**Математички факултет**
**Универзитет у Београду**

## Семинарски рад

У оквиру курса Истраживање података 2

# Тема:

Истраживање образаца у патогеним острвима секвенци *Escherichia coli* и *Helicobacter pylori*

**Професор:**
Ненад Митић

**Студенти:**
Велько Продан, 163/2019
Маја Миленковић, 160/2019

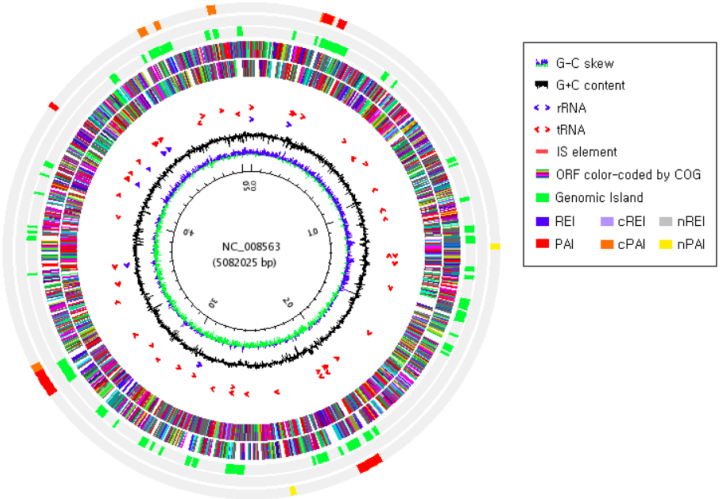
**Мај 2024**

# Садржај

# 1 Увод

## 1.1 Патогена острва

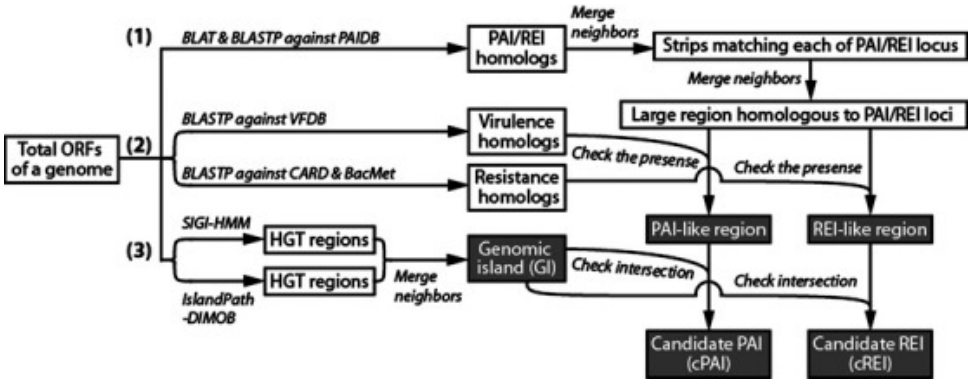
Патогена острва (*Pathogenicity island* - PAI) су посебни генетски елементи присутни на хромозомима великог броја бактеријских патогена. PAI кодирају различите факторе вируленције и обично су одсутни код непатогених сојева исте или блиско сродних врста. Сматрају се подкласом геномских острва која се стичу хоризонталним преносом гена путем трансдукције, конјугације и трансформације, и пружају "квантне скокове"у микробној еволуцији, што доприноси способности микроорганизама да еволуирају [5].




Слика 1: Графички приказ острва унутар генома

Једна бактеријска врста може имати више од једног PAI сегмента. PAI сегменти су кластери гена уграђени у геном патогених организама, хромозомално или екстрахромозомално. PAI могу варирати од 10-200 кб. Они носе гене који им омогућавају да производе различите факторе вируленције, укључујући адхезине, токсине, системе секреције и друге елементе који им помажу да се вежу за ћелије домаћина, избегну имуни систем и изазову болест [10]. Подаци засновани на бројним секвенцираним бактеријским геномима показују да су PAI присутни у широком спектру грам-позитивних и грам-негативних бактеријских патогена људи, животиња и биљака. Недавна истраживања усмерена на PAI довела су до идентификације многих нових фактора вируленције које ове врсте користе током инфекције својих домаћина [5].

Осим PAI, бавићемо се и nPAI и cPAI регионима. “Регион попут PAI” се односи на регију у геному која одговара познатим PAI регионима, и садржи барем један хомолог гена вируленције из PAI региона. Ако се овакав регион преклапа са геномским острвом, зовемо га кандидатом за PAI (cPAI), а у супротном се означава као nPAI (non-probable PAI) [12].




Слика 2: Процедура одређивања типа региона [12]

## 1.2 Ешерихија коли

Ешерихија коли (Escherichia coli) је разноврсна бактеријска врста која обухвата како безопасне коменсалне сојеве, тако и патогене сојеве који се налазе у гастроинтестиналном тракту људи и топлокрвних животиња. Патогеност Ешерихије коли, која се постиже хоризонталним трансфером гена који одређују факторе вируленције, омогућава овој бактерији да постане врло разноврстан и адаптибилан патоген одговоран за цревне или ванцревне болести код људи и животиња. Сходно томе, сојеви E. коли могу се класификовати у три групе: коменсалне/пробиотске сојеве, интестинално патогене сојеве и екстраинтестинално патогене сојеве.

Растућа количина информација о ДНК секвенцама, генерисаних у "ери геномике", помогла је у повећању разумевања фактора и механизама укључених у диверзификацију ове бактеријске врсте [3].

## 1.3 Хеликобактер пилори

Хеликобактер пилори је грам-негативни патоген спиралног облика који насељава антрум и корпус желуца. У последњој деценији, идентификовани су бројни фактори вируленције. Ови елементи омогућавају бактерији да преживи у изузетно киселој средини гастроинтестиналног тракта, доспе до неутралније средине слузног слоја и одупре се имунолошком одговору човека, што резултује трајним задржавањем инфекције [7].

