Ročníkový projekt

Analýza frekvencií oligomerov v genomických dátach

Eva Herencsárová

Vedúci projektu: doc. Mgr. Bronislava Brejová, PhD

Konzultant: Mgr. Askar Gafurov

1 Úvod

Genóm sa skladá z DNA, ktorý môžeme chápať, ako reťazec tvorený symbolmi A, C, G, T. V tomto projekte chceme skúmať, ako často sa opakujú k-mery (sekvencie v DNA dĺžky k) v DNA. Budeme sa snažiť fitnúť distribúcie (krivky) na tieto dáta, ktoré ich čo najlepšie opisujú.

2 Zimný semester

2.1 Cieľ

1. fáza: Analýza 1 histogramu: vytvoríme štatistický model, krivku, ktorá čo najpresnejšie opisuje náš histogram
2. fáza: Zopakovanie analýzy pre rôzne organizmy

2.2 Priebeh 1. fázy

Začali sme s genetickou informáciou *Escherichia coli* (baktéria žijúca v hrubom čreve). Najprv s vlastnými triedami som si naprogramovala niekoľko distribúcií. Porovnala som ich s hotovými z knižnice scipy.stats (Statistical functions) a neskôr sme ich aj nahradili s knižničnými.
Vlastná implementácia niektorých distribúcií:
pmf = probability mass function (function that gives the probability that a discrete random variable is exactly equal to some value)

- Poisson 01

```
from scipy.special import gamma
import numpy as np
```

```
class Poisson01:
    def pmf(self, k, alpha):
        alpha = np.exp(alpha)
        return alpha**k * np.exp(-alpha) / (gamma(k+1) * (1 - np.exp(-alpha)))
```

– Poisson 02

```
class Poisson02:
    def pmf(self, k, alpha):
        return alpha**(k-1) * np.exp(-alpha) / gamma(k)
```

– Geometric

```
class Geometric:
    def pmf(self, k, p):
        return p*(1-p)**(k)
```

– Pareto

```
class Pareto:
    def pmf(self, k, alpha):
        a=1
        return alpha*a**alpha / k**(alpha+1)
```

Zistili sme, že krivky by lepšie opisovali dáta, ak by bol výpočet "posunutý - shiftnutý". Na to sme si vytvorili triedu Tailer, ktorá nám vráti niekoľko pôvodných hodnôt (veľkosť shiftnutia) a ostatné budú normalizované a prerobené podľa pmf danej distribúcie:

```
class Tailer:
    def __init__(self, distr, y, q):
        self.distr = distr
        self.y = y
        self.q = q

    def pmf(self, x, alpha):
        y_temp = self.distr.pmf(x, alpha)
        y_result = y_temp * (1-sum(self.y[0:self.q]))/(1-sum(y_temp[0:self.q]))
        for i in range(self.q):
            y_result[i] = self.y[i]
        return np.array(y_result)
```

Ďalej sme chceli použiť aj spojité distribúcie. Preto som vytvorila triedu Discretize na diskretizáciu:

```
class Discretize:
    def __init__(self, cdf):
        self.cdf = cdf

    def pmf(self, x, params):
        return (self.cdf(x+0.5, params) - self.cdf(x-0.5, params))/(1-self.cdf(0.5, params))
```

Potom nasledovalo vykreslovanie dát a kriviek. Použila som funkcie z knižnice matplotlib.pyplot. (matplotlib.pyplot is a collection of command style functions that make matplotlib work like MATLAB)

