

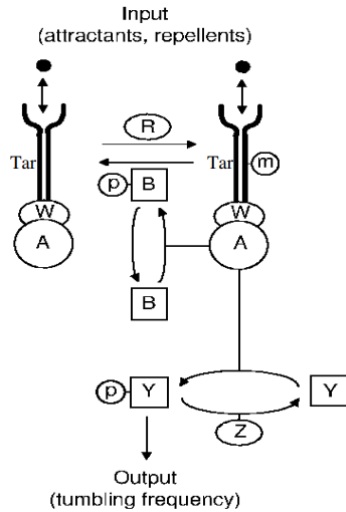
# Chemotaxis szimuláció

Bogye Balázs

2019. Június 27.

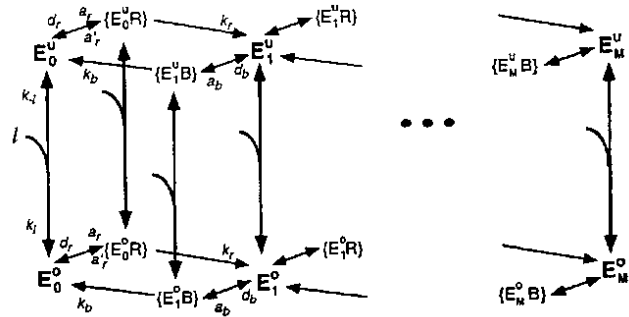
## 1 Bevezető

A bakteriális kemotaxist szimuláltam N. Barkai és S. Leibler modellje[1] alapján. A baktériumok pontszerűen érzékelik a táplálék koncentrációt, mégis tudnak felé mozogni. A ligandum koncentráció növekedésére egy belső "aktivitásuk" lecsökken, aminek hatására az alapvető véletlen bolyongásból kilépve előre kezdenek haladni. Ha sokáig tartózkodnak viszont a nagyobb koncentrációban, akkor megszokják és az aktivitásuk visszaáll.



(a) Chemotaxis network

Forrás: [3]



(b) A receptor komplexum lehetséges állapotai és az ezek közötti átmenetek

Forrás: [1]

A célom a ligandum koncentrációhoz való adaptálódás replikálása (avagy működjön a szimuláció), illetve amennyiben sikerül pár egyszerűbb esetben a baktérium mozgásának szimulálása.

A a reakció rátákat és a kezdeti koncentrációkat [1]-ből vettem.

## 2 Program

A python script és a c++ program elérhető: [4]  
A programok  $\mu\text{M}$  és ms mértékegységben dolgoznak.

### 2.1 Python

A gyors, egyszerű és aránylag letisztult programozhatóság érdekében pythonban írtam meg eredetileg a kódot, egyszerűen el tudtam tárolni a reakciókat és gond nélkül lépésenként végigiteráltam rajtuk, kiszámoltam a kiindulási anyagok koncentrációsorozatát, beszoroztam a reakció rátával és már meg is volt az egyes reakciókhoz tartozó koncentráció változás. Az állapotot sima Euler lépésekkel léptettem. Viszont a gyors egyszerű megoldás nagyon lassú<sup>1</sup> volt.

### 2.2 C++

A sebesség nagyságrendekkel növelése érdekében megírtam pythonban egy c++ kód generátort, ami az ott már megadott reakciókból legenerálta az állapot deriváltjának a függvényét, ezt beraktam egy adaptív Runge-Kutta (4.5) integrátor<sup>2</sup> mögé és készen is van a szimuláció.

## 3 Eredmények

### 3.1 Egyetlen lépéső

Az első és legfontosabb eredmény, hogy az koncentrációhoz való adaptáció létrejön.