Сојеви Хеликобактер пилори показују висок степен генетске хетерогености због геномских преуређења, тачкастих мутација, убацивања и/или брисања гена. Генетички јединствене варијанте једног соја присутне су у желуцима сваког човека, а генетски састав ових популација може се мењати током времена. Ова адаптабилност доприноси и њеној високој заразности [8].

Већина инфекција се јавља у детињству, а само мали проценат инфекција напредује до тежих стања. Инфекција овом бактеријом може изазвати различите гастроинтестиналне проблеме, укључујући хронични гастритис, чир на дванаестопалачном цреву, па чак и рак [7].

# 2 Алгоритми истраживања образаца

## 2.1 TF-IDF

TF-IDF [6] (Term Frequency - Inverse Document Frequency) алгоритам је статистички метод који се користи за процену важности речи за документ или категорију у скупу датотека или корпусу.

Главна идеја је да ако се нека реч или фраза често појављује у чланку, а ретко се налази у другим чланцима, сматра се да реч или фраза има добру способност разликовања класе и погодна је за класификацију. То је најчешће коришћена функција за израчунавање тежине речи у тренутном векторском моделу простора. Углавном се састоји од два дела, а то су учесталост речи и инверзна учесталост датотеке. Учесталост речи се односи на број појављивања дате речи у датотеци. Инверзна учесталост датотеке представља меру опште важности речи. Инверзна учесталост речи се рачуна деобом укупног броја докумената са бројем докумената који садрже тај термин, а затим се логаритмује резултат количника. Формуле за учесталост речи (TF) и инверзну учесталост текста (IDF) су следеће:

$$\mathrm{tf}_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

$n_{i,j}$ је број појављивања речи $t_i$ у датотеци $d_j$, $\sum_k n_{k,j}$ је збир појављивања свих речи у датотеци $d_j$.

$$\mathrm{idf}_i = \log\left(\frac{|D|}{|\{j : t_i \in d_j\}|}\right)$$

$|D|$ је укупан број датотека у корпусу, $|\{j : t_i \in d_j\}|$ је број докумената који садрже реч $t_i$. Ако речи нема у корпусу, то ће довести до деобе са нулом. Зато се уопштено користи $1+|\{j : t_i \in d_j\}|$.

$$\mathrm{tfidf}_{i,j} = \frac{\mathrm{tf}_{i,j} \times \mathrm{idf}_i}{\sqrt{\sum_{t_i \in d_j} \left[\mathrm{tf}_{i,j} \times \mathrm{idf}_i\right]^2}}$$

$\mathrm{tfidf}_{i,j}$ је тежина речи $t_i$. Може се видети да висока учесталост речи у одређеној датотеци и ниска учесталост датотеке речи у целом скупу датотека могу генерисати високу TF-IDF вредност.

## 2.2 NOSEP

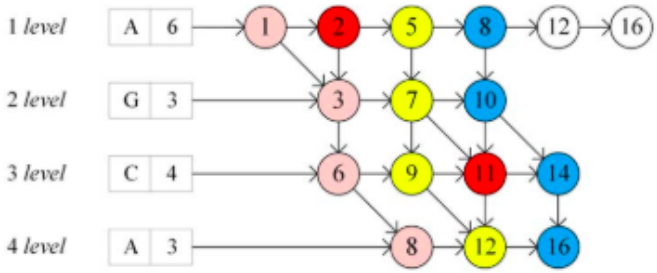
NOSEP [11] (Nonoverlapping Sequence Pattern Mining With Gap Constraints) представља исцрпан алгоритам за рударење образаца секвенци који користи Nettree структуру података. Ова структура омогућава прецизно израчунавање учесталости појављивања одређеног обрасца унутар дате секвенце, узимајући у обзир услов непреклапања.

Приликом рударења образаца секвенци, два главна фактора која утичу на перформансе су израчунавање подршке и смањење простора кандидата образаца.

За израчунавање подршке, користи се NETGAP алгоритам. NETGAP треба да разликује знакове у секвенци који се могу поново користити за упаривање образаца, да би се испунио услов непреклапања. Овај изазов се односи на немогућност једноставне замене одговарајућих знакова у секвенци знаком "X", јер би то спречило

проналажење будућих непреклапајућих појава. Стога, NETGAP мора ефикасно разликовати знакове који се могу поново користити.

Да би се превазишао наведени проблем и имплементирао алгоритам без претраге уназад, предлаже се коришћење Nettree структуре података. Nettree се конструише на основу обрасца и секвенце, а затим се итеративно проналазе минимална путање које представљају појаве обрасца. После сваке итерације, минимална путања и неважећи чворови се уклањају из дрвета. Овај процес се понавља док дрво не постане празно, гарантујући проналажење свих појава обрасца.




Слика 3: Приказ структуре Nettree

Са друге стране, NOSEP алгоритам користи приступ раста образаца за смањење простора кандидата образаца, који се показао ефикаснијим од приступа претраге у ширину и дубину. NOSEP поседује Apriori својство, што имплицира да ако подобразац није фреквентан, његов надобразац такође не може бити фреквентан. Захваљујући овом својству, простор кандидата образаца се ефикасно редукује.

NOSEP започиње рад проналажењем свих честих узорака дужине 1, а затим итеративно генерише кандидатске узорке растуће дужине комбиновањем честих узорака краћих дужина. За сваки кандидатски узорак, његова подршка се израчунава коришћењем NETGAP алгоритма и Nettree структуре података. Кандидати чија подршка задовољава минимални праг се додају у скуп честих узорака за ту дужину. Овај процес се наставља док се не може пронаћи више честих узорака. На крају, NOSEP враћа унију свих пронађених честих узорака свих дужина, ефикасно рударећи комплетан скуп честих секвенцијалних образаца.

# 3 Улазни подаци

Датотеке са геномским секвенцама, и информације о локацијама острва у тим секвенцама, прикупљени су са веб стране *paidb.re.kr* [12]. Геномске секвенце су чуване у FASTA формату, док су индекси почетака и крајева острва чувани у датотекама са одговарајућим суфиксима: *.pai*, *.cpai*, и *.npai*. Број прикупљених генома Ешерихије коли је 90, а генома Хеликобактер пилори је 53.