2.3 Priebeh 2. fázy

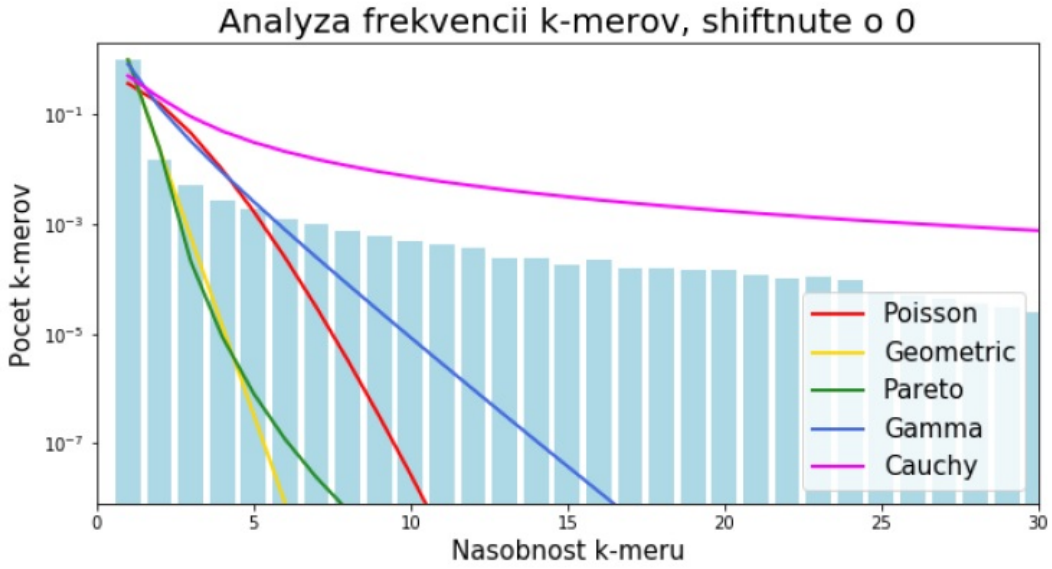
Pomocou programu Jellyfish (Jellyfish is a command-line program for fast, memory-efficient counting of k-mers in DNA) som si vytvorila textové súbory z genomických dát nasledujúcich organizmov pre $k \in \{11, 21, 31\}$:

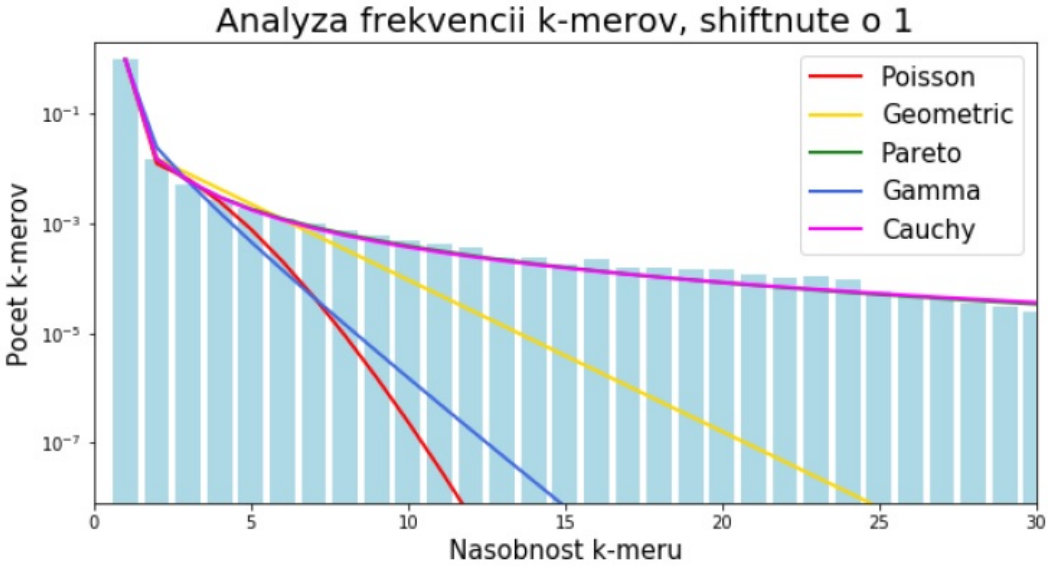
1. *Drosophila melanogaster* (ovocná muška)
2. *Arabidopsis thaliana* (rastlina)
3. *Saccharomyces cerevisiae* (kvasinka pivná)
4. *Staphylococcus aureus* (baktéria)
5. *Escherichia coli* (baktéria)

