Spis treści

[I. Wstęp 2](#_Toc128342665)

[1. Podział dokumentacji 2](#_Toc128342666)

[2. Organizacja pracy 2](#_Toc128342667)

[3. Proces developmentu 2](#_Toc128342668)

[4. Ogólne spostrzeżenia 3](#_Toc128342669)

[II. Funkcjonalnośc aplikacji 4](#_Toc128342670)

[1. Interfejs 4](#_Toc128342671)

[2. Analiza sekwencji 4](#_Toc128342672)

[3. Wizualizacja białka 6](#_Toc128342673)

[4. Wykres właściwości fizycznych białka 9](#_Toc128342674)

[III. Struktura aplikacji 12](#_Toc128342675)

[1. Główna architektura aplikacji 12](#_Toc128342676)

[2. amino\_acid\_interpreter 12](#_Toc128342677)

[3. amino\_acid\_plots 13](#_Toc128342678)

[4. amino\_acid\_visualizer 13](#_Toc128342679)

[5. tests 13](#_Toc128342680)

# I. Wstęp

## 1. Podział dokumentacji

Dokumentacja została podzielona na 4 główne sekcje:

- *Wstęp* – informacje o przebiegu, organizacji oraz indywidualnych odczuciach związanych z pracą nad aplikacją

- *Funkcjonalność aplikacji* – „właściwa dokumentacja” określająca sposób użytkowania aplikacji, wraz z wszelkimi informacjami na temat jej działania

- *Struktura aplikacji* – sekcja opisująca techniczną architekturę aplikacji, poświęcona w głównej mierze opisie autorskich modułów, z których została utworzona

- *Źródła* – artykuły, poradniki, bądź badania, które w szczególny sposób przysłużyły się rozwojowi aplikacji

## 2. Organizacja pracy

Nietypowa tematyka aplikacji - bioinformatyka zmusiła nas do podziału drużyna na 2 obozy. Wśród nich jeden stanowiły osoby zajmujące się poszukiwaniem informacji na temat aminokwasów, następnie segregacją znalezionych danych, przygotowaniem funkcjonalnych przykładów a także edukacją drugiej części drużyny. Ta odpowiedzialna była za utworzenie funkcjonalnego programu w oparciu o zdobytą wiedzę oraz szablony utworzone przez osoby zajmujące się częścią teoretyczną.

## 3. Proces developmentu

Aplikacja zaczęła powstawać od końca listopada, a pracę nad nią trwały aż do końca lutego. Wspomniany okres był szczególnie intensywny na początku grudnia, kiedy to aby odpowiedzieć sobie na pytanie jak coś funkcjonuje, musieliśmy najpierw postawić przed sobą dwa zupełnie nowe pytania. Każdy początkowy krok do przodu wymagał od nas dwóch kroków w tył, co pierwotnie sprawiało błędne wrażenie, że zamiast iść do przodu – cofamy się. Wraz ze zdobyciem informacji, przygotowaniem danych do testów oraz czasochłonną edukacją pozostałych członków drużyny, rozpoczął się proces właściwego powstawania aplikacji – wreszcie pojawiły się pierwsze linijki kodu.

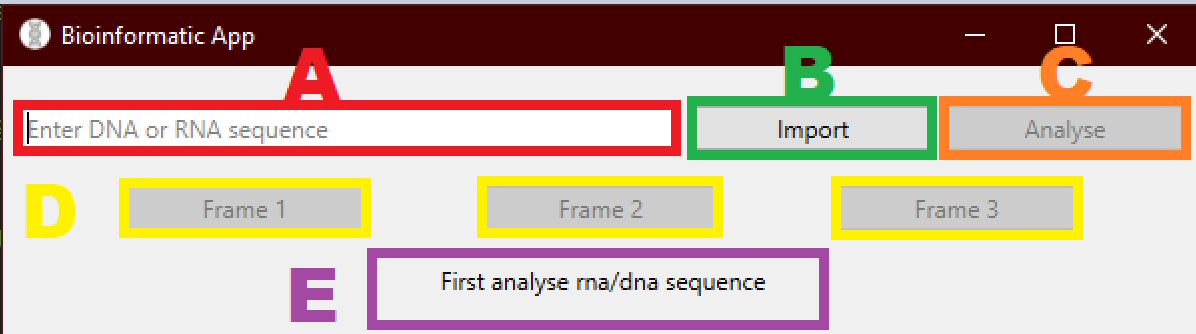
Do napisania aplikacji zdecydowaliśmy się posłużyć językiem Python ze względu na jego wszechstronność oraz satysfakcję z pracy. Wadą tego rozwiązania okazała się być wydajność, lecz istotna była dla nas przede wszystkim zabawa. Zdecydowaliśmy jej sobie nie odbierać i użyć języka, w którym każdy się odnajduje. Pełny kształt aplikacja osiągnęła na początku lutego, lecz ilość rzeczy do poprawy, bądź przebudowy wciąż była przytłaczająco duża. Mimo narastającej presji czasu pod koniec lutego aplikacja osiągnęła swój finalny kształt oraz została zabezpieczona przed błędami.