Сви улазни и излазни подаци за Ешерихију коли се налазе у директоријуму *e_coli*, док се подаци за Хеликобактер пилори налазе у директоријуму *h_pylori*. Улазни подаци се налазе унутар директоријума *e_coli_data* и *h_pylori_data*. FASTA датотеке, као и датотеке са индексима острва за сваки геном се налазе у одговарајућим директоријумима именованим по називу генома.

# 4 Припрема података

FASTA датотеке су учитаване коришћењем библиотеке *Biopython* [2], и секвенце су заједно са индексима острва чуване у Python речнику. За потребе TF-IDF алгоритма, секвенце су конвертоване у ниске.

NOSEP алгоритам је покретан помоћу библиотеке *SPMF* [4], која је имплементирана у програмском језику Java, али је позивана помоћу Python библиотеке *spmf-py* [1]. За употребу *SPMF* библиотеке потребно је кодирати секвенце у одређени формат. Улазна секвенца симбола се прво кодира у низ позитивних целих бројева, где сваки број представља одређени симбол. Поред тога, користе се и специјални симболи за раздвајање карактера (-1) и означавање краја секвенце.

На пример, ако имамо следећу секвенцу:

$$A\,A\,G\,T\,A\,C\,G\,A\,C\,G\,C\,A\,T\,C\,T\,A$$

где су симболи кодирани као:

$$1 = A,\ 3 = C,\ 7 = G,\ 20 = T$$

добија се следећа кодирана секвенца:

$$1\,-1\,1\,-1\,7\,-1\,20\,-1\,1\,-1\,3\,-1\,7\,-1\,1\,-1\,3\,-1\,7\,-1\,3\,-1\,1\,-1\,20\,-1\,3\,-1\,20\,-1\,1\,-1\,-2$$

# 5 Примена алгоритама

## 5.1 TF-IDF

Алгоритам TF-IDF је коришћен помоћу Python библиотеке *scikit-learn* [9], са подразумеваним параметрима. Тражени су обрасци унутар сваког генома, тако да могу да се налазе унутар било којег острва генома, али не и ван њега. Такође су тражени и обрасци који се налазе само делимично унутар неког острва. Минимална дужина образаца који се траже је 4. Максимална дужина тражених образаца за геноме Ешерихије коли је 100, а за геноме Хеликобактер пилори је 50.

Приликом претраге одређених образаца унутар геномске секвенце, узети су у обзир и обрасци који садрже уметања и брисања појединих карактера. Ово је постигнуто коришћењем *pairwise2* модула из библиотеке *Biopython* [2], који користи алгоритам динамичког програмирања за поравнање секвенци.

## 5.2 NOSEP

Улаз NOSEP алгоритма је кодирана секвенца карактера и пет параметара:

1. минимално ограничење дужине (*minlen*)
2. максимално ограничење дужине (*maxlen*)
3. минимално ограничење празнине (*mingap*)
4. максимално ограничење празнине (*maxgap*)

5. задати праг минималне подршке (*minsup*)

Вредности наведених параметара су биле постављене на следећи начин: минимална дужина 4, максимална дужина 100 за геноме Ешерихије коли и 50 за геноме Хеликобактер пилори, минимална празнина 0, максимална празнина 0 и праг минималне подршке 2.

NOSEP је потом покренут над свим геномима и њиховим острвима ради проналажења образаца у оквиру острва који се не налазе ван острва. Међутим, да би се осигурало да пронађени обрасци буду јединствени за острва, уклоњени су они обрасци који су били присутни и ван острва, у свим осталим геномима. Због дугачког времена извршавања овог алгоритма, у неким острвима већих дужина нису претраживани обрасци.

Приликом претраге одређених образаца унутар генома, као и код алгоритма TF-IDF, коришћен је *pairwise2* модул из библиотеке Biopython.

# 6 Излазни подаци

Током извршавања алгоритама, обрасци су чувани у текстуалним датотекама, и у датотекама JSON формата. Текстуалне датотеке се налазе у директоријумима под називом *patterns*, унутар одговарајућег директоријума за сваки геном, у ком се налазе и улазни подаци. JSON датотеке се налазе унутар директоријума *json*, у почетним *e_coli* и *h_pylori* директоријумима. У случају да нема пронађених образаца, одговарајућа текстуална датотека ће да буде празна.

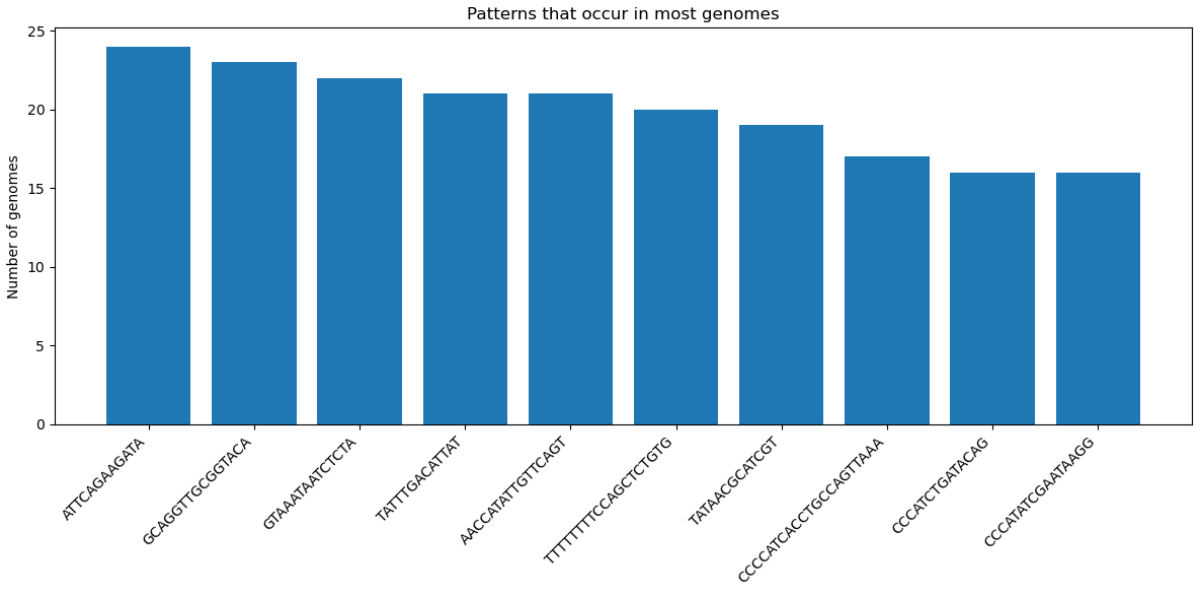
Након примене алгоритама, сви нађени обрасци су сачувани унутар Python речника, где се сваки образац слика у низ генома у којима је нађен, геноми се сликају у низ острва у којима се образац налази, док се острва сликају у број појављивања обрасца у том острву.

