњих неколико пасуса је резултат покретања програма `extractKeywords.ipynb`, док се одговарајуће кључне речи налазе у документу `keywords.txt`.

Даље је, за потребе олакшавања посла који обухвата прављење документа који ће касније бити коришћен за налажење правила придруживања, направљена мапа која пресликава јединствени број идентификатора протеина из базе у скуп кључних речи који се уз њега појављују, те је дата мапа преписана у датотеку под називом keywordDictionary.txt. Одговарајући код се може наћи у saveDictionary.ipynb документу.



Слика 14 – један од графова који приказује где се и колико често налазе кључне речи у документу

3.3 Прављење фајла за налажење правила придруживања

Да бисмо могли да применимо алгоритме за налажење правила придруживања, морамо прво наше резултате објединити у један фајл, како би сви потребни подаци били на једном месту. Пошто података о поновцима дужине 2 и више има преко четири милиона (види Сliku 8), за потребе овог рада биће коришћени подаци о поновцима минималне дужине 3, којих има око 400000.

Међутим, како је и фајл са 400000 линија велики и захтева доста времена за обраду, приступ који је био коришћен током овог истраживања обухвата покретање неколико кориснички дефинисаних нити (класа `FileThreadRunnable.java`) у оквиру програма `splitFile.java` које ће поделити фајл на 8 мањих фајлова са приближно истим бројем линија. Након тога, када је подела фајла завршена, потребно је покренути скрипт `appendFiles.py` чија је улога да покрене више мини-програма који ће да врше анализу новодобијених фајлова. Сада коначно имамо документ над којим можемо да применимо алгоритме за правила придруживања.

Аналогно претходном поступку, на исти начин долазимо и до фајла за понављајуће секвенце. Прво је потребно покренути програм `splitIndirect.java` који ће поделити наш фајл на 8 мањих фајлова (што је опет урађено уз помоћ нити), које затим спајамо покретањем програма `appendFilesIndirect.py` који, као и у претходном случају, има задатак да позове више мини-програма који ће да врше конкатенацију резултата. Када је и тај посао обављен, имамо фајл који је спреман за даљу обраду.

3.4 Прављење фајлова који садрже трансакције

Пошто ће се за генерисање правила придруживања користити програмски језик R (јер је процес налажења правила придруживања у језику Python веома временски захтеван), да бисмо олакшали тај део посла од фајлова које смо припремили за обраду направимо нове фајлове који ће садржати само информације које ће нам бити значајне за издвајање правила. Трансакције издвајамо тако што издвојимо колоне почетног фајла које су нам значајне, при чему ћемо, све целобројне вредности које се јављају у индексима да заменимо стринговним вредностима (па ћемо, на пример, уместо вредности 0 имати вредност `IND_0`, а уместо 1 имаћемо `IND_1`) јер је таква репрезентација погоднија за налажење правила придруживања у R-у. На овај начин, извршавањем програма `generateTransactions.ipynb` и `generateTransactionsIndirect.ipynb` добијамо документе који садрже информације о поновцима, индексима пресека поновака са неуређеним регионима и евентуално неке додатне информације попут таксономије или списка кључних речи. Најпре се генеришу фајлови под називом `transactions.csv`, тј. `transactionsIndirect.csv` који садрже само поновке и индексе пресека директних, тј. инверзних понављајућих секвенци са неуређеним регионима

респективно. Даље, попуњавају се датотеке које садрже поновке, индексе пресека и таксономију врста у којој су они пронађени. Ове информације се смештају у фајлове `transactionsWithTaxonomy.csv` и `transactionsIndirectWithTaxonomy.csv`. На крају, генеришу се и документи који од информација садрже поновке, индексе пресека и листу кључних речи које се везују за дати протеин. Потоње датотеке су назване `transactionsWithKeywords.csv` и `transactionsIndirectWithKeywords.csv`.

На исти начин издвајамо информације од интереса за индекс који садржи информацију да ли неуређени регион садржи поновак или не. Програми који врше ову функцију названи су редом `generateTransactionsDirectContains.ipynb` и `generateTransactionsIndirectContains.ipynb`. Као и у претходном случају, прво су генерисани фајлови који садрже само поновак и одговарајући индекс и названи су `transactionsContains.csv` и `transactionsIndirectContains.csv`. Затим су у датотеке издвојени поновци, индекси садржања и информације о таксономији, које су, слично малопређашњем случају, назване `transactionsWithTaxonomyContains.csv` и `transactionsIndirectWithTaxonomyContains.csv`. Најзад, на једно место издвојени су поновак, индекс и листа кључних речи, обједињени у датотеке `transactionsWithKeywordsContains.csv` и `transactionsIndirectWithKeywordsContains.csv`.

4. Правила придруживања и априори алгоритам

4.1 Априори алгоритам

Априори алгоритам је један од најпознатијих и најутицајнијих алгоритама за проналажење правила придруживања. Он ради тако што проналази честе скупове, на основу којих се касније издвајају правила. Може бити веома користан за откривање веза између података за које се не би помислило на први поглед да су повезани. Још неке од позитивних страна овог алгоритма јесу његова једноставност за разумевање, интуитивно схватање добијених резултата и флексибилност. Са друге стране, алгоритам је веома временски и просторно комплексан, не ради са нумеричким подацима и лако може да нађе погрешна правила ако му је дат редак скуп података. Међутим, подаци за израду овог рада су обимни, па последња ставка није превелика брига.

Веома битни појмови повезани са априори алгоритмом су: *подршка*, *поузданост* и *лифт мера*. Подршка представља проценат учесталости појављивања леве и десне стране правила у односу на све трансакције. Поузданост представља однос броја појављивања леве и десне стране у пару у односу на број појављивања леве стране у трансакцијама. Обе ове мере узимају вредности из опсега $[0,1]$. Лифт мера правила придруживања представља однос поузданости правила и подршке десне стране правила. Ова мера се користи за утврђивање квалитета правила. Може узети било коју вредност из скупа $(-\infty, \infty)$. Вредности из интервала $[-1,1]$ нису посебно занимљиве јер оне говоре да је правило очекивано. Од већег су интереса вредности које су веће од 1 што говори да се последична страна правила (десна) много чешће јавља заједно са узрочном (левом) него без ње, или мање од -1, ако се узрочна и последична страна правила не јављају у пару онолико често колико је то очекивано.