## 4. Ogólne spostrzeżenia

Wszyscy członkowie drużyny musieli zmierzyć się z ogromnym wyzwaniem, aby zająć się tematem zupełnie dla nas obcym. Zdecydowanie dobrym pomysłem był podział drużyny na dwie podgrupy, gdyż część zawodników naszej drużyny ma naturalną zdolność do przyswajania nowej wiedzy na wszelkie tematy, podczas gdy dla pozostałych temat biologii i chemii to obcy i dość przerażający świat. To rozwiązanie pomogło przydzielić każdemu zajęcie, które sprawia mu najmniej trudności i daje siłę do dalszej pracy. Ostatecznie każdy czuje zadowolenie z wykonanej aplikacji, ponieważ wyzwanie którego się podjęliśmy nie należało do najprostszych. Mówiąc szczerze mogliśmy wybrać coś innego, co wykonalibyśmy po najprostszej linii oporu, jednak to by nas tak nie usatysfakcjonowało. Koniec końcu stworzyliśmy coś z czego jesteśmy niesamowicie dumni, niezależnie czy przyniesie nam to przepustkę do trzeciego etapu czy też nie.

# II. Funkcjonalnośc aplikacji

## 1. Interfejs



***A*** – input służący do wczytania sekwencji DNA, bądź RNA

***B*** – przycisk do importu sekwencji DNA lub RNA z pliku tekstowego, wczytana sekwencja pojawia się w inpucie A

***C*** – przycisk do wykonania analizy podanego DNA lub RNA. Jest on zablokowany, aż do momentu gdy w inpucie A znajdzie się całkowicie poprawna sekwencja DNA lub RNA

***D***– przyciski wyboru ramek odczytu(Frame 1: +1; Frame 2: +2; Frame 3: +3). Stają się one aktywne dopiero po przeanalizowaniu sekwencji za pomocą przycisku C

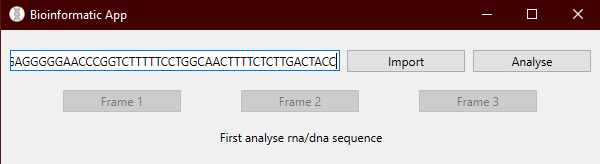
***E*** – białka zawarte w danej ramce odczytu. Treść tej sekcji pojawia się dopiero przeanalizowaniu sekwencji, a następnie wybraniu danej ramki. Blok E może być pusty, gdy w danej ramce odczytu nie udało się znaleźć żadnego białka, bądź zawierać listę przeanalizowanych białek

## 2. Analiza sekwencji

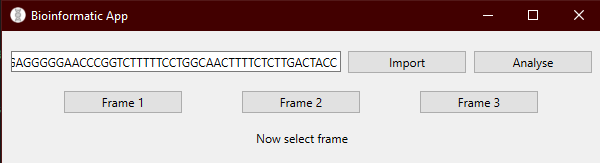
Wprowadź poprawną sekwencję RNA(litery: A,U,C,G) lub DNA(litery: A,T,C,G) ręcznie, bądź wybierz przycisk *„Import”* aby zaimportować sekwencję z pliku tekstowego. Do momentu aż prawidłowa sekwencja nie pojawi się w inpucie - program nie umożliwi wykonywania dalszej części operacji.