Речник је на крају филтриран тако да у њему остану сачувани искључиво обрасци који се не налазе ван острва било ког од низа генома. Број нађених образаца генома Ешерихије коли је 89333, а генома Хеликобактер пилори 21042. Ови обрасци су затим сачувани у табелама *csv* формата, где се обрасци налазе у првој колони, имена генома у заглављима осталих колона, док се у ћелијама налази тип острва тог генома (PAI, cPAI, nPAI) уз одговарајући редни број острва и број појављивања обрасца у том острву. Табеле су сачуване у директоријумима назива *csv*, у почетним *e_coli* и *h_pylori* директоријумима.

# 7 Анализа резултата

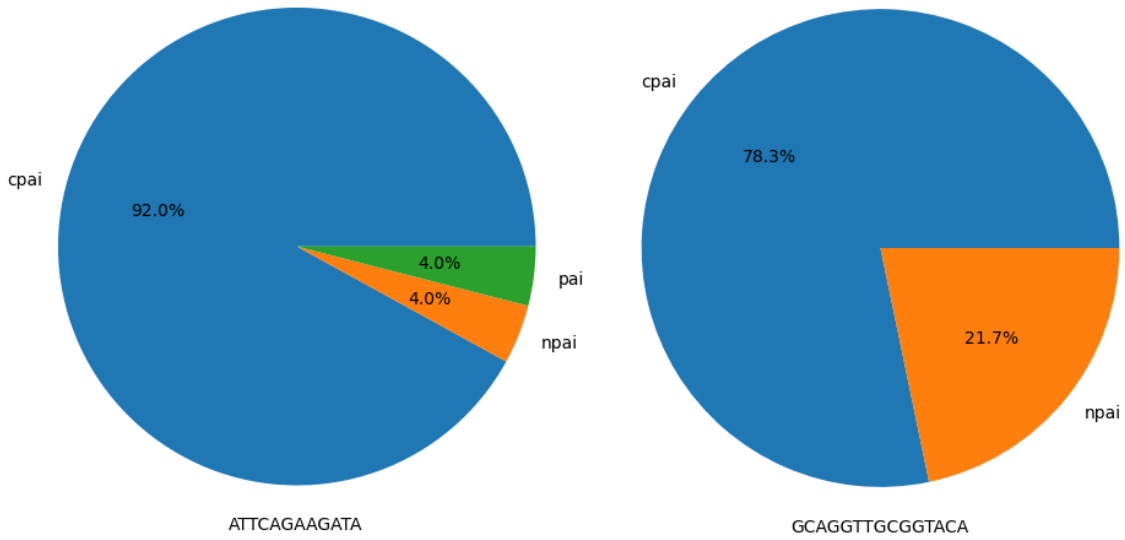
## 7.1 Ешерихија коли

Међу нађеним јединственим обрасцима за острва Ешерихије коли, образац појављиван у највише генома налази се у 24 од 90 генома.