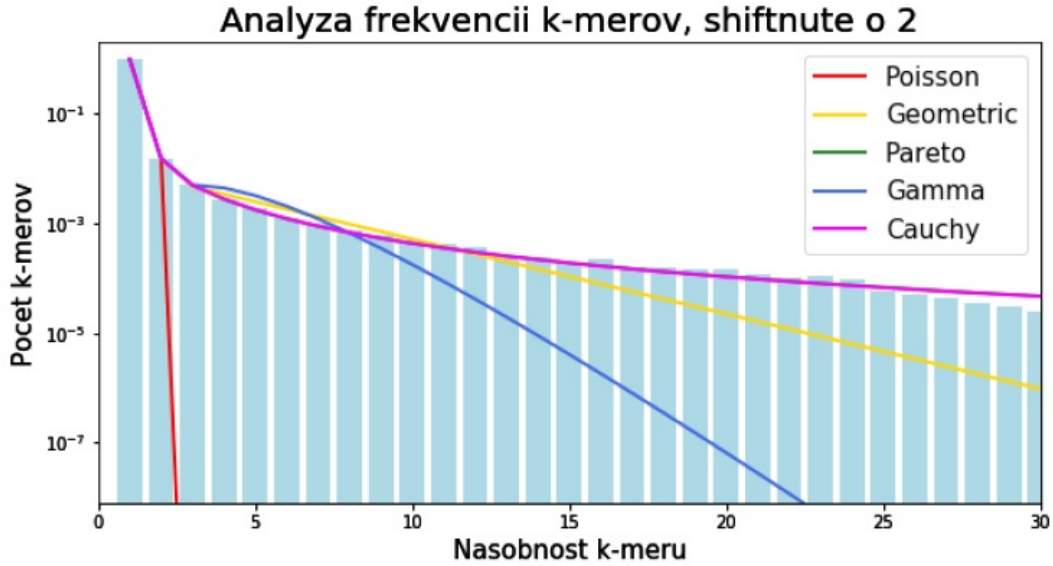
Pre zvolený organizmus a k som vykreslila genomické dáta a distribúcie (*Poisson*, *Geometric*, *Pareto*, *Gamma*, *Cauchy*) pre zadaný počet shiftnutí. Ďalej pre všetky výsledky som spočítala rôzne metriky ($L1$ a $L2$ *norma*) a vypísala 3 *najlepšie* distribúcie.
Následne pre všetky metriky ($L1$ a $L2$) som znázornila presnosť daných rozdelení pre pre zadaný počet shiftnutí.

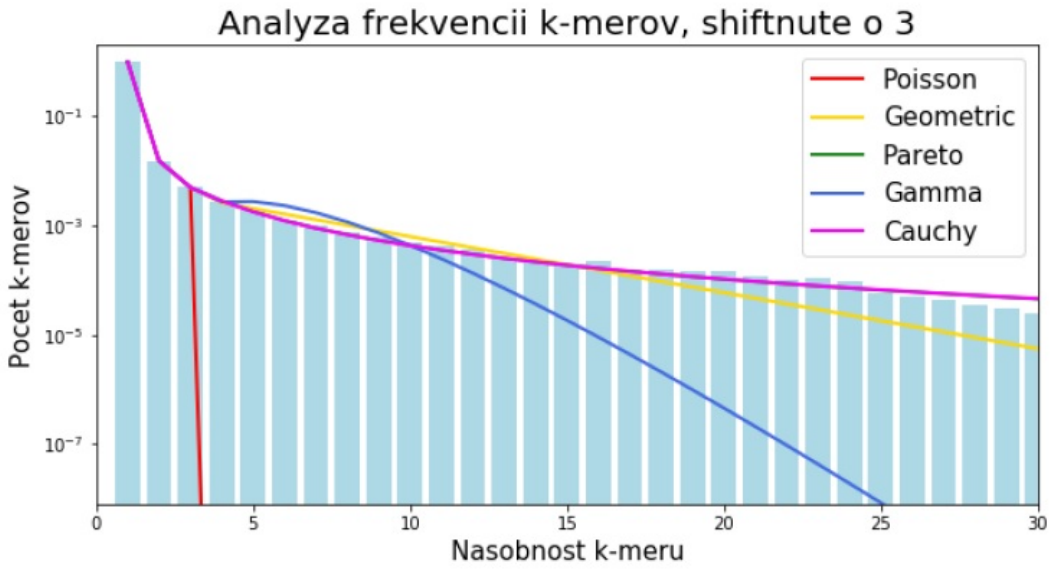
2.4 Výsledky

Výsledkom boli podľa 2.fázy nasledujúce grafy a výpis najpresnejšej distribúcie pri danej metrike:
Na nasledujúcich grafoch som zvolila genomické dáta *Drosophila melanogaster* pre $k = 31$, shiftnutia 0 až 3.