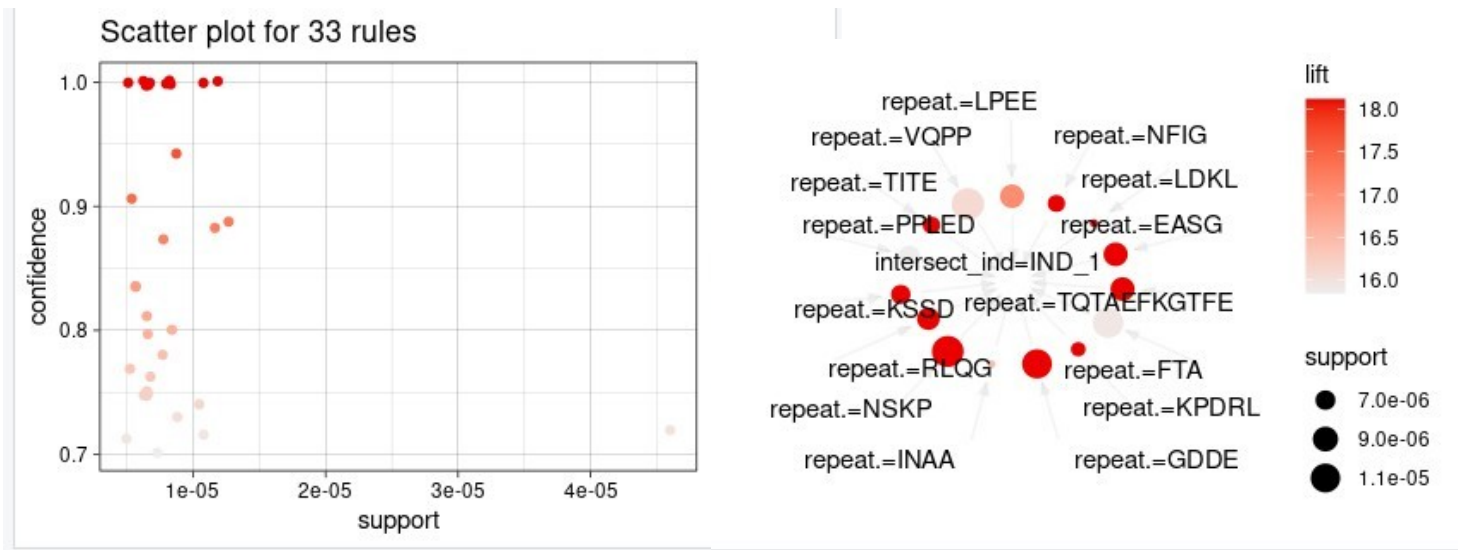
$$\begin{array}{c}
 \text{Rule: } X \Rightarrow Y \\
 \begin{array}{l}
 \nearrow \text{Support} = \frac{\text{freq}(X, Y)}{N} \\
 \rightarrow \text{Confidence} = \frac{\text{freq}(X, Y)}{\text{freq}(X)} \\
 \searrow \text{Lift} = \frac{\text{Support}}{\text{Supp}(X) \times \text{Supp}(Y)}
 \end{array}
 \end{array}$$

Слика 15- начин рачунања вредности подршке, поузданости и лифт мере правила придруживања

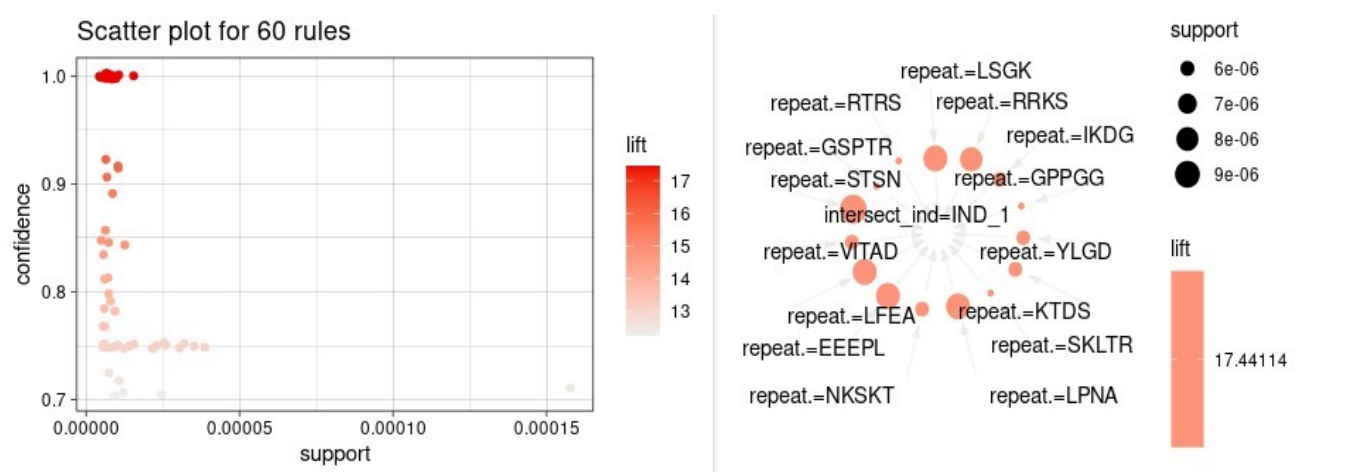
4.2 Налажење правила придруживања

Као што је у претходном делу рада већ поменуто, генерисање одговарајућих правила придруживања обавићемо у програмском језику R. Најпре је потребно скинути неопходне пакете који се користе за генерисање правила придруживања - *arules* и *arulesViz*. То се може урадити покретањем команди `install.packages("arules")` и `install.packages("arulesViz")` редом. Када је тај посао обављен, потребно је одговарајуће фајлове у којима смо издвојили

транзакције преместити у фолдер у коме је инсталиран програмски језик R. Након тога, све је спремно за даљу обраду. Да бисмо добили жељена правила придруживања потребно је покренути програме `associationRulesDirect.R`, `associationRulesIndirect.R` (који редом генеришу правила придруживања на основу фајлова који садрже само поновак и индикатор да ли тај поновак сече регион).

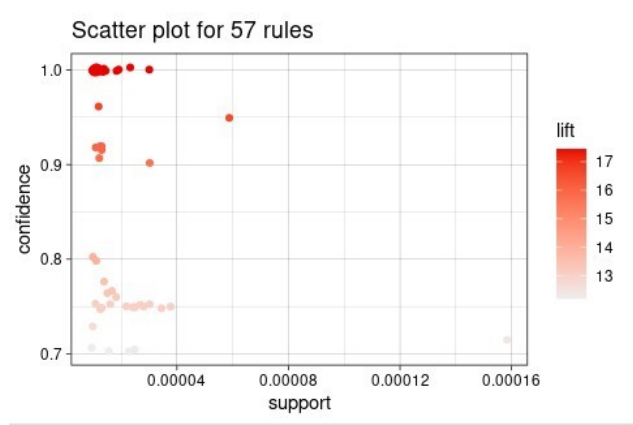


Слика 16- графовске репрезентације правила придруживања за које важи: $\text{minsup}=0.000005$, $\text{minconf}=0.7$, директне секвенце



Слика 17- графовске репрезентације правила придруживања за које важи: $\text{minsup}=0.000005$, $\text{minconf}=0.7$, секвенце

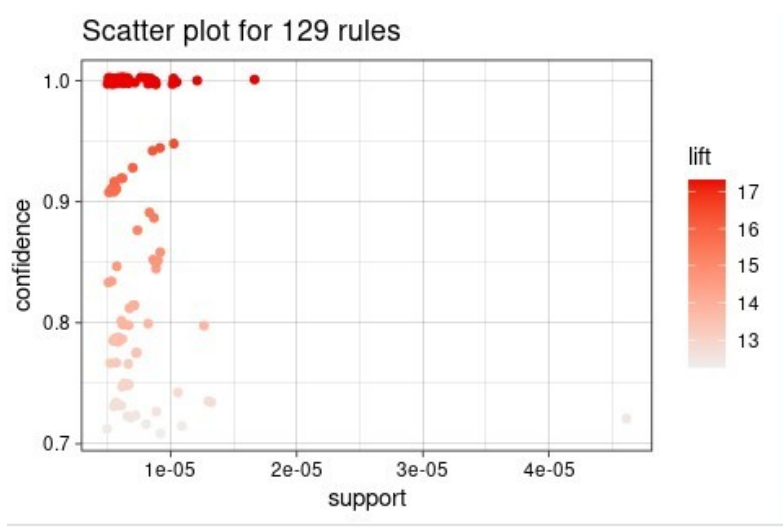
Затим, да бисмо генерисали правила придруживања на основу поновка, индикатора да ли тај поновак пресеца регион и таксономије, написани су програми под називима `associationRulesWithTaxonomyDirect.r` и `associationRulesWithTaxonomyIndirect.R`.