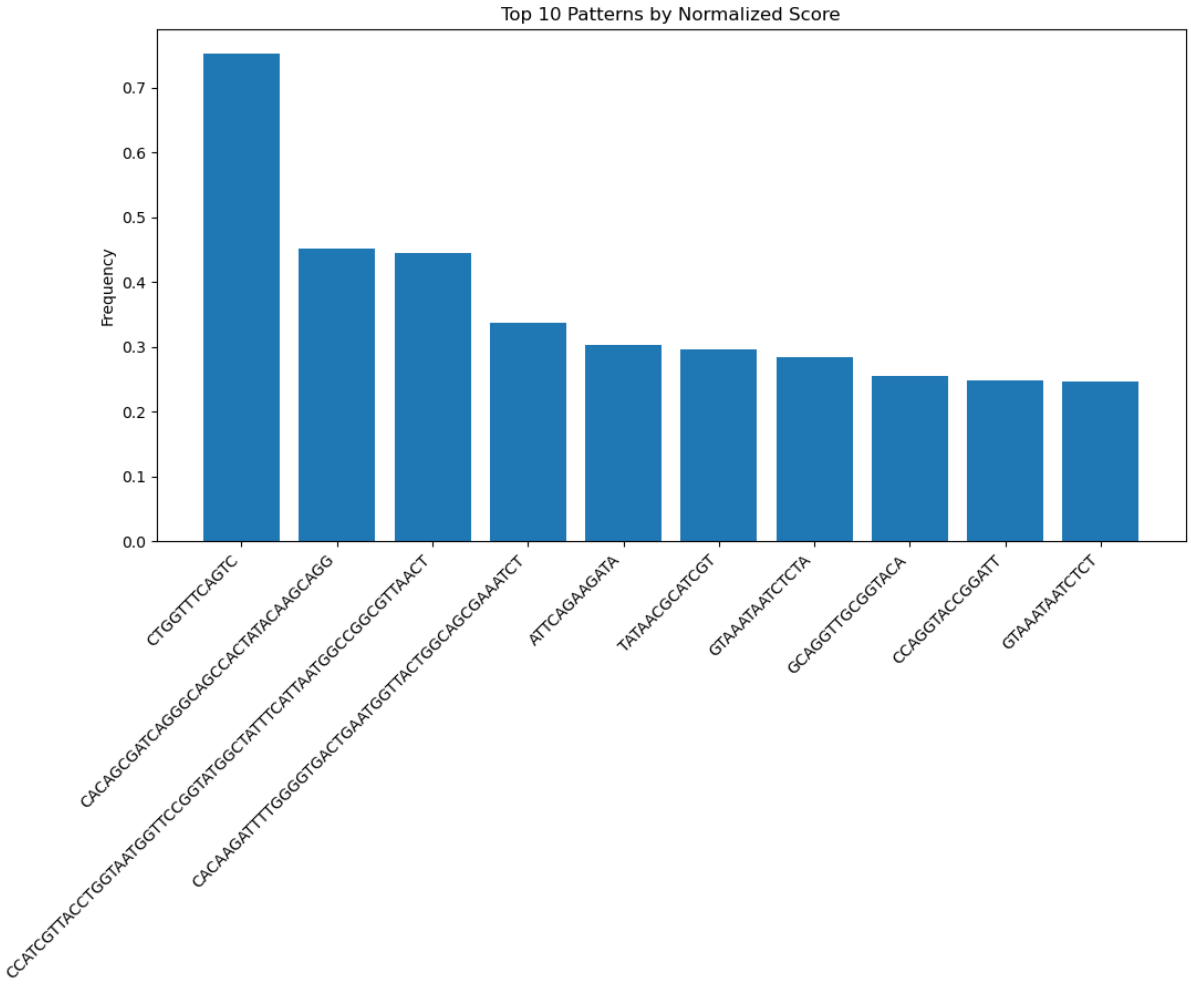

Слика 4: График 10 најчешћих образаца по броју генома *Escherichia coli*

Међу најчешће појављиваним обрасцима за острва Ешерихије коли, већина образаца се налази у cPAI сегментима.



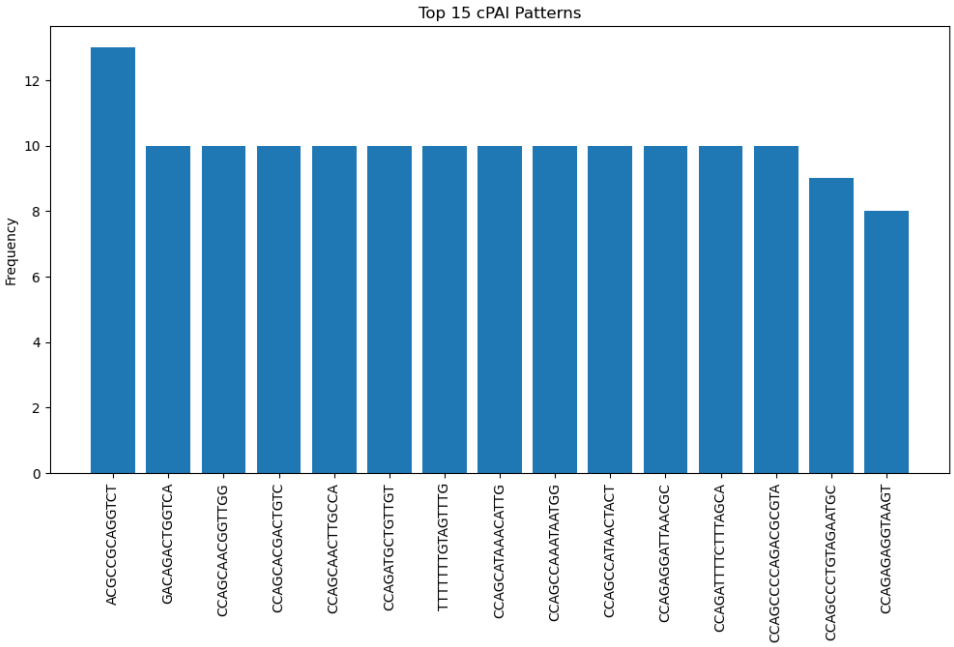

Слика 5: Проценат појављивања два најчешћа обрасца по типу острва *Escherichia coli*

На следећем графику приказани су обрасци са највећом сумом броја појављивања у геномима, у односу на укупну величину острва унутар тих генома.

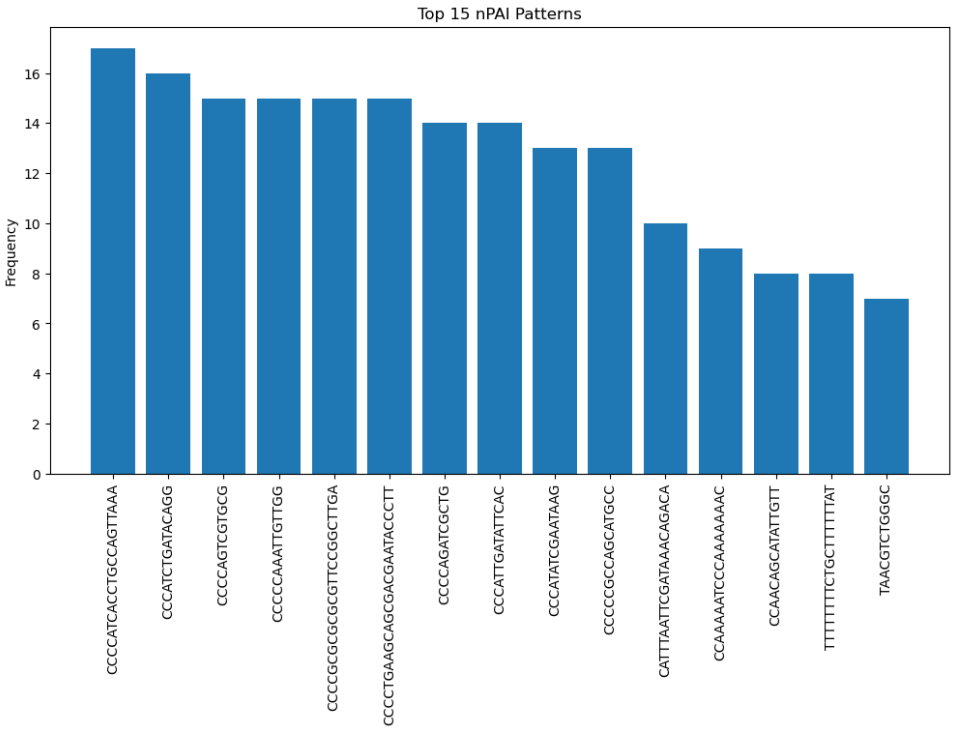



Слика 6: График 10 најчешћих образаца по нормализованом броју појављивања *Escherichia coli*

Образац који се појављује искључиво у cPAI сегментима са највише појављивања се налази у 13 од 90 генома Ешерихије коли. Најчешћи образац јединствен за nPAI сегменте се појављује у 17 генома. Није пронађен ни један образац јединствен за PAI тип острва.