*UWAGA!*

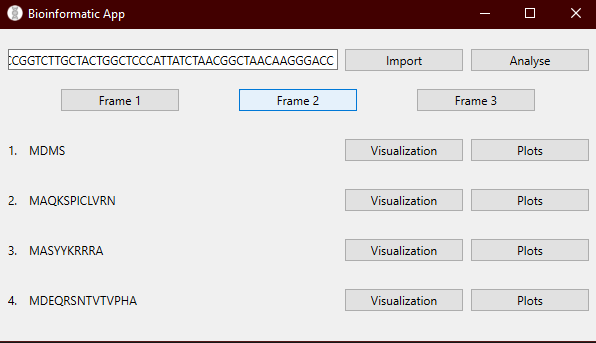
*Wprowadzana sekwencja nie może zawierać żadnych białych znaków(spacji, enterów itp.) ani w inpucie, ani w pliku tekstowym.*



Po przeanalizowaniu białek należy wybrać jedną z trzech ramek odczytu(Frame 1: +1; Frame 2: +2; Frame 3: +3). Ramka odczytu, definiuje od którego nukleotydu podana sekwencja będzie zamieniana na kodony.

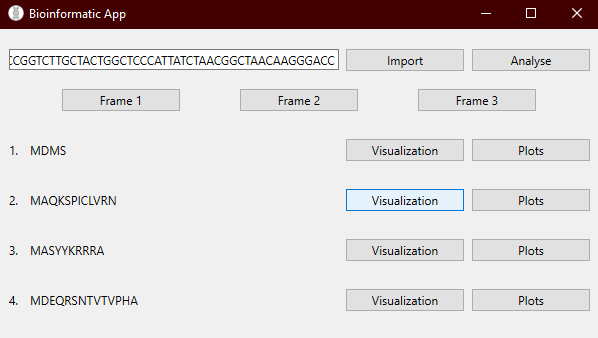


Po wybraniu jednej spośród trzech dostępnych ramek odczytu, pojawiają się zawarte w niej białka. Obok każdego białka będą są dostępne dwa przyciski*: „Visualization”*(wizualizacja białka) oraz *„Plots”(*właściwości fizyczne białka).



## 3. Wizualizacja białka

Po wczytaniu sekwencji RNA lub DNA, a następnie wybraniu ramki odczytu, pojawia się lista białek wraz z dwoma widocznymi przyciskami obok każdego elementu. Aby wyświetlić wzór strukturalny wybranego białka należy kliknąć w podany obok niego przycisk *„Visualization”*.



Po kliknięciu pojawia się load screen, który znika w momencie gdy wizualizacja białka jest gotowa do wyświetlenia.

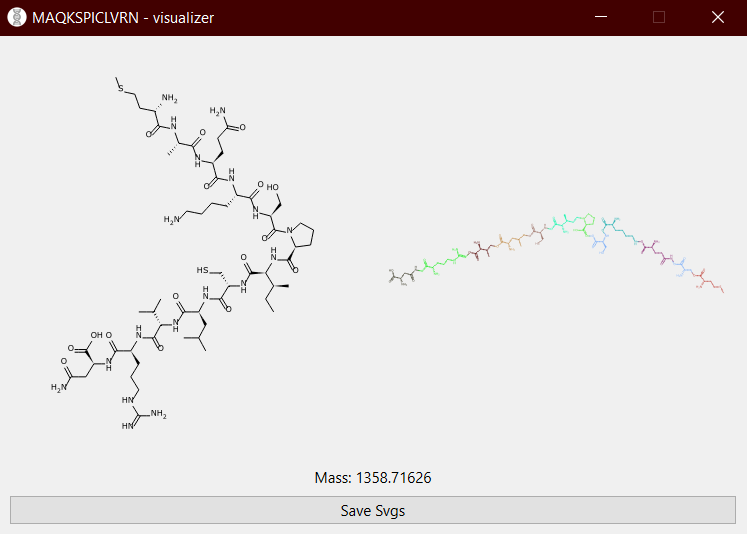
*Uwaga!*

*Im dłuższa sekwencja tym dłuższy czas oczekiwania na wygenerowanie wzoru strukturalnego. W skrajnych przypadkach proces generacji może się wydłużyć nawet do kilku minut.*

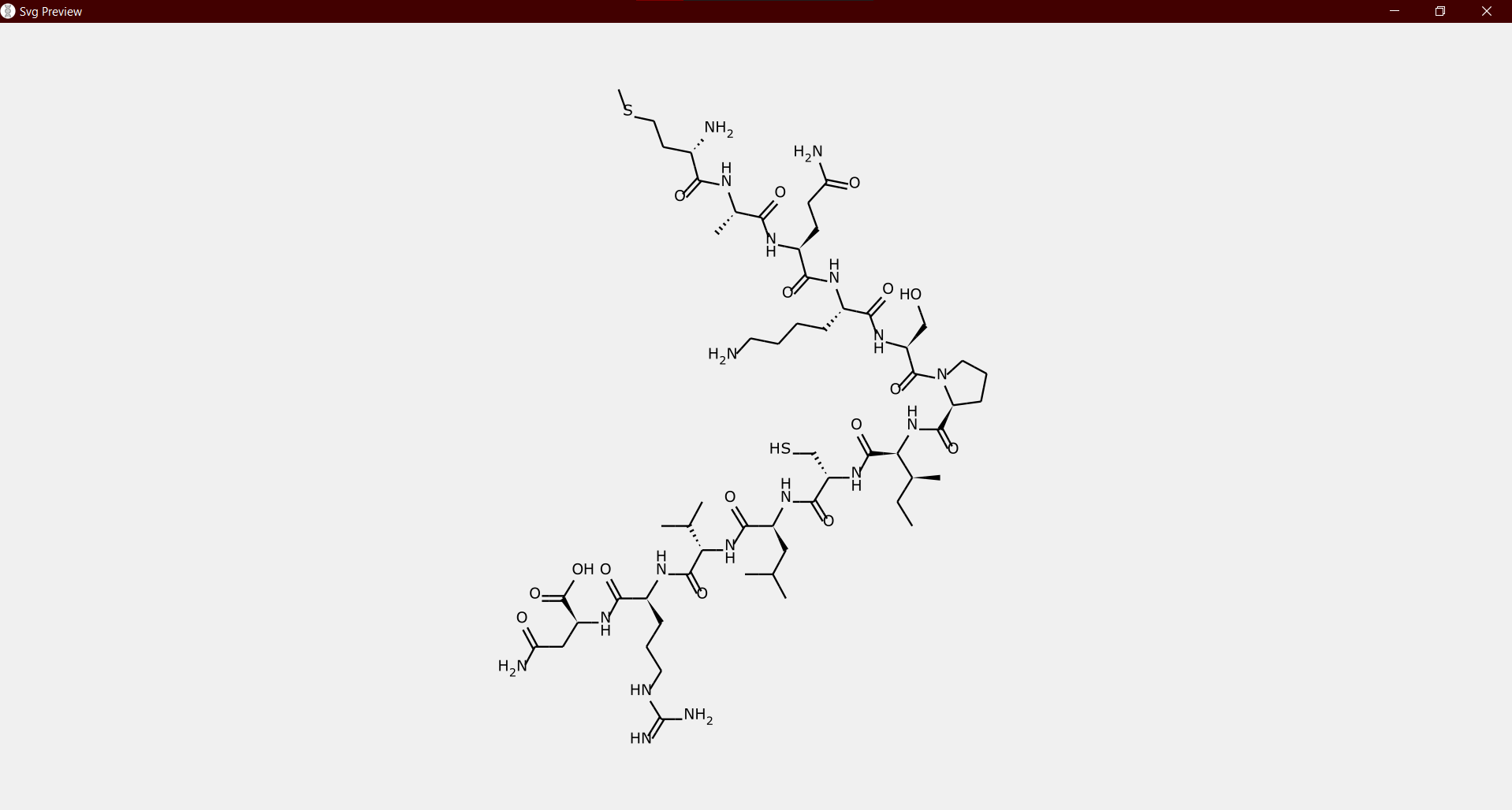


Po wygenerowaniu wizualizacji użytkownik otrzymuje rezultat w postaci dwóch wzorów strukturalnych. *Pierwszy* z nich jest przedstawiony w sposób szablonowy, bez wyszczególnionych aminokwasów, podczas gdy *drugi* umożliwia rozróżnienie konkretnych aminokwasów we wzorze za pomocą kolorów(legenda po kliknięciu we wzór). Aby uzyskać bardziej szczegółowy widok należy kliknąć w jeden z dwóch wzorów.

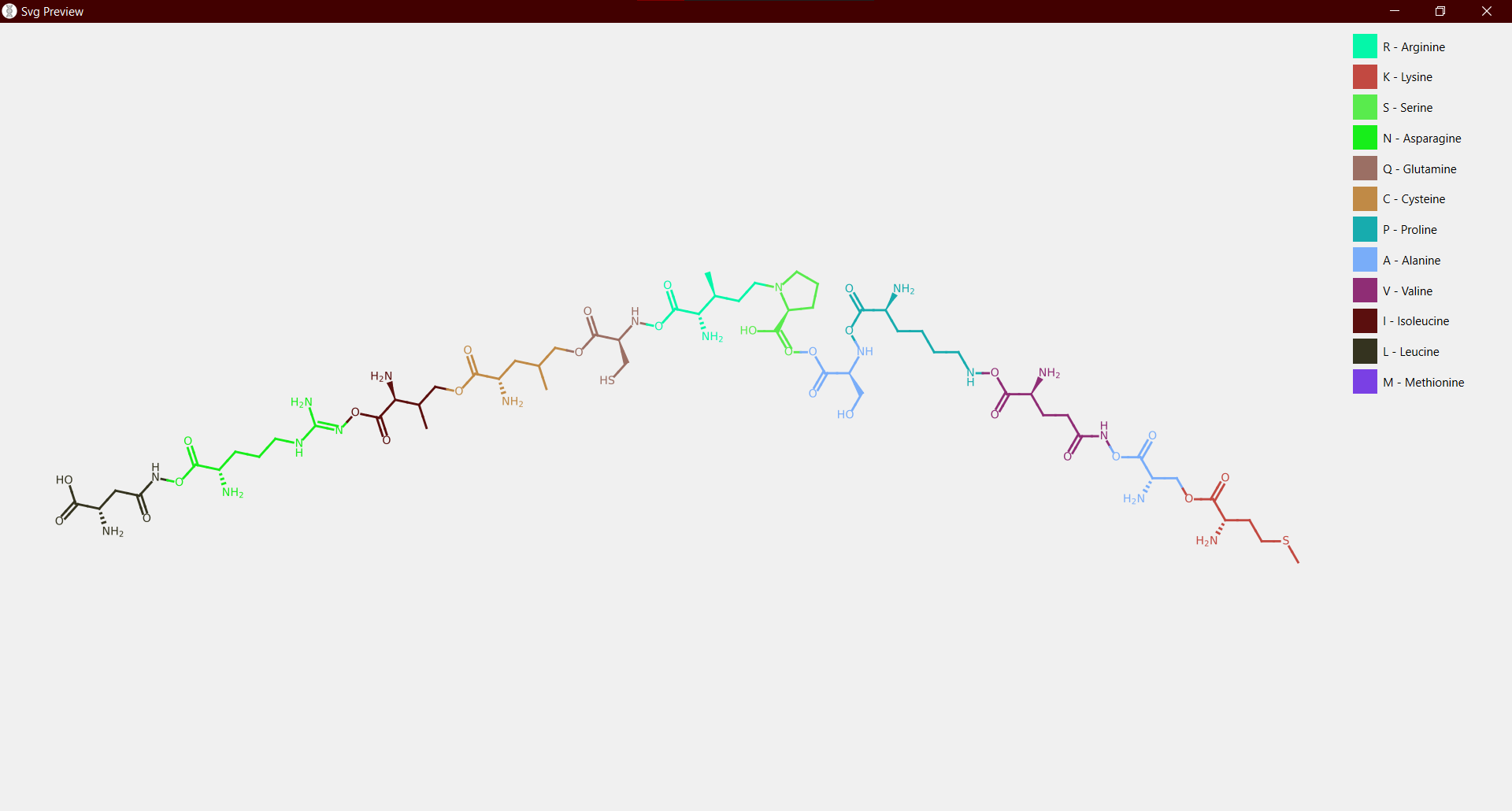
Co więcej tuż nad przyciskiem „Save Svgs” znajduje informacja o masie wybranego białka.



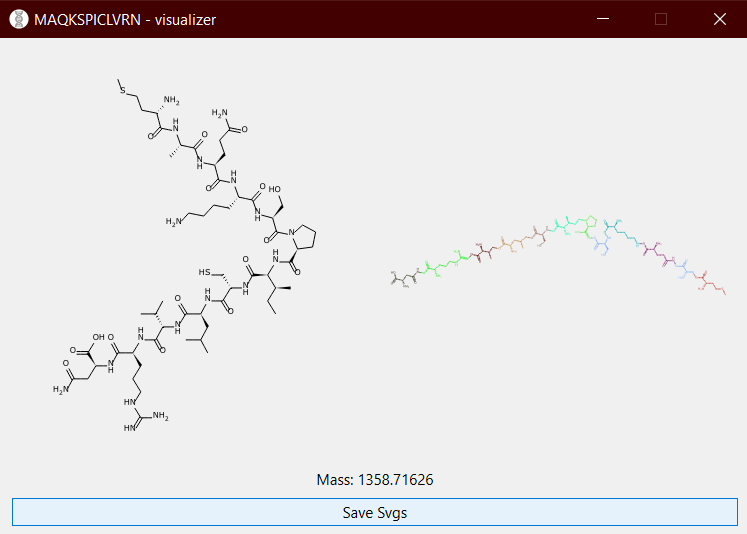
Po kliknięciu w pierwszy wzór strukturalny otwiera się okienko przedstawiające wizualizacje w powiększeniu.



Kliknięcie w drugi spośród wzorów również poskutkuje otwarciem okienka wraz z powiększoną wizualizacją, w tym wypadku w prawym górnym rogu widoczna jest legenda, przypisującą kolory konkretnym aminokwasom.

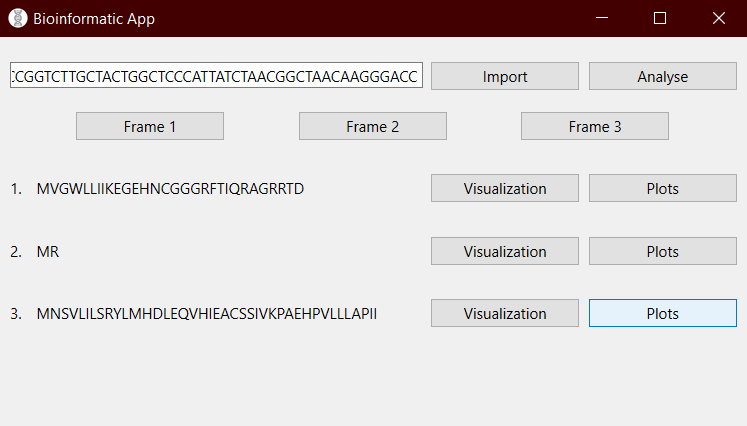


Użytkownik ma także możliwość zapisu wygenerowanych wzorów strukturalnych w postaci plików SVG(zawierających kompresję bezstratną). Aby zapisać wzory należy kliknąć w przycisk „Save Svgs” głównego okienka wizualizacji.



## 4. Wykres właściwości fizycznych białka

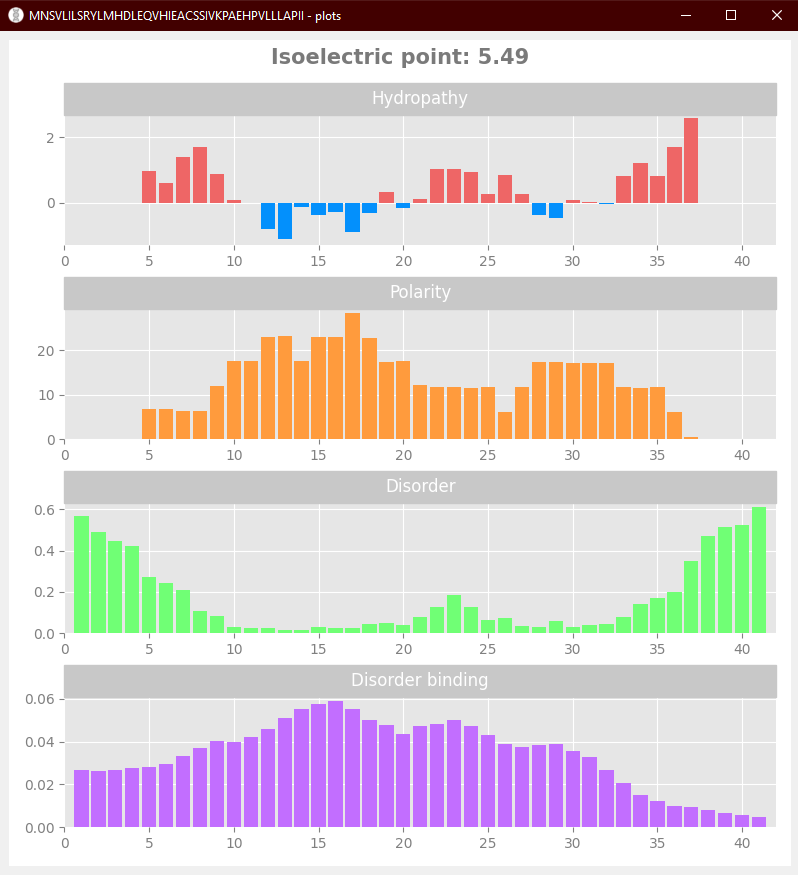
W celu analizy właściwości fizycznych białka, należy uprzednio podać poprawną sekwencję RNA lub DNA, następnie ją przeanalizować i wybrać jedną z pośród trzech ramek odczytu. Następnie obok każdego białka w wybranej ramce odczytu pokaże się przycisk „Plots”, który należy kliknąć w celu uzyskania konkretnej analizy właściwości fizycznych białka.



Po kliknięciu przycisku pojawi się gotowa analiza właściwości fizycznych białka. Na osi pionowej Y znajdują się wartości, z kolei na osi X pozycje aminokwasów w sekwencji. Nad wykresami znajduje się również informacja o wartości punktu izoelektrycznego.