---

<sup>1</sup>kb 1:1 idő, egy 20 perces szimuláció 10-20 perc volt

<sup>2</sup>az integrátort  $10^{-6}$ -os lépésenkénti pontossággal futtattam

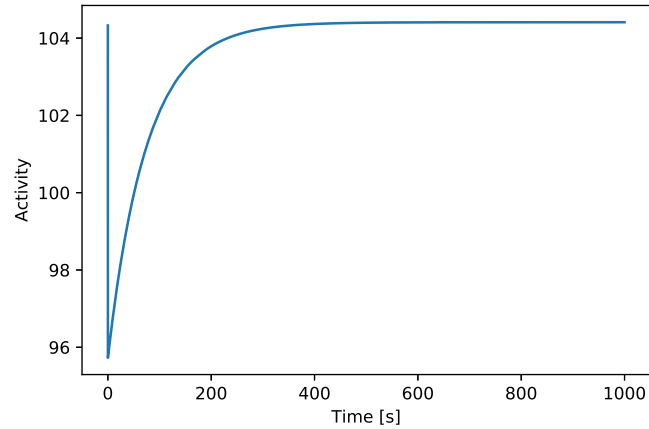


Figure 2: Egyetlen  $1\mu\text{M}$ -os [L] lépcsőre adott aktivitás válasz

### 3.2 Változó mértékű ugrásszerű koncentráció változás

Miután láttam, hogy csinál valami működőképesnek tűnőt a program megpróbáltam reprodukálni [1] 2. ábráját, ami mint látható sikerült.

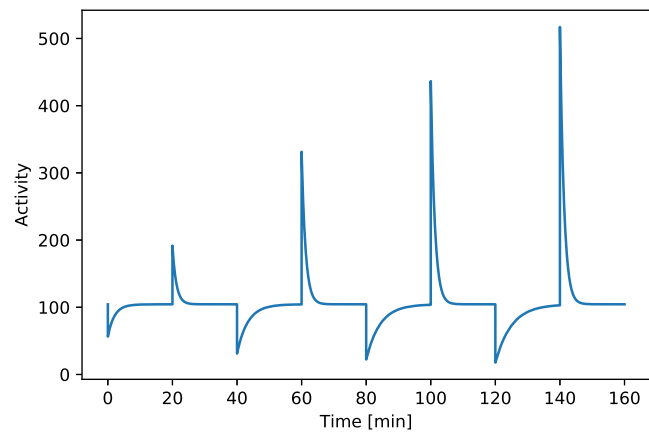


Figure 3: Az aktivitás különböző ligandum koncentrációk a baktérium környezetébe való pillanatszerű hozzáadására illetve kivonására. 20 perces időközönként,  $1, 3, 5, 7 \mu\text{M}$ -os koncentrációkkal, felváltva hozzáadva, majd kivonva

## 4 2D mozgás

A 2D mozgás szimulálásához szükség van az aktivitás függvényében a tumbling frekvenciára ( $f_t$ ), tumbling ( $t_t$ ) illetve run ( $t_r$ ) időtartamra. [2] alapján arra jutottam, hogy aktivitás független  $t_t=0.2$  s időtartamot használok, illetve:

$$f_t(A) = C \frac{A^{2.5}}{B^{2.5} + A^{2.5}}$$

ahol A az aktivitás. A paraméterek választásához úgy döntöttem<sup>3</sup>, hogy:

$$f_t(\infty) = 2$$

$$f_t(A_{[L]=0}) \approx 0.5$$

így:

$$C = 2$$

$$B^{2.5} = 3 \times 10^5 \rightarrow B \approx 155$$

Mivel két tumbling között mindig van egy run, így  $f_t$ ,  $t_t$  és  $t_r$  nem függetlenek:

$$f_t = \frac{1}{t_t + t_r}$$

$$t_r(A) = \frac{1}{f_t(A)} - t_t$$

Innentől létre lehet hozni egy buta szimulációt, ami mindig ekkora adott időtartamokig vagy véletlenül bolyong kis időközönként "ugrálva", vagy egyenesen fut tovább utána a megfelelő  $t_r$  ideig.

Ez egy elég egyszerű modell tényleg, valószínűleg jobb lenne, ha a tumbling és run vége valami valószínűségi eloszlásból lenne húzva mindig (mondjuk exponenciális eloszlással), akkor bele lehetne venni a run közbeni aktivitás változást is figyelembe vegye, nehezebb lenne rossz irányba haladni. Nem beszélve arról, hogy egy nagyon triviális véletlen sétával van helyettesítve a tumbling.

Hogy ne csak vakon bolyongjon a baktