

Metody bioinformatyki

„Mieszaniny DNA” – dokumentacja wstępna

Piotr Jastrzębski
Marcin Nazimek

1 Treść projektu

Przy badaniu DNA dla celów kryminalistycznych określa się warianty w wybranych miejscach genomu (nazywanych markerami). Projekt zakłada napisanie aplikacji, która dla danego profilu mieszaniny (obejmującego wiele markerów) obliczy wszystkie możliwe profile dla dwu osób i dla trzech osób, a następnie określi poprawne profile drugiej osoby (zakładając, że w mieszaninie będą dwie osoby i profil pierwszej osoby jest znany), lub drugiej i trzeciej osoby (zakładając, że w mieszaninie będą trzy osoby)

2 Funkcjonalność do zrealizowania

Projekt polega na stworzeniu programu, w którym zaimplementowany będzie algorytm generacji na podstawie mieszaniny DNA profili genetycznych osób nieznanych w sytuacji, gdy znane są profile niektórych osób, których DNA występuje w mieszaninie.

3 Założenia projektowe

Program realizowany będzie jako aplikacja konsolowa lub prosta aplikacja okienkowa (decyzja zostanie podjęta na etapie implementacji). Jego przebieg składał będzie się z poniższych etapów:

- Wczytanie danych – profil mieszaniny, liczba osób, których DNA jest w niej obecne oraz profile osób znanych będą podane w pliku wejściowym. Sparsowane dane zostaną przekazane do funkcji odpowiedzialnych za generację profilu
- Analiza danych pod kątem poprawności zarówno składniowej jak i logicznej
- Generacja profili – generacja profili osób, których DNA może być obecne w mieszaninie

- Wydrukowanie danych – zwrócenia na ekran lub do pliku możliwych profili osób

4 Algorytm generacji profili

Weźmy pod uwagę przykład kiedy mamy mieszaninę $X \{1, 2, 3, 4\}$ i wiemy, że są w niej zawarte DNA 3 osób. Znamy profil jednej osoby: $X \{1, 2\}$

1. Na podstawie zadanej mieszaniny i profili znanych osób obliczamy różnicę zbiorów: $\{1, 2, 3, 4\} - \{1, 2\} = \{3, 4\}$
2. Wiemy teraz, że wśród profili pozostałych dwóch osób musi znajdować się zarówno 3 i 4. Ich zestawienie musi pasować do jednego z poniższych wzorców:
 $(3,i), (4,j)$
 $(3,4), (i,j)$
 $(i,j), (3,4)$
3. Generujemy wszystkie możliwe pary permutacji:
 $(1,1), (3,4)$
 $(1,1), (3,4)$
 $(1,2), (3,4)$
 $(1,3), (1,4)$
 $(1,3), (2,4)$
 $(1,3), (3,4)$
 $(1,3), (4,4)$
 $(1,4), (2,3)$
 $(1,4), (3,3)$
 $(1,4), (3,4)$
 $(2,2), (3,4)$
 $(2,3), (2,4)$
 $(2,3), (3,4)$
 $(2,3), (4,4)$
 $(2,4), (3,3)$
 $(2,4), (3,4)$
 $(3,3), (3,4)$
 $(3,3), (4,4)$
 $(3,4), (3,4)$
 $(3,4), (4,4)$

Litera X jest tutaj wyznacznikiem mówiącym o typie mieszaniny. W pliku wejściowym możemy zdefiniować dowolną liczbę różnych mieszanin jak

również dowolną liczbę profili osób. Ich parowanie przebiegać będzie na podstawie znalezienia identycznego wyróżnika umieszczonego przed profilem lub mieszaniną. Wynikiem będą zatem listy możliwych permutacji dla każdego typu danych wejściowych.

5 Języki i narzędzia

Aplikacja stworzona zostanie w języku Java w środowisku Eclipse[1]. Ze względu na wysoką przenośność, możliwość szybkiego jej przygotowania oraz bogate wbudowane biblioteki zdecydowaliśmy się na ten wybór.

6 Testowanie

Testy aplikacji zostaną wykonane za pomocą biblioteki JUnit[2]. Przy pomocy testów jednostkowych zapewnimy, że wyniki zwracane przez aplikację są co do założeń zgodne, a *test-driven development* pozwoli na wczesne wykrywanie ewentualnych problemów.

Literatura

- [1] *Eclipse*
<http://www.eclipse.org>
- [2] *JUnit – framework testów jednostkowych*
<http://junit.org>
- [3] *Wikipedia*
http://en.wikipedia.org/wiki/{DNA|DNA_marker|DNA_profiling}