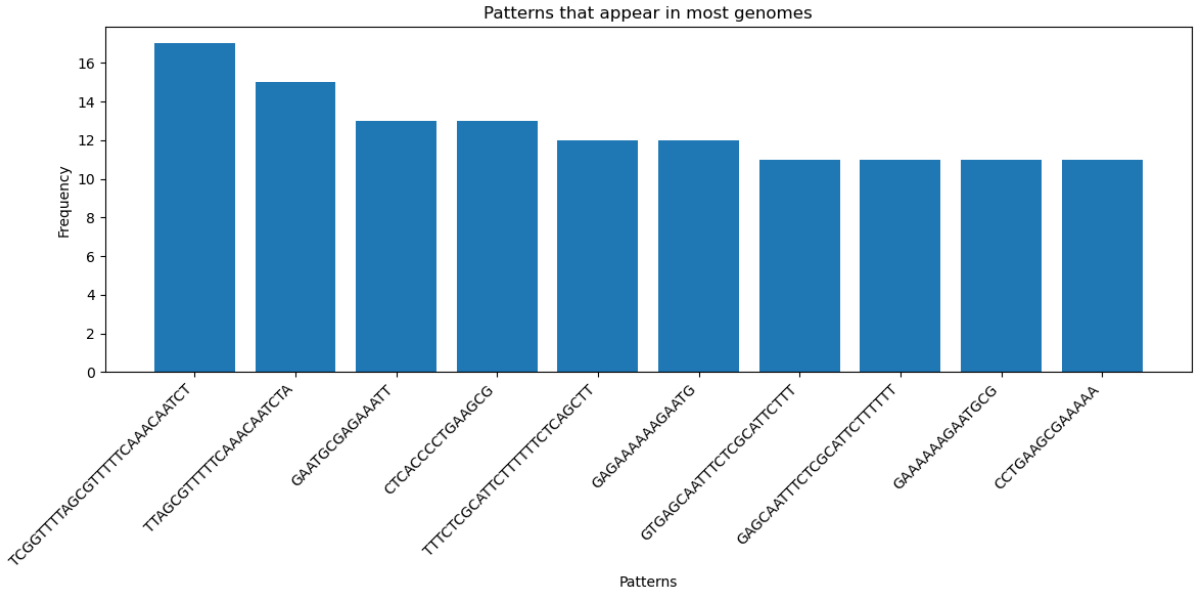

Слика 7: Најчешћи обрасци јединствени за cPAI сегменте *Escherichia coli*




Слика 8: Најчешћи обрасци јединствени за nPAI сегменте *Escherichia coli*

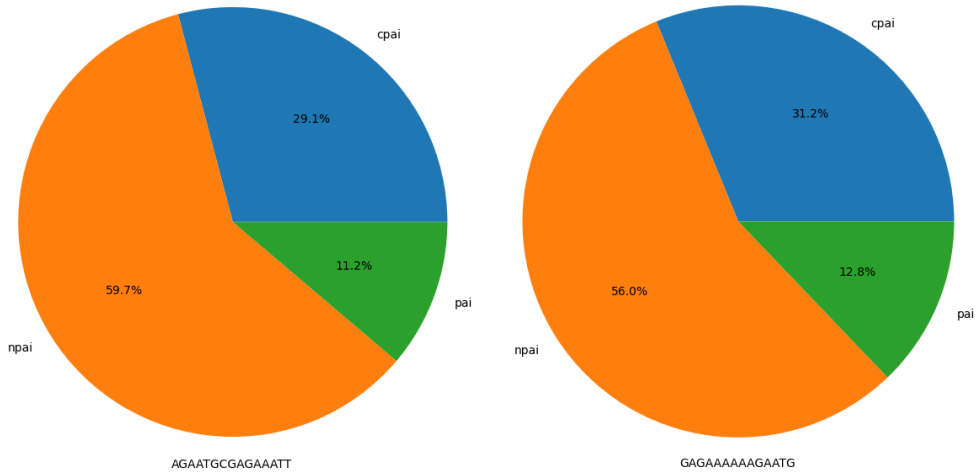
## 7.2 Хеликобактер пилори

Међу нађеним јединственим обрасцима за острва Хеликобактер пилори, образац појављиван у највише генома налази се у 17 од 53 генома.



