

# Modelowanie ewolucji populacji ze specjacją

Asia Dąbrowska i Ania Szymik

19 kwietnia 2024

Celem naszego projektu, było stworzenie modelu ewolucji populacji uwzględniającego specjację gatunków, który pomógłby w zrozumieniu dynamiki adaptacji organizmów na skutek zmian w środowisku.

## 1 Założenia

Stworzony przez nas model umożliwia przeprowadzenie symulacji ewolucji przez *generations* pokoleń. Podczas pojedynczego pokolenia zachodzą następujące etapy: mutacja, selekcja osobników, reprodukcja, sprawdzenie warunku na pojemność środowiska oraz zmiana w środowisku.

Na początku każdego pokolenia dochodzi do **mutacji** poszczególnych genotypów w populacji. Z prawdopodobieństwem *mutation\_prob* dla pojedynczego osobnika zajdzie mutacja losowo wybranego genu. Wartość, o którą się ta ekspresja zmieni, jest losowana z rozkładu normalnego o odchyleniu standardowym *mutation\_std*.

**Selekcja** polega na obliczaniu wartości dostosowania każdego osobnika,  $\phi_\alpha$ , za pomocą miary odwrotnie proporcjonalnej do odległości od optimum

$$\phi_\alpha = \exp\left(-\frac{\|o - \alpha\|}{2\sigma^2}\right),$$

gdzie:

$o$  – genotyp osobnika,

$\alpha$  – optymalny genotyp

$\sigma$  – odchylenie standardowe dostosowania względem optymalnego genotypu

Następnie z populacji usuwane są osobniki niewystarczająco dostosowane.

**Reprodukcja** przebiega bezpłciowo, poprzez klonowanie genotypu organizmu rodzicielskiego. Organizm po rozmnożeniu się umiera (długość życia organizmów w naszym modelu wynosi jedno pokolenie).

Następnie, jeśli liczba organizmów w populacji przekroczy maksymalną pojemność środowiska – innymi słowy, jeśli jest zbyt dużo osobników w stosunku do dostępnych zasobów – następuje redukcja populacji.

Zakończenie pokolenia następuje w wyniku **zmiany środowiska**, która powoduje zmianę optimum. Dodatkowo, co  $T$  pokoleń zachodzi specjacja – pojawia się nowy genotyp optymalny w danym środowisku.

Parametry opisujące populację:

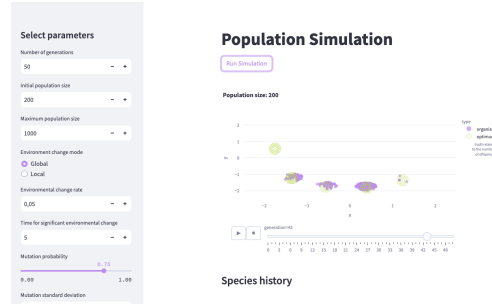
- $N \rightarrow$  liczebność populacji
- $n \rightarrow$  liczba genów pojedynczego osobnika
- *mutation\_prob*  $\rightarrow$  prawdopodobieństwo zajścia mutacji
- *mutation\_std*  $\rightarrow$  odchylenie standardowe rozkładu, z którego losujemy wartość zmiany ekspresji zmutowanego genu

Parametry opisujące środowisko:

- *max\_N*  $\rightarrow$  maksymalna pojemność środowiska
- *env\_change*  $\rightarrow$  stała opisująca ciągłą zmianę optymalnego genotypu
- $T \rightarrow$  okres czasu, po którym zachodzi specjacja - pojawia się nowy genotyp optymalny

- *angle* → kąt, w stopniach, pomiędzy nowo powstałym optimum, a wektorem, z którego powstało
- *fitness\_thr* → minimalne dostosowanie osobnika umożliwiające mu przeżycie
- *fitness\_std* → odchylenie standardowe dostosowania genotypów organizmów względem optymalnego genotypu

## 2 Implementacja



Rysunek 1: Interfejs graficzny aplikacji

W naszej implementacji **populacja** reprezentowana jest jako macierz, której wiersze – wektory liczb – reprezentują genotypy osobników. Każdy osobnik charakteryzowany jest przez dwie liczby (geny), co umożliwia łatwo interpretowalną wizualizację, bez potrzeby redukcji wymiaru, i skraca czas renderowania symulacji. Wszystkie dane, hiperparametry i metody dotyczące populacji przechowywane są w dedykowanej klasie.

Osobną klasę stanowi **optymalny genotyp**, który „pamięta” swoją historię – liczbę osobników będących bliżej niego niż innych optimum w kolejnych pokoleniach.

Przejdźmy teraz do opisu kolejnych kroków symulacji, powtarzających się w każdym pokoleniu:

1. **Mutacja:** Dla każdego osobnika generowana jest liczba z  $Uniform(0, 1)$ . Jeśli jest ona mniejsza od prawdopodobieństwa mutacji (*mutation\_prob*), losujemy indeks genu do mutacji, a następnie dodajemy do niego wartość wylosowaną z  $\mathcal{N}(0, mutation\_std^2)$ .
2. **Selekcja:** Dla każdego osobnika obliczamy maksimum z wartości funkcji fitness w stosunku do każdego z optymalnych genotypów. Następnie przepisujemy do macierzy populacji tylko te osobniki, które przekraczają próg *fitness\_thr*.
3. **Reprodukcja:** Przyjęty przez nas model reprodukcji to klonowanie, a liczba pozostawionego potomstwa zależy od dostosowania osobnika. Zakres jego wartości od *fitness\_thr* (bo osobniki poniżej niego nie przeżyły) dzielimy na *max\_num\_children* + 1 równych przedziałów, odpowiadających liczbie potomstwa od 0 do *max\_num\_children*. Może się zatem zdarzyć, że osobnik przeżyje, ale będzie za słabo dostosowany do środowiska, by pozostawić potomstwo. Po rozmnożeniu, macierzą opisującą populację staje się nowa macierz składająca się wyłącznie z osobników potomnych.
4. **Redukcja wielkości populacji:** Jeśli wielkość populacji przekracza pojemność środowiska, losowo tasujemy wiersze macierzy populacji i zostawiamy tylko pierwsze *max\_N*.
5. **Zmiana środowiska:** W każdym pokoleniu (poza tymi kiedy następuje duża zmiana) każdy z optymalnych genotypów zmienia się o *env\_change* jednostek w ustalonym przez swój wektor kierunku, zaś co *T* pokoleń następuje specjacja – rozejście się losowo wybranego optymalnego genotypu na dwa nowe genotypy poprzez jego rotację w dwóch przeciwnych kierunkach o kąt *angle*.

Aby uczynić naszą symulację bardziej interaktywną, wykorzystaliśmy bibliotekę Plotly. Przy jej pomocy stworzyliśmy animację, którą można odtwarzać, zatrzymywać i przewijać, a także, dzięki suwakowi pod wykresem, niezależnie analizować, co dzieje się z populacją w każdym pokoleniu.

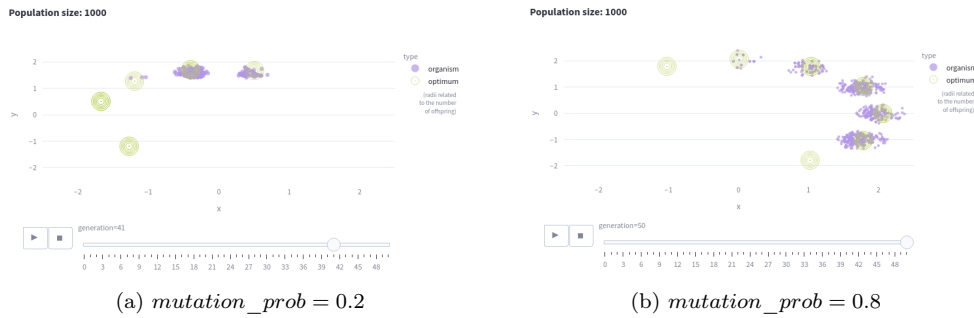
Aby było to możliwe, konieczne jest przechowywanie danych o populacji i optymalnych genotypach w każdym kroku czasowym. W tym celu wykorzystujemy obiekt `pandas.DataFrame`,

w którym przechowujemy współrzędne punktów, a także promienie okręgów reprezentujących odległość od optymalnego genotypu związaną z liczbą potomstwa.

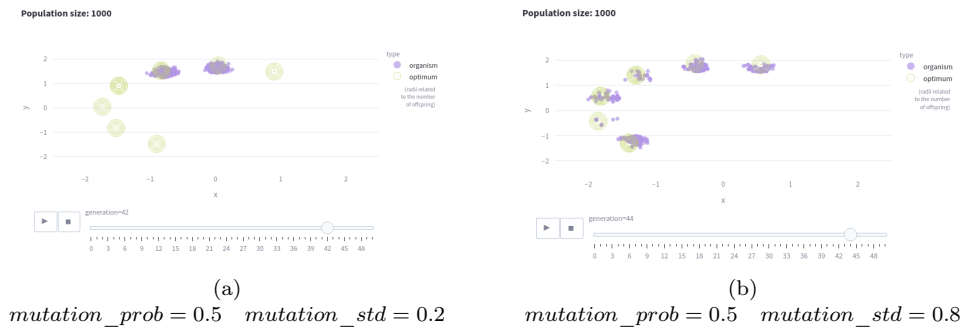
Całość zamknęliśmy w aplikacji stworzonej za pomocą biblioteki Streamlit. Dzięki temu można łatwo zmieniać parametry symulacji i obserwować, jak wpływają one na wyniki, a dodatkowo śledzić historię liczebności gatunków. Interfejs graficzny aplikacji jest przedstawiony na Rys. 1.

### 3 Wyniki

Analizując symulację w zależności od kombinacji parametrów doszliśmy do kilku ciekawych obserwacji. Po pierwsze, nie zawsze przy nowopowstałych optymalnych genotypach mamy jakieś osobniki, a skorelowane jest to z prawdopodobieństwem zajścia mutacji oraz odchyleniem standardowym jej rozkładu (im te dwa parametry są mniejsze, tym częściej występuje taka sytuacja).



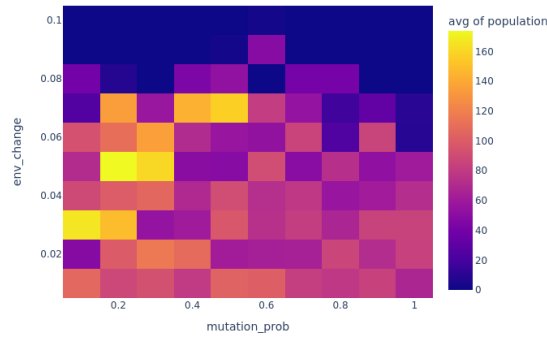
Rysunek 2: Wpływ  $mutation\_prob$  na ewolucję



Rysunek 3: Wpływ  $mutation\_std$  na ewolucję

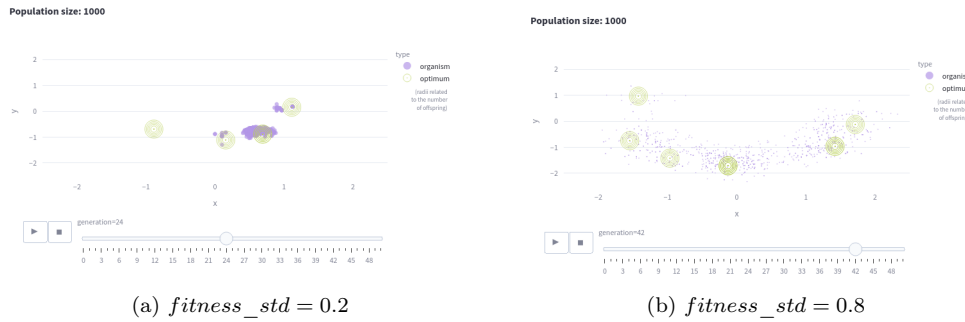
Przechodząc do wniosków niezwiązanych ze specjacją, zauważyliśmy zależność między  $mutation\_prob$  a  $env\_change$  w kontekście liczebności populacji: jeśli prawdopodobieństwo mutacji jest za małe w stosunku do tego, jak szybko zmienia się środowisko, populacja wymiera – np. dla  $mutation\_prob = 0.3$  i  $env\_change = 0.1$ , przy pozostałych parametrach domyślnych, populacja w większości przypadków ginie.

Analizując mapę ciepła przedstawioną na Rys. 4 można zauważyć, że przy dużej wartości  $env\_change$  populacja zaczyna wymierać niezależnie od  $mutation\_prob$ , ponieważ optima ewoluują zbyt szybko, aby populacja mogła za nimi nadążyć. Ponadto można z niej odczytać optymalne pary wartości tych parametrów, w dobrej proporcji do siebie (np.  $mutation\_prob = 0.3$  i  $env\_change = 0.05$ ).



Rysunek 4: Liczebność populacji przypadającej na optimum w zależności od parametrów

Kolejną ciekawą obserwacją jest wpływ  $fitness\_std$  na dynamikę populacji. Im mniejsza wartość tego parametru, tym genotypy osobników skupione są bliżej optimum. Co ciekawe, dzięki temu, że najlepiej dostosowane osobniki pozostawiają najwięcej potomstwa, nawet przy  $fitness\_thr$  równym 0 osobniki i tak skupiają się wokół optymalnych genotypów, jeśli  $fitness\_std$  jest odpowiednio małe. Natomiast  $fitness\_std \rightarrow \infty$  powoduje rozproszenie się genotypów i stworzenie się chmury punktów, zupełnie niezwiązanej z optimumami.



Rysunek 5: Wpływ  $fitness\_std$  na ewolucję

Ponadto, zauważyliśmy, że jeśli redukcja liczebności populacji do  $max\_N$  następowała przy selekcji, przed reprodukcją, liczebność naszej populacji dążyła do  $max\_num\_children \cdot max\_N$ , dlatego zdecydowaliśmy się przenieść ją po reprodukcji.