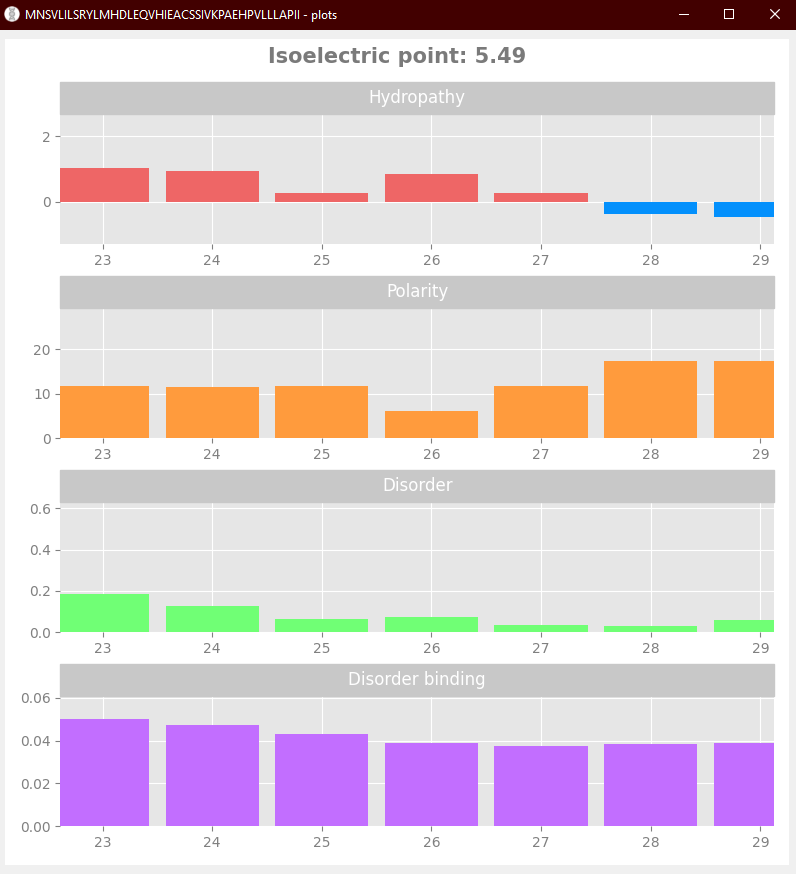
*Uwaga!*

*Niezwykle krótkie białka(składające się z mniej niż 6 aminokwasów) mogą uniemożliwić jego pełną analizę, dlatego niektóre właściwości nie zostaną obliczone.*



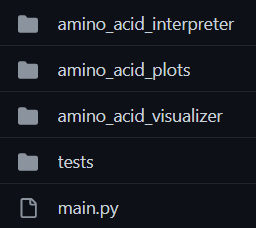
Przedstawiona analiza umożliwia użytkownikowi interakcję. W celu przybliżenia wykresów należy posłużyć się scroll’em myszki.

Ponadto poza przybliżeniem wykresu użytkownik może również przesuwać wykres, w celu skupienia się na interesujących go fragmentach sekwencji. Aby przesunąć wykres należy kliknąć lewy przycisk myszki a następnie przesunąć go w daną stronę(kierunek przesuwania zgodny jest z metodą drag).



# III. Struktura aplikacji

## 1. Główna architektura aplikacji



Aplikacja została podzielona na trzy główne moduły wykonawcze oraz jeden moduł testowy. Każdy z konkretnych modułów odpowiada za inne funkcje:

- *amino\_acid\_interpreter* – analiza wprowadzonej sekwencji RNA lub DNA oraz zwrócenie białek zawartych w podanej sekwencji, wraz z rozróżnieniem na konkretne ramki odczytu

- *amino\_acid\_plots* – algorytmy obliczające wartości właściwości fizycznych dla wybranego białka wraz z architekturą i implemetacją okienka przedstawiającego wyliczone wartości

- *amino\_acid\_visualizer* – generowanie wzorów strukturalnych, a także masy podanego białka, wraz z architekturą okienka oraz jego implementacją

- *tests* – testy jednostkowe sprawdzające poprawność zaimplementowanej funkcjonalności

- *main.py* – główny plik definiujący interfejs aplikacji

## 2. amino\_acid\_interpreter



Moduł odpowiedzialny za analizę otrzymanej sekwencji DNA lub RNA. Zaimplementowano w nim autorski algorytm interpretujący sekwencję w trzech ramkach odczytu, a także zabezpieczający przed wczytaniem niepoprawnych znaków w sekwencji. Jako rezultat zwracany jest słownik zawierający białka w konkretnych ramkach odczytu.

TECH STACK: Python

## 3. amino\_acid\_plots

****

Moduł implementujący interfejs umożliwiający wgląd w konkretne właściwości fizyczne białka, jak również interakcję z uzyskanymi wynikami(przybliżenie i przesuwanie wykresów). Ponadto zawiera on algorytmy obliczające właściwości fizyczne dla danego białka. Wyszczególniony moduł podzielony jest na dwa podmoduły:

- *analysis\_tools* – zawierający definicję algorytmów obliczających właściwości fizyczne

- *plots\_tools* – zawierający definicję algorytmów umożliwiających interakcje z wykresem

TECH STACK: Python, PySide6, Matplotlib

## 4. amino\_acid\_visualizer



Moduł generujący wizualizację białka, a także interfejs graficzny umożliwiający wgląd w jego dokładną strukturę. Zawiera on również algorytm obliczającą masę białka. Moduł generuje strukturę jako plik SVG, dzięki czemu użytkownik ma możliwość zapisu go na własnym urządzeniu.

TECH STACK: Python, PySide6, Rdkit, Pickachu, Pickachu-Chem

## 5. tests



Moduł odpowiedzialny za przeprowadzenie testów jednostkowych sprawdzających poprawność wykonywanej aplikacji.

TECH STACK: Unittest