Analyza frekvencii k-merov, shiftnute o 1
Poisson
Geometric
Pareto
Gamma
Cauchy
Pocet k-merov
Nasobnost k-meru


Analyza frekvencii k-merov, shiftnute o 2
Poisson
Geometric
Pareto
Gamma
Cauchy
Pocet k-merov
Nasobnost k-meru



Top 3 for L1 norm:

1. Pareto shifted: 3
2. Cauchy shifted: 3
3. Cauchy shifted: 2

Top 3 for L2 norm:

1. Pareto shifted: 3
2. Cauchy shifted: 2
3. Cauchy shifted: 3

Ďalším výsledkom je aj znázornenie presností daných rozdelení pre daný počet shiftnutí pre všetky metriky ($L1$ a $L2$):
(na obrázku je znova zobrazený predchádzajúci vstup)

3 Letný semester

3.1 Cieľ

Analýza frekvencií pokrytia zo sekvenačných čítaní a rôznych faktorov, ktoré ich frekvenciu ovplyvňujú.

3.2 Spracovanie *bedGraphov*

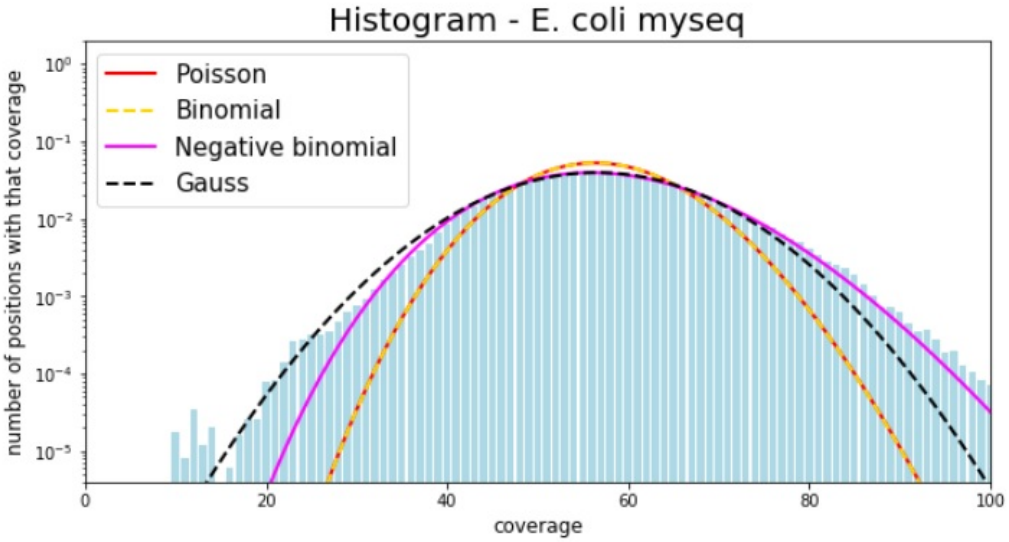
V tomto semestri sme spracovali histogramy zo súborov, kde sme mali pre každú pozíciu v genóme počet, koľko sekvenačných čítaní je na danej pozícií zarovnaných.

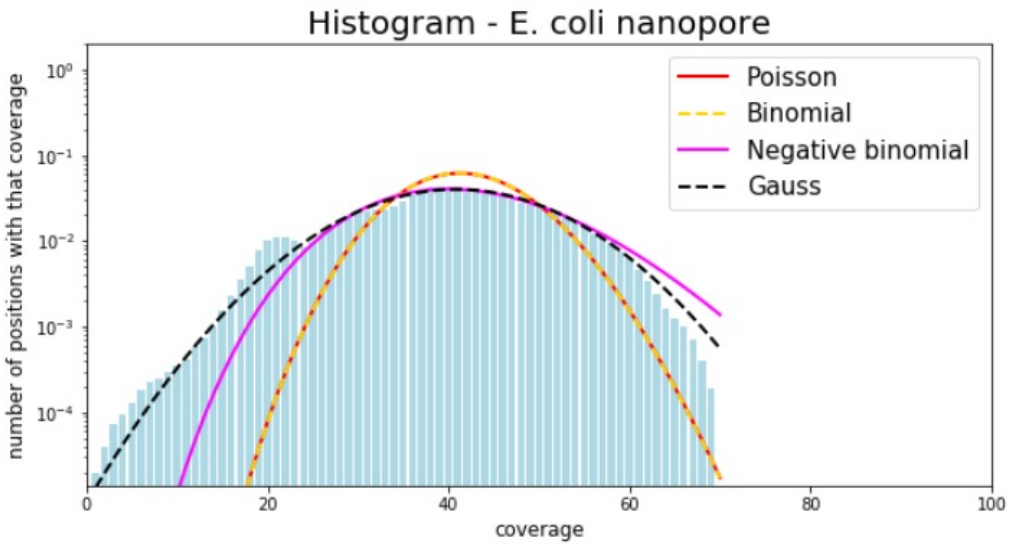
Súbory boli vo formáte *bedGraph*:

$$chromName \;\; chromStart \;\; chromEnd \;\; dataValue$$

kde $dataValue$ značí počet čítaní v polootvorenom intervale $chromStart$ až $chromEnd$.

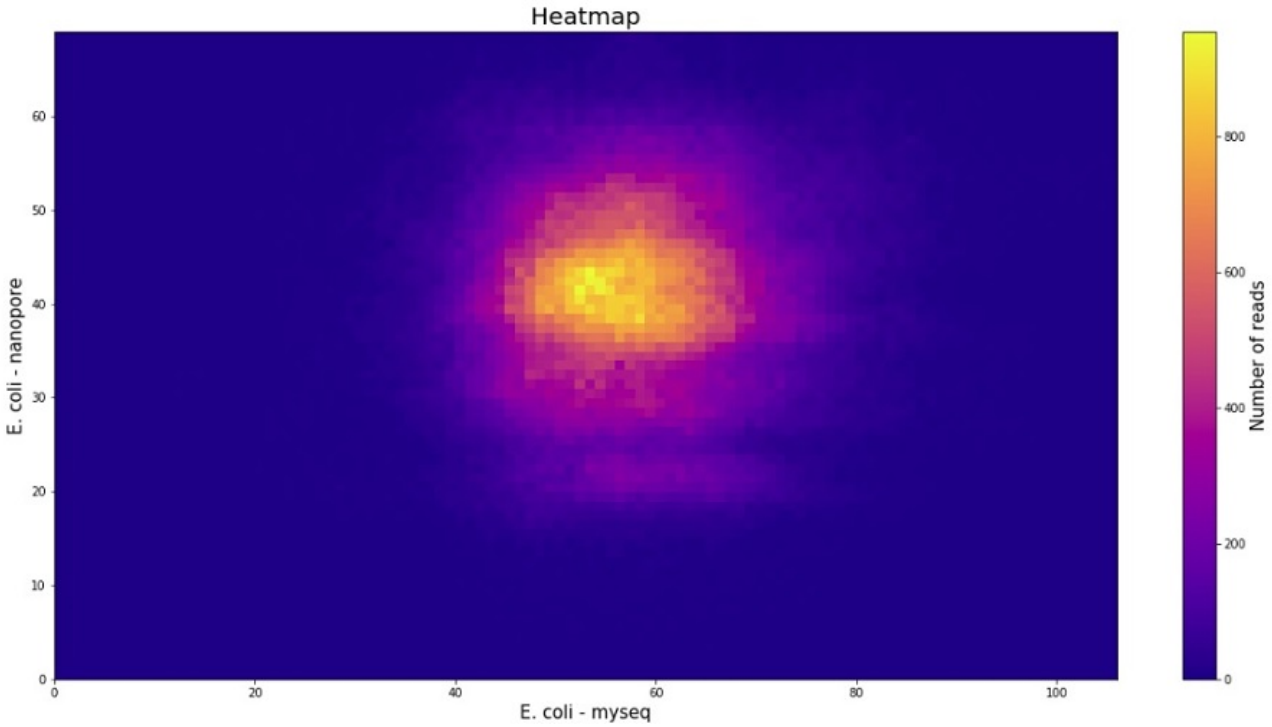
Najprv som prerobila vstupný súbor do vhodného formátu, aby som mohla využiť vykreslovaciu funkciu zo zimného semestra. Použili sme distribúcie: $Gauss$, $Poisson$, $Binomial$, $NegativeBinomial$.
Na nasledujucich obrázkoch vidíme fitovanie týchto distribúcií na dvoch *bedGraphoch* z genomických údajov baktérie *E.coli*.





3.3 Ďalšie vizualizácie *bedGraphov*

Dáta som tiež upravila, aby sme ich vedeli zobraziť *2D histogramom* (heatmap-om). Takto sme mali pre každú pozíciu dve hodnoty, ktoré ju opisujú.
Na obrázkoch vidíme údaje z predošlých dvoch *bedGraphoch* z genomických údajov baktérie *E.coli*.