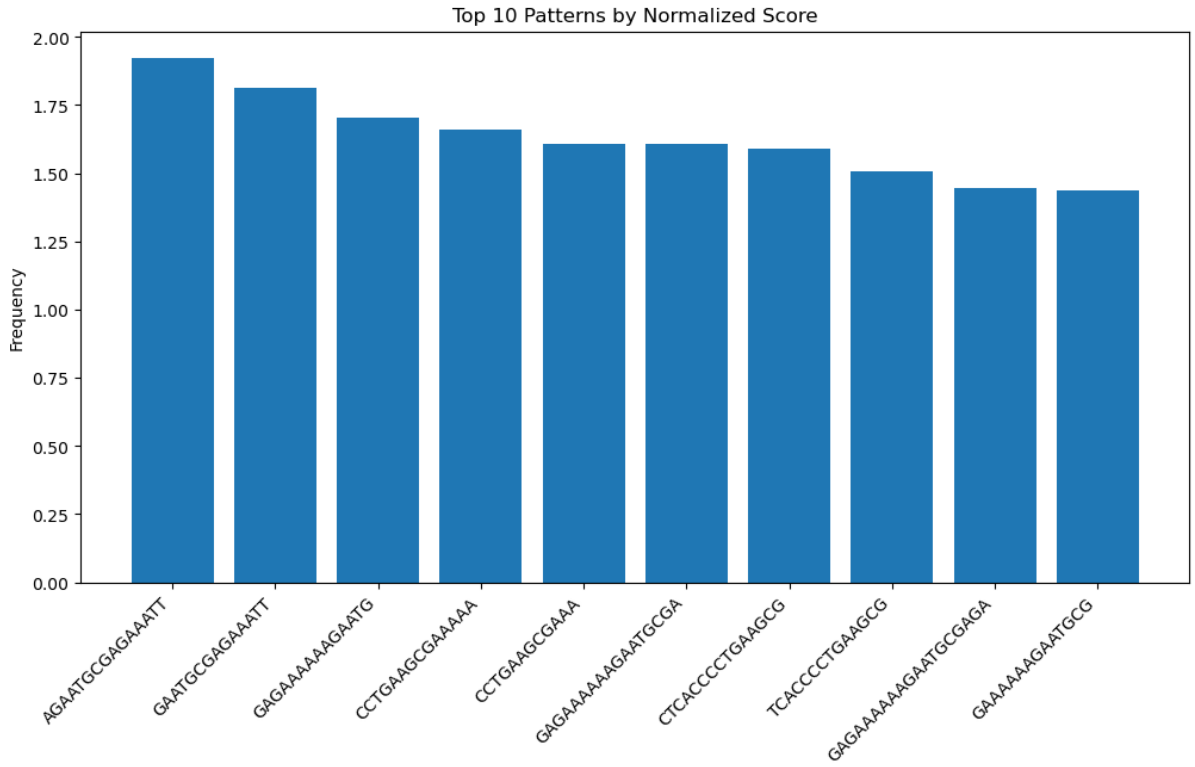

Слика 9: График 10 најчешћих секвенци *Helicobacter pylori*

Два најчешћа обрасца се већином налазе у nPAI сегментима.



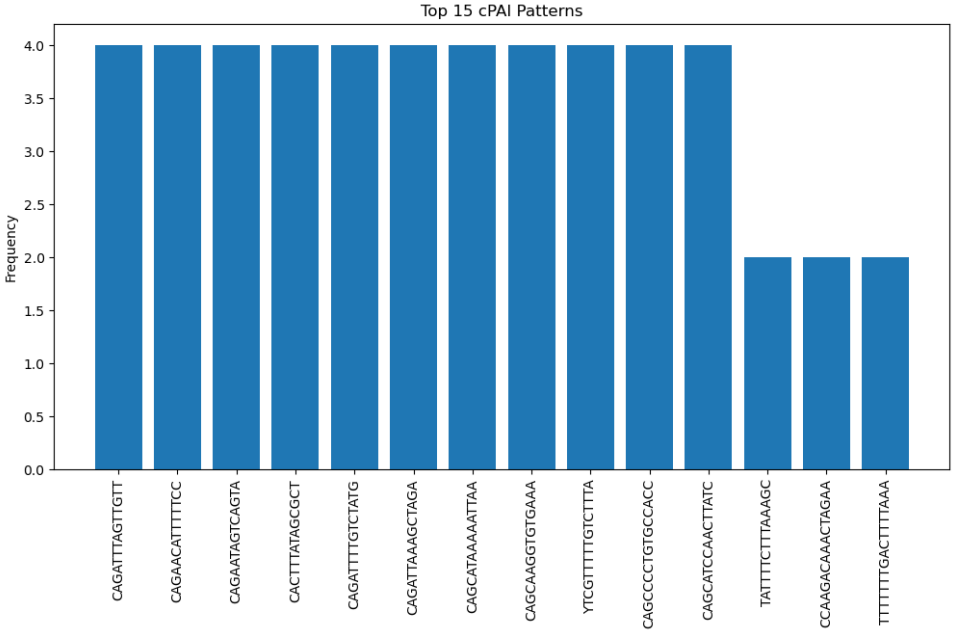

Слика 10: Проценат појављивања два најчешћа обрасца по типу острва *Helicobacter pylori*

На следећем графику приказани су обрасци са највећом сумом броја појављивања у геномима, у односу на укупну величину острва унутар тих генома.

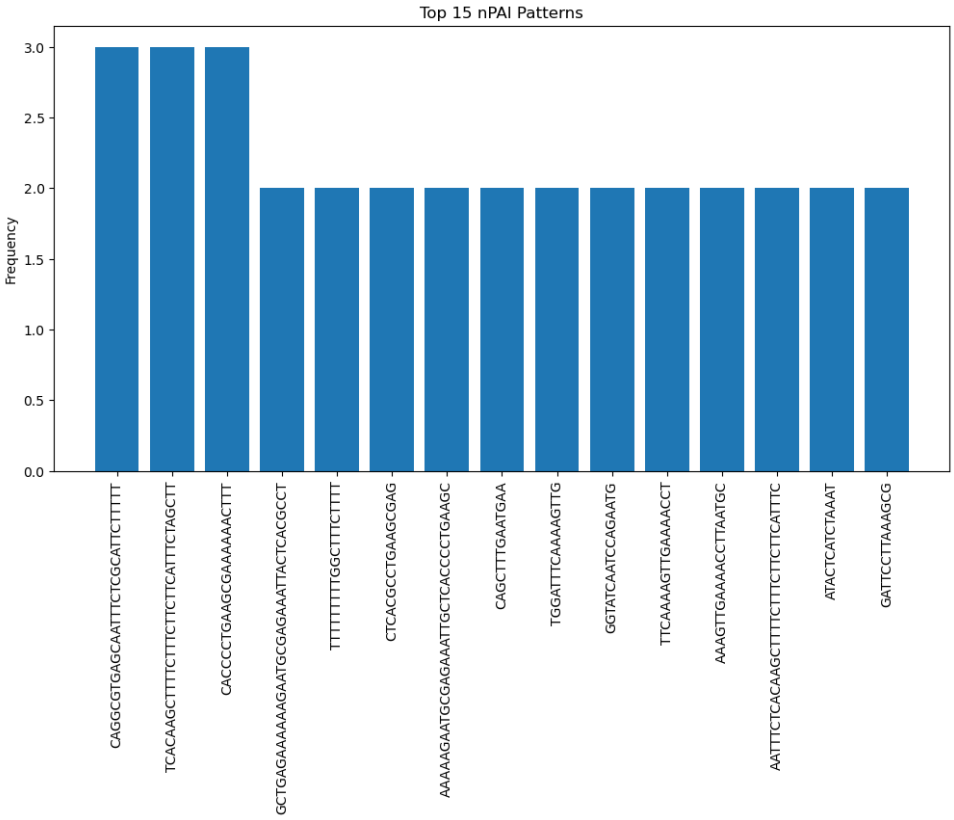



Слика 11: График 10 најчешћих образаца по нормализованом броју појављивања

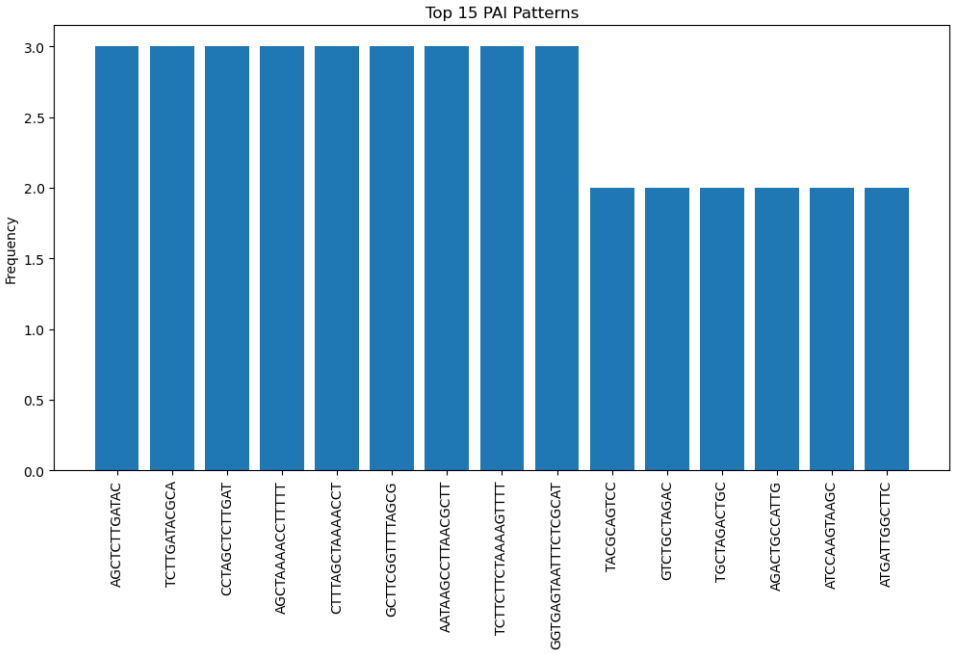
Образац који се појављује искључиво у cPAI сегментима са највише појављивања се налази у 4 од 53 генома Хеликобактер пилори. Најчешћи образац јединствен за nPAI сегменте, и најчешћи образац јединствен за PAI сегменте се појављују у 3 генома.




Слика 12: Најчешћи обрасци јединствени за cPAI сегменте *Helicobacter pylori*




Слика 13: Најчешћи обрасци јединствени за nPAI сегменте *Helicobacter pylori*




Слика 14: Најчешћи обрасци јединствени за PAI сегменте *Helicobacter pylori*

Није пронађен ни један образац који се појављује и у острвима Ешерихије коли, и острвима Хеликобактер пилори.

# Литература


[1] Python wrapper for spmf. `https://pypi.org/project/spmf/`.

[2] Peter JA Cock, Tiago Antao, Jeffrey T Chang, Brad A Chapman, Cymon J Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, et al. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, 2009.

[3] Mickaël Desvaux, Guillaume Dalmasso, Racha Beyrouthy, Nicolas Barnich, Julien Delmas, and Richard Bonnet. Pathogenicity factors of genomic islands in intestinal and extraintestinal escherichia coli. *Front Microbiol*, 11:2065, September 2020.

[4] P. Fournier-Viger, C. W. Lin, A. Gomariz, T. Gueniche, A. Soltani, Z. Deng, and H. T. Lam. The SPMF open-source data mining library version 2. *Proc. 19th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD*, pages 36–40, 2016.

[5] Ohad Gal-Mor and B Brett Finlay. Pathogenicity islands: a molecular toolbox for bacterial virulence. *Cell Microbiol*, 8(11):1707–1719, August 2006.

[6] Cai-Zhi Liu, Yan-Xiu Sheng, Zhi-Qiang Wei, and Yong-Quan Yang. Research of text classification based on improved TF-IDF algorithm. In *2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE)*, pages 218–222, 2018.

[7] Tilahun M, Gedefie A, Belayhun C, Sahle Z, and Abera A. Helicobacter pylori Pathogenicity Islands and Giardia lamblia Cysteine Proteases in Role of Coinfection and Pathogenesis. 2022.

[8] Jennifer M Noto and Richard M Peek, Jr. The helicobacter pylori cag pathogenicity island. *Methods Mol Biol*, 921:41–50, 2012.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[10] Herbert Schmidt and Michael Hensel. Pathogenicity islands in bacterial pathogenesis. *Clin Microbiol Rev*, 17(1):14–56, January 2004.

[11] Youxi Wu, Yao Tong, Xingquan Zhu, and Xindong Wu. Nosep: Nonoverlapping sequence pattern mining with gap constraints. *IEEE Transactions on Cybernetics*, 48(10):2809–2822, 2018.

[12] Sung Ho Yoon, Young-Kyu Park, and Jihyun F Kim. PAIDB v2.0: exploration and analysis of pathogenicity and resistance islands. *Nucleic Acids Res*, 43(Database issue):D624–30, October 2014.