Слика 18- тачкасти дијаграм правила придруживања за које важи: $\text{minsup}=0.00001$, $\text{minconf}=0.7$, секвенце са таксономијом



Слика 19- графовска репрезентација правила придруживања за које важи: $\text{minsup}=0.00001$, $\text{minconf}=0.7$, секвенце са таксономијом

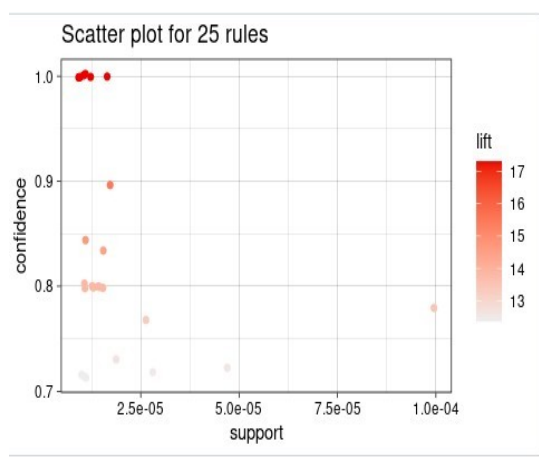


Слика 20- тачкасти дијаграм правила придруживања за које важи: $\text{minsup}=0.000005$, $\text{minconf}=0.7$, директне секвенце са таксономијом

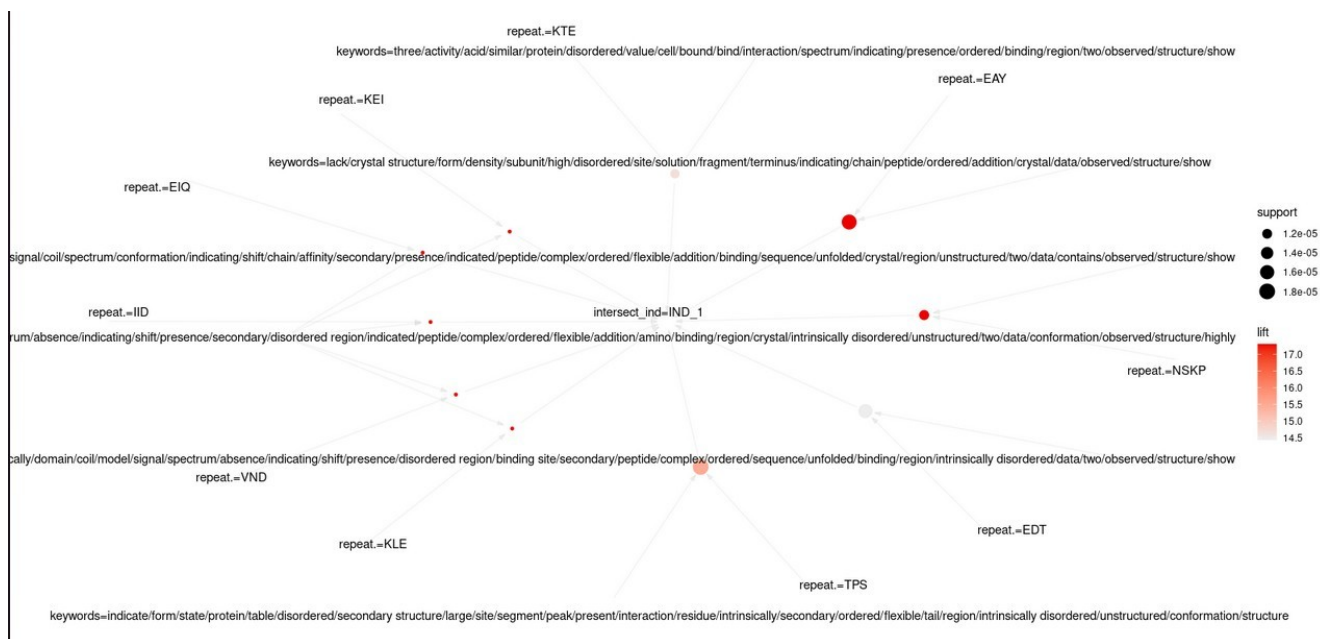


Слика 21- графовска репрезентација правила придруживања за које важи: $\text{minsup}=0.000005$, $\text{minconf}=0.7$, директне секвенце са таксономијом

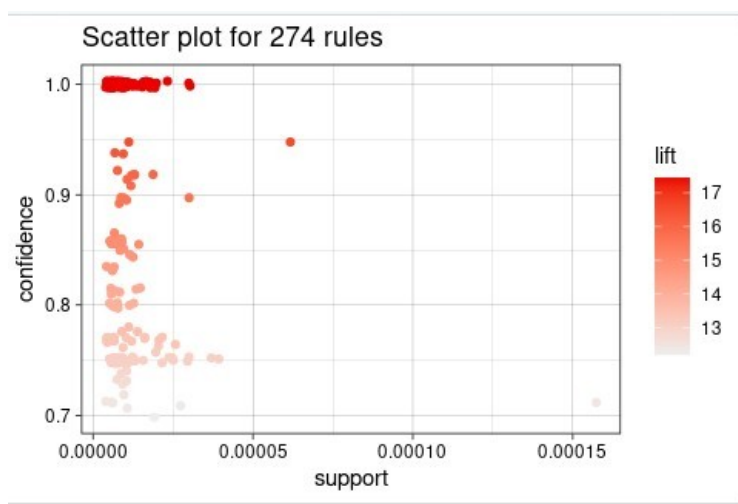
Аналогно претходном случају, уколико желимо и правила која поред стандардних елемената укључују и кључне речи, треба покренути програме `associationRulesWithKeywordsDirect.r` и `associationRulesWithKeywordsndirect.R`.



Слика 22- графовске репрезентације правила придруживања за које важи: $\text{minsup}=0.00001$, $\text{minconf}=0.7$, директне секвенце



Слика 23- графовске репрезентације правила придруживања за које важи: $\text{minsup}=0.00001$, $\text{minconf}=0.7$, директне секвенце са кључним речима

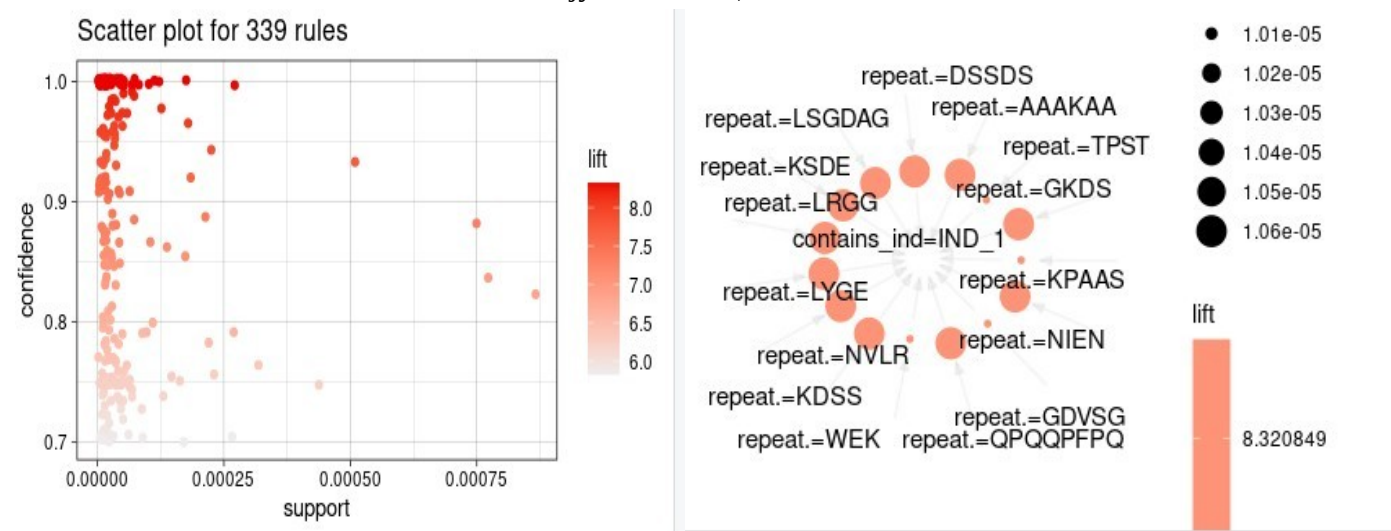


Слика 24- графовске репрезентације правила придруживања за које важи: $\text{minsup}=0.000005$, $\text{minconf}=0.7$, секвенце

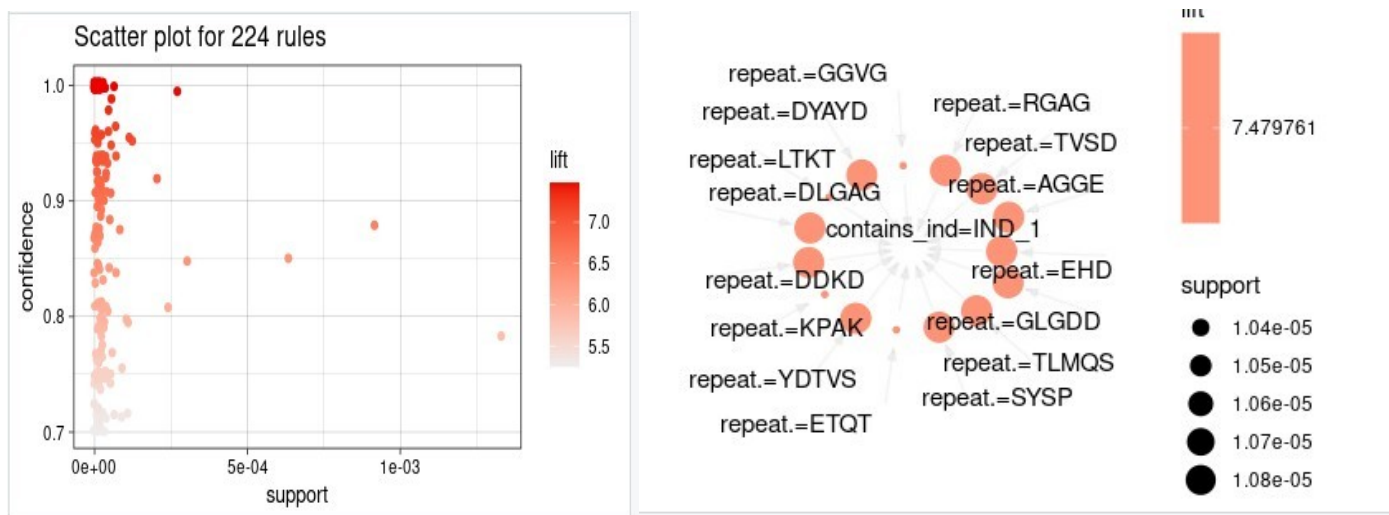


Слика 25- графовске репрезентације правила придруживања за које важи: $\text{minsup}=0.000005$, $\text{minconf}=0.7$, секвенце са кључним речима

Исти поступак применимо и на датотеке које уместо индекса пресека садрже информације да ли неуређени региони садрже поновке. За правила која садрже само индекс и понављачи написани су програм `assocRulesDirectContains.R` за директне, те програм `assocRulesIndirect.Contains.R` за понављајуће секвенце.

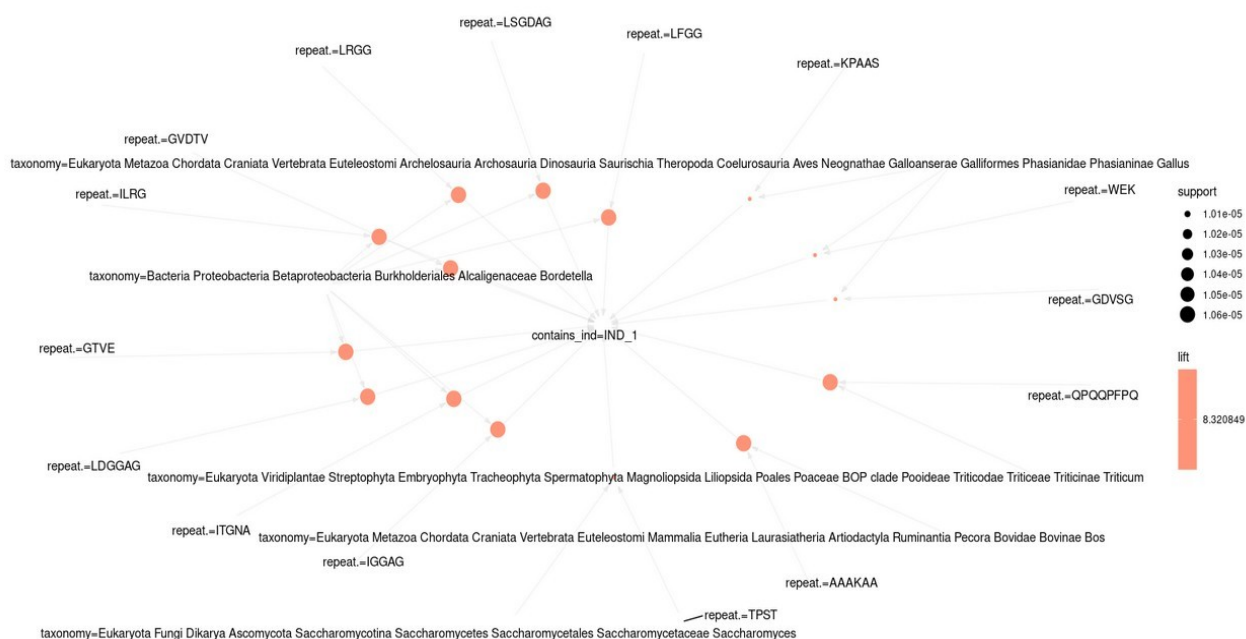


Слика 26- графовске репрезентације правила придруживања за које важи: $\text{minsup}=0.00001$, $\text{minconf}=0.7$, директне секвенце, индекс садржања

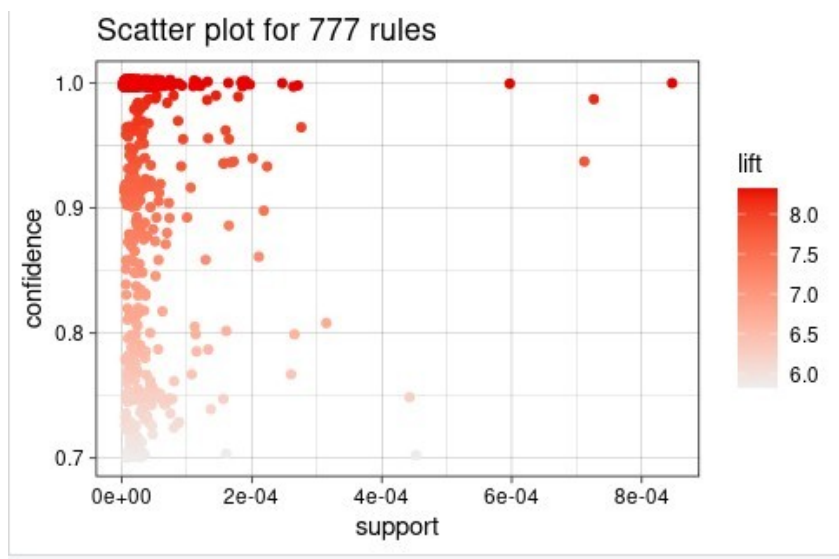


Слика 27- графовске репрезентације правила придруживања за које важи: $\text{minsup}=0.00001$, $\text{minconf}=0.7$, секвенце, индекс садржања

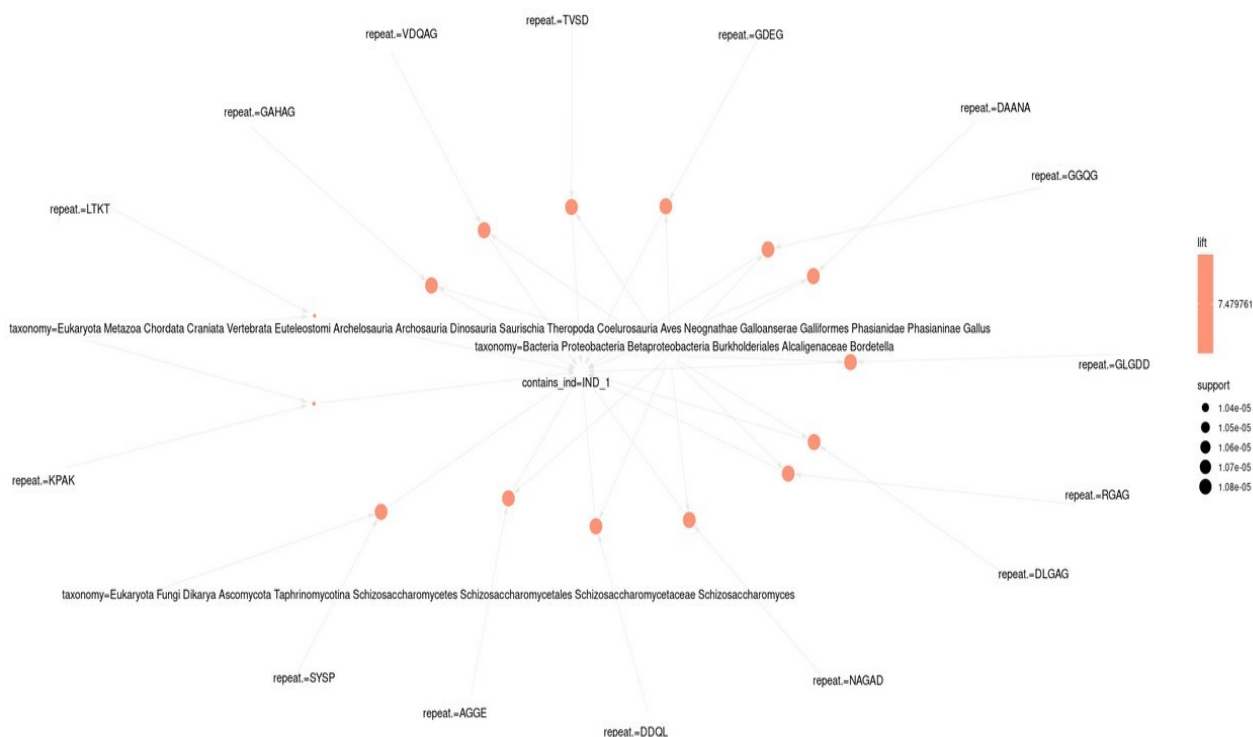
Да бисмо имали увид у то како таксономија врсте утиче на издвајање правила придруживања, написани су следећи програми: `assocRulesWithTaxonomyDirectContains.r` за директне, као и `assocRulesWithTaxonomyIndirectContains.R` за секвенце.



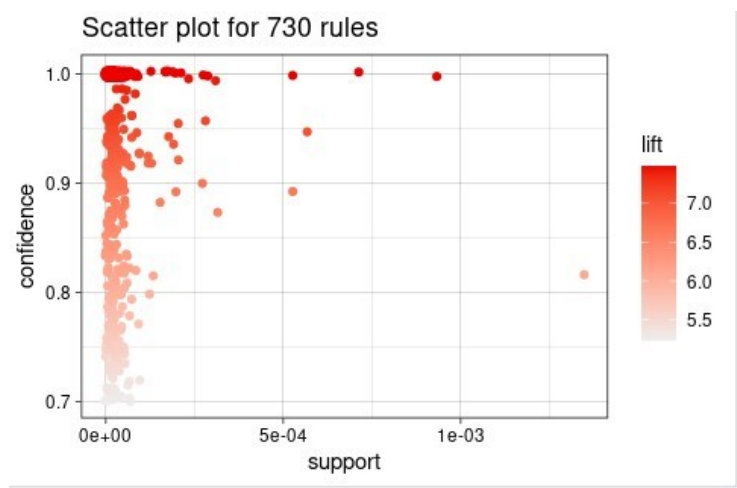
Слика 28- графовска репрезентација правила придруживања за које важи: $\text{minsup}=0.00001$, $\text{minconf}=0.7$, директне секвенце са таксономијом, индекс садржања