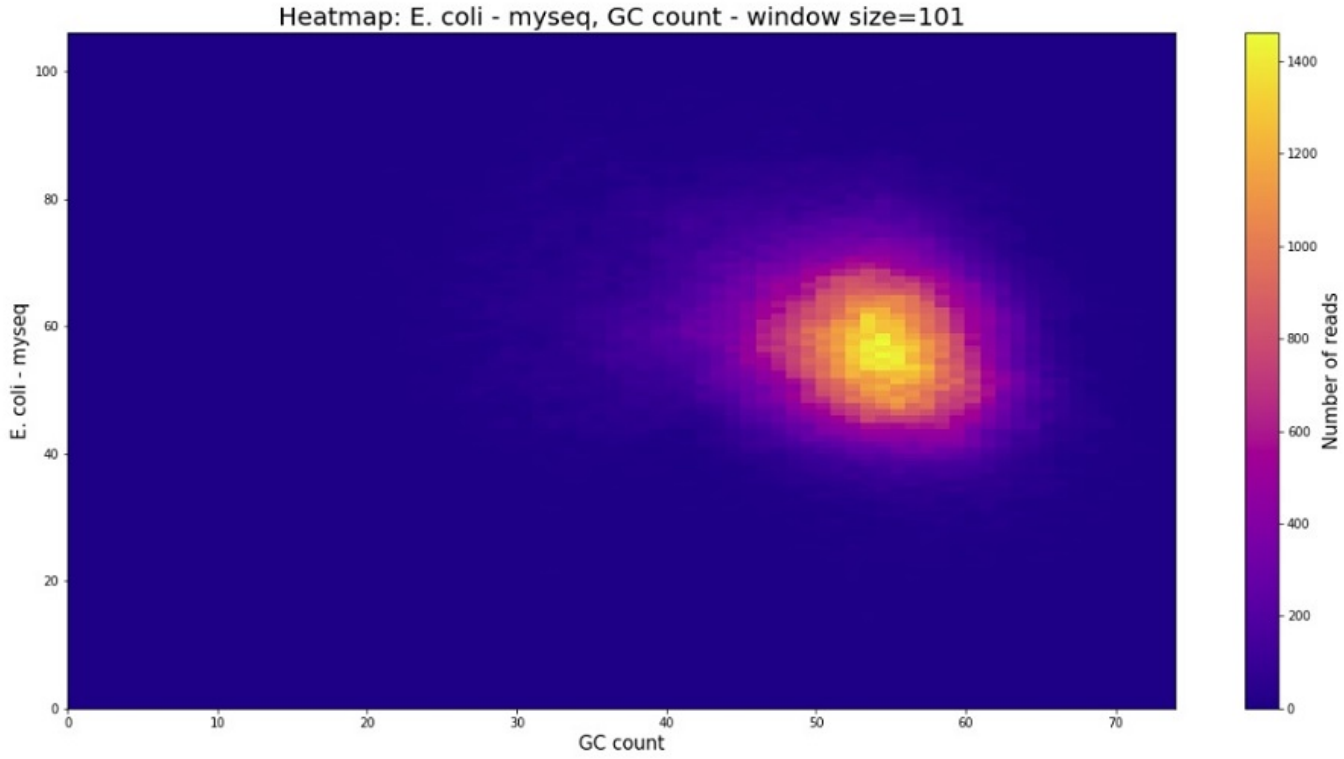


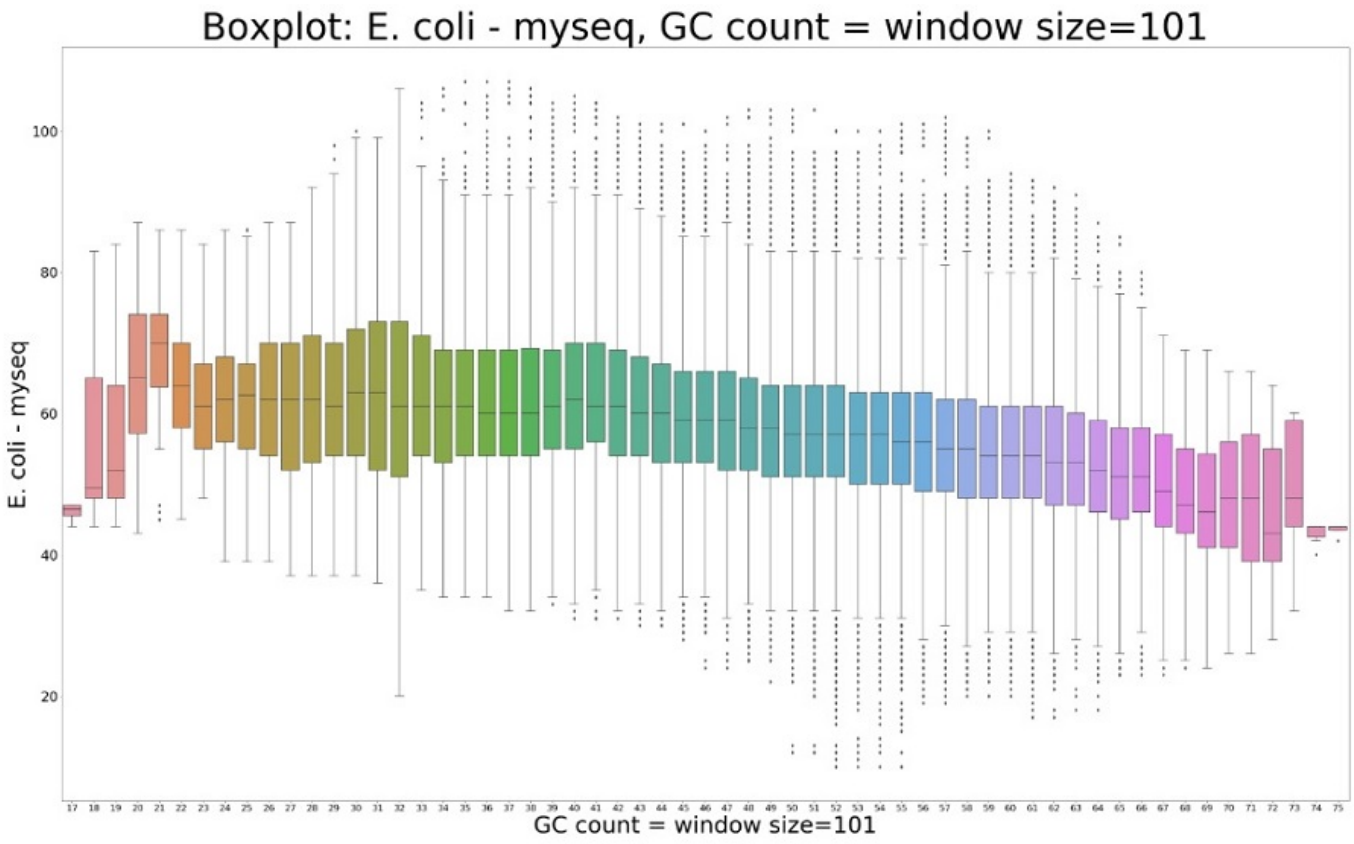
3.4 Vizualizácie podľa rôznych charakteristík genómu

Neskôr sekvenciu nukleotidov sme spracovali zo súborov, ktoré boli v textovom formáte *FASTA*. Takto sme mohli počítať aj rôzne charakteristiky genómu. Tzv. oknové štatistiky sme mohli použiť ako hodnoty pre pozície v genóme. Tieto štatistiky si môžeme predstaviť tak, že máme okno - podreťazec - nejakej fixnej nepárnej dĺžky. Týmto oknom sme prechádzali (posúvali vždy o pozíciu okno) sekvenciu z *FASTA* súboru a vždy aplikovali nejakú fuknkciu na to okno. Tieto funkčné hodnoty sme si uložili pre každú jednu pozíciu, a potom zobrazili *2D histogramom* a *boxplotom*.

3.4.1 Vizualizácia počtu GC v genóme E.coli

Jedna takáto jednoduchá charakteristika je výpočet počtu G, C. Na obrázku vidíme túto štatistiku pre E.coli s oknom veľkosti 101.




Boxplot: E. coli - myseq, GC count = window size=101
E. coli - myseq
GC count = window size=101

3.4.2 Vizualizácia entropie v genóme E.coli

Ďalšia charakteristika bola *entropia* - miera náhodnosti. Na obrázku vidíme túto štatistiku pre E.coli s oknom veľkosti 101.