Слика 29- тачкасти дијаграм правила придруживања за које важи: $\text{minsup}=0.00001$, $\text{minconf}=0.7$, директне секвенце са таксономијом, индекс садржања

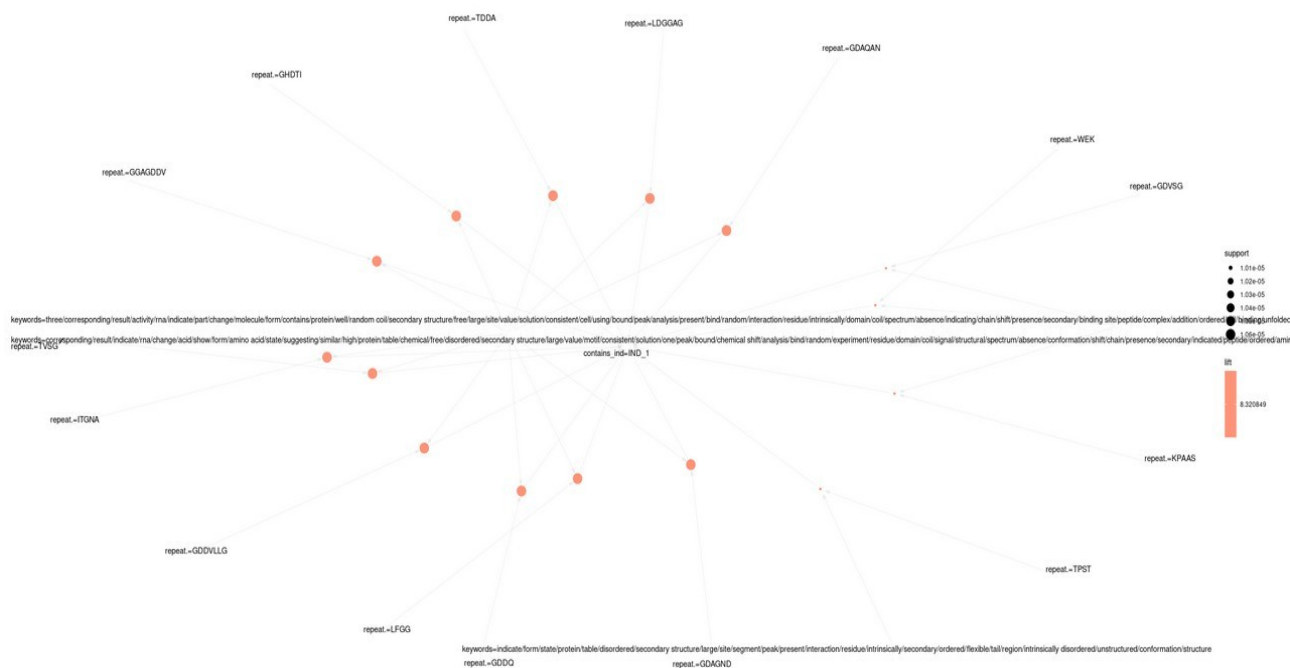


Слика 30- графовска репрезентација правила придруживања за које важи: $\text{minsup}=0.00001$, $\text{minconf}=0.7$, секвенце са таксономијом, индекс садржања

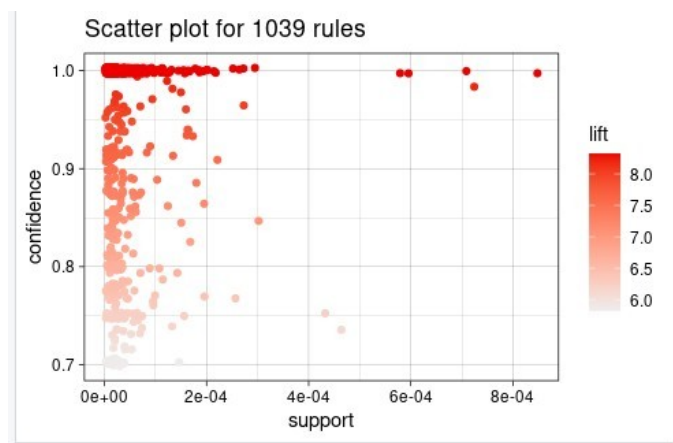


Слика 31- тачкасти дијаграм правила придруживања за које важи: $\text{minsup}=0.00001$, $\text{minconf}=0.7$, секвенце са таксономијом, индекс садржања

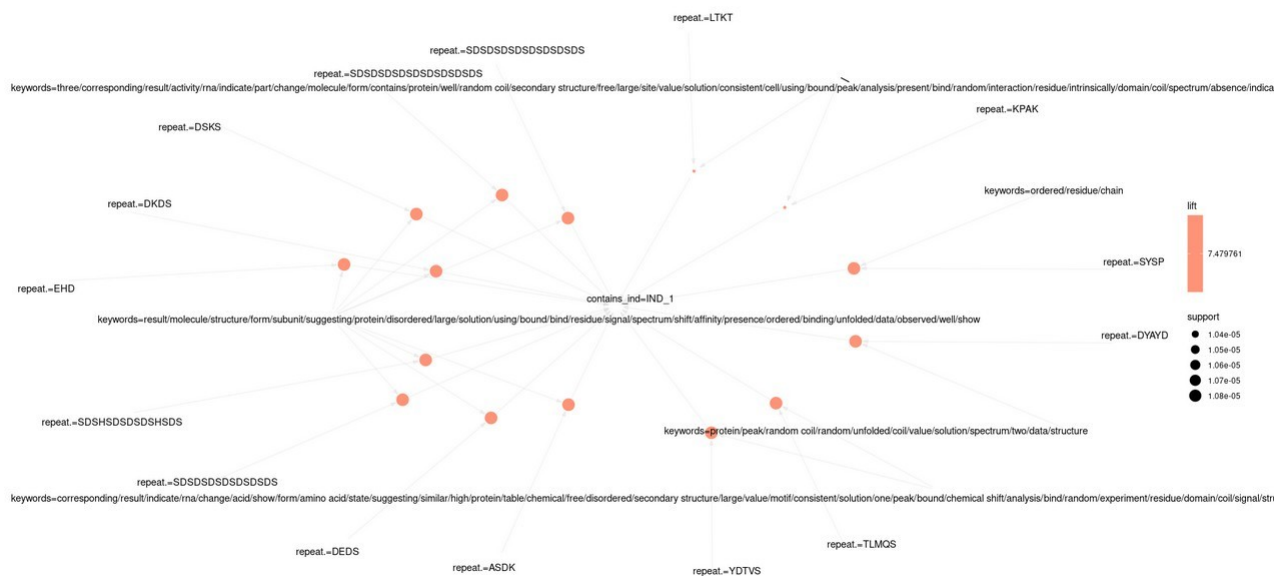
Најзад, за налажење правила која у себи садрже поновак, индекс и листу кључних речи, потребно је покренути програме назване `assocRulesWithKeywordsDirectContains.R` и `assocRulesWithKeywordsndirectContains.R`.



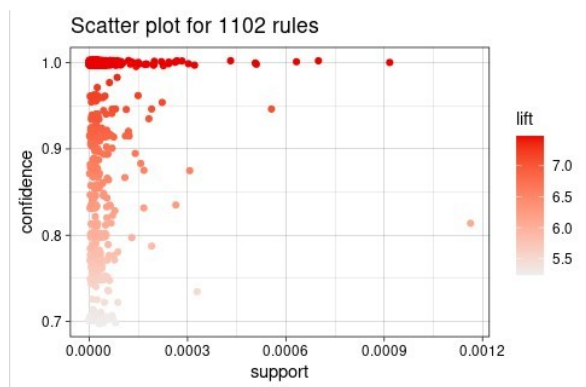
Слика 32- графовска репрезентација правила придруживања за које важи: $\text{minsup}=0.00001$, $\text{minconf}=0.7$, директне секвенце са кључним речима, индекс садржања



Слика 33- тачкасти дијаграм правила придруживања за које важи: $\text{minsup}=0.00001$, $\text{minconf}=0.7$, директне секвенце са кључним речима, индекс садржања



Слика 34- графовска репрезентација правила придруживања за које важи: $\text{minsup}=0.00001$, $\text{minconf}=0.7$, секвенце са кључним речима, индекс садржања



Слика 35- тачкасти дијаграм правила придруживања за које важи: $\text{minsup}=0.00001$, $\text{minconf}=0.7$, секвенце са кључним речима, индекс садржања

Сви горенаведени програми писани су тако да се наредбе извршавају у конзоли, једна по једна, због информација које даје позив функције `arules()`. Као резултат сваког од ових програма генеришу се текстуални фајлови који садрже правила придруживања која је рачунар успео да изгенерише. Сваки добијени фајл садржи следеће информације:

- информација о минималној подршци и поузданости које су прослеђене као аргументи функцији `arules()`
- поновак (који чини леву страну правила)
- информацију о таксономији организама код којих се дати протеин може наћи (уколико је алгоритам покренут над фајлом који садржи ове информације)
- индикатор да дати поновак пресеца неки неуређени регион (десна страна правила)
- вредност подршке датог правила
- вредност поузданости датог правила
- покривеност свеукупних података датим правилом
- вредност лифт мере сваког правила, као и
- укупан број генерисаних правила.

Поред наведених датотека, ради лакше анализе, за сва добијена правила придруживања генерисани су и графици који су добијени њиховом визуализацијом. Неки од тих графова приказани су у тексту горе, док се остатак графова може наћи у прослеђеном материјалу. Један од графикона приказује позицију правила придруживања у координатном систему одређеном мерама подршке (x оса) и поузданости (y оса). Свака тачка представља по једно правило, док њена боја представља вредност лифт мере тог правила (што је тачка тамније боје, то је већа вредност лифт мере), док други представља графовску репрезентацију (најчешће првих 15) правила придруживања. Правила су и у овом случају представљена тачкама, чија величина зависи од вредности подршке правила (што већа тачка, то је већа подршка датог правила), док њена боја зависи од лифт мере (поново, што је већа вредност лифт мере, то је тачка тамније боје и обратно). Елементи који се налазе са узрочне стране правила налазе се исписани на графу, стрелицама су повезани са свим тачкама (правилима) у којима учествују, док су све тачке стрелицама повезане са елементима који се

налазе са последичне стране правила (у нашем случају, у центру графа су увек одговарајући индекси).

У следећој табели могу се видети најважније информације о сваком фајлу са правилима придруживања који је генерисан.

| filename | min supp | min conf | min lift gen | max lift gen | num of rules |
|--------------------------------|----------|----------|--------------|--------------|--------------|
| rules1.txt | 0.00001 | 0.7 | 12.93787 | 18.11301 | 7 |
| rules2.txt | 0.000005 | 0.7 | 12.67911 | 18.11301 | 33 |
| rules3.txt | 0.000005 | 0.5 | 9.056506 | 18.11301 | 85 |
| rulesIndirect1.txt | 0.0001 | 0.7 | | | 1 |
| rulesIndirect2.txt | 0.00001 | 0.7 | 12.27340 | 17.44114 | 21 |
| rulesIndirect3.txt | 0.000005 | 0.7 | 12.27340 | 17.44114 | 60 |
| rulesIndirect4.txt | 0.000005 | 0.5 | 8.720572 | 17.441144 | 92 |
| rulesWithTaxonomy1.txt | 0.00001 | 0.7 | 12.36353 | 17.30894 | 12 |
| rulesWithTaxonomy2.txt | 0.000005 | 0.7 | 12.26050 | 17.30894 | 129 |
| rulesWithTaxonomy3.txt | 0.000005 | 0.5 | 8.654468 | 17.308936 | 320 |
| rulesWithKeywords1.txt | 0.00001 | 0.7 | 12.36353 | 17.30894 | 25 |
| rulesWithKeywords2.txt | 0.000005 | 0.7 | 12.21807 | 17.30894 | 226 |
| rulesWithKeywords3.txt | 0.000005 | 0.5 | 8.654468 | 17.308936 | 528 |
| rulesWithTaxonomyIndirect1.txt | 0.0001 | 0.7 | 12.43964 | 12.43964 | 1 |
| rulesWithTaxonomyIndirect2.txt | 0.00001 | 0.7 | 12.20880 | 17.44114 | 57 |
| rulesWithTaxonomyIndirect3.txt | 0.000005 | 0.7 | 12.20880 | 17.44114 | 177 |
| rulesWithTaxonomyIndirect4.txt | 0.000005 | 0.5 | 8.720572 | 17.44114 4 | 294 |
| rulesWithKeywordsIndirect1.txt | 0.0001 | 0.7 | 12.43964 | 12.43964 | 1 |
| rulesWithKeywordsIndirect2.txt | 0.00001 | 0.7 | 12.20880 | 17.44114 | 76 |

| | | | | | |
|--|----------|-----|----------|-----------|------|
| rulesWithKeywordsIn direct3.txt | 0.000005 | 0.7 | 12.20880 | 17.44114 | 274 |
| rulesWithKeywordsIn direct4.txt | 0.000005 | 0.5 | 8.720572 | 17.441144 | 455 |
| rulesContains1.txt | 0.0001 | 0.7 | 5.843998 | 8.320849 | 27 |
| rulesContains2.txt | 0.00001 | 0.7 | 5.824594 | 8.320849 | 339 |
| rulesContains3.txt | 0.000005 | 0.7 | 5.824594 | 8.320849 | 619 |
| rulesContains4.txt | 0.000005 | 0.5 | 4.160424 | 8.320849 | 812 |
| rulesIndirectContains 1.txt | 0.001 | 0.7 | 5.844892 | 5.844892 | 1 |
| rulesIndirectContains 2.txt | 0.0001 | 0.7 | 5.365341 | 7.465124 | 12 |
| rulesIndirectContains 3.txt | 0.00001 | 0.7 | 5.248955 | 7.479761 | 224 |
| rulesIndirectContains 4.txt | 0.000005 | 0.7 | 5.248955 | 7.479761 | 430 |
| rulesIndirectContains 5.txt | 0.00005 | 0.5 | 3.739881 | 7.479761 | 636 |
| rulesWithKeywordsC ontains1.txt | 0.0001 | 0.7 | 5.855412 | 8.320849 | 65 |
| rulesWithKeywordsC ontains2.txt | 0.00001 | 0.7 | 5.824594 | 8.320849 | 1039 |
| rulesWithKeywordsC ontains3.txt | 0.000005 | 0.7 | 5.824594 | 8.320849 | 2047 |
| rulesWithKeywordsIn directContains1.txt | 0.001 | 0.7 | 6.094832 | 6.094832 | 1 |
| rulesWithKeywordsIn directContains2.txt | 0.0001 | 0.7 | 5.502967 | 7.479761 | 55 |
| rulesWithKeywordsIn directContains3.txt | 0.00001 | 0.7 | 5.235833 | 7.479761 | 1102 |
| rulesWithKeywordsIn directContains4.txt | 0.000005 | 0.7 | 5.235833 | 7.479761 | 2386 |
| rulesWithTaxonomyC ontains1.txt | 0.0001 | 0.7 | 5.832023 | 8.320849 | 51 |
| rulesWithTaxonomyC ontains2.txt | 0.00001 | 0.7 | 5.824594 | 8.320849 | 777 |
| rulesWithTaxonomyC ontains3.txt | 0.000005 | 0.7 | 5.824594 | 8.320849 | 1361 |
| rulesWithTaxonomyI | 0.001 | 0.7 | 6.08858 | 6.08858 | 1 |

| | | | | | |
|--|----------|-----|----------|----------|------|
| ndirectContains1.txt | | | | | |
| 1ulesWithTaxonomyI ndirectContains2.txt | 0.0001 | 0.7 | 5.370085 | 7.479761 | 32 |
| rulesWithTaxonomyI ndirectContains3.txt | 0.00001 | 0.7 | 5.235833 | 7.479761 | 730 |
| rulesWithTaxonomyI ndirectContains4.txt | 0.000005 | 0.7 | 5.235833 | 7.479761 | 1387 |

Табела 1- Сажетак најважнијих информација у вези правила придруживања

5.Анализа

Подсетимо се, неуређени региони имају неке веома битне улоге у нормалном функционисању протеина. Међутим, испоставило се да грешке у њиховом раду узрокују разне болести попут Алцхајмера, Паркинсонове болести и рака. Због тога је веома битно иамти механизам који ће анализирати ове регионе, наћи поновке који потенцијално доводе до тешких болести, те на тај начин допринети раном откривању сколностима ка болестима на које овакви протеини утичу.

Један начин за откривање поновака који се налазе у неуређеним регионима или их секу јесте откривање правила придруживања, чиме се овај рад и бавио. Најпре је било потребно пронаћи позиције директних и инверзних комплементарних секвенци у протеинима из базе. Затим су издвојене позиције неуређених региона, као и неке помоћне карактеристике које би могле помоћи при генерисању правила попут таксономије и кључних речи. Када имамо све потребне податке, потребно их је објединити на једно место, додавањем индекса пресецања (има вредност 1 ако се поновак сече са неуређеним регионом, 0 иначе), и индекс садржања (аналогно претходном случају, има вредност 1 уколико неуређени регион садржи поновак, 0 иначе). Напокон, из добијених података издвојена су правила придруживања. Највећи проблем јесте било одређивање прага подршке зато што је скуп података веома обиман, а секвенце су ретке. За анализу квалитета добијених правила користимо податке из горенаведене табеле, као и изгенерисане графове.

Као што можемо видети, сва добијена правила придруживања имају високу вредност лифт мере, што значи да постоји јака корелација између узрочне и последичне стране правила. Но, и упркос томе, правила која садрже индекс пресека имају већу вредност од правила у којима се јавља индекс садржања. Даље, анализом графова можемо приметити да правила у којима учествује индекс пресека претежно имају поузданост у интервалу [0.7-0.8],

док се поузданост правила са индексом садржања најчешће налази у опсегу [0.9-1], те можемо закључити да је већа вероватноћа да су пронађена правила заиста тачна, а не да су се ту нашла случајно. Такође, индекси садржања истовремено дају и већу вредност подршке у односу на индекс пресека, што нас доводи до закључка да се ова правила чешће јављају. Дакле, индекси садржања дају нам квалитетнија правила од индекса пресека.

Да се приметити и да се у оба случаја генерише знатно више правила када се дода и таксономија. И не само то, већ су и вредности лифт мере приближне вредностима добијеним само на основу поновака и индекса, што нам говори о високом квалитету добијених правила. Занимљива је и чињеница да су правила добијена анализом директних секвенци које садрже индекс пресека боља од правила добијених посматрањем инверзних секвенци. У оба случаја, вредности подршке и лифт мере су сличне, али је поузданост у просеку већа код правила добијених из инверзних секвенци. Са друге стране, ако посматрамо индекс садржања, долазимо до закључка да су овај пут квалитетнија правила добијена из инверзних секвенци. Поново, вредност подршке и лифт мере је приближно иста у оба случаја, али поузданост је боља код правила издвојених из инверзних секвенци.

Још већи број правила добија се ако се посматрају и кључне речи, што нас доводи до закључка да особине протеина и процеси који се у њима дешавају више утичу на то да ли се поновак налази у региону или га сече него ли сама врста у којој је узорак нађен. И опет, као и у претходном случају, вредности лифт мере су веома блиске вредностима које су добијене када кључне речи нису узете у обзир. Исто важи за оба индекса. Као и претходном случају, најбоља правила добијена су обрадом инверзних секвенци са индексом садржања, нешто лошија правила су добијена из директних секвенци (са истим индексом), а следе правила са индексом пресека, прво она добијена из директних секвенци, па тек онда она издвојена из инверзних секвенци.

6. Литература

1. Бабу Мадан (2016), "Допринос неуређених секвенци функционисању протеина, комплексности ћелије и људским болестима", може се наћи на линку; (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5095923/>)
2. Уверски Владимир (2019), "Неуређени региони и њихова "мистериозна" (мета)физика", <https://www.frontiersin.org/articles/10.3389/fphy.2019.00010/full>

3. База неуређених секвенци DisProt, може се наћи на: <https://disprot.org/download>
4. Документација за Python sklearn, Biopython, apyori
5. Онлајн документација везана за .fasta фајлове, која се може пронаћи на линку <https://www.ncbi.nlm.nih.gov/WebSub/html/help/fasta.html>
6. Пратећа документација за коришћење програма StatRepeats
7. Упутство за коришћење nltk библиотеке, <https://realpython.com/nltk-nlp-python/>
8. Списак стоп речи које садржи nltk библиотека, <https://gist.github.com/sebleier/554280>
9. инсталација апликације Jupyter notebook може се наћи на следећем линку <https://jupyter.org/>
10. Званични сајт компаније Oracle, где је могуће наћи информације о програмском језику Java, као и инсталације уколико је то потребно, <https://www.oracle.com/in/java/>
11. Званични сајт за програмски језик Java, где је могуће наћи инсталацију Јава виртуелне машине уколико је то потребно, <https://www.java.com/en/download/manual.jsp>
12. Званични сајт за програмски језик R, где је могуће наћи инсталацију датог програмског језика уколико је то потребно, <https://www.r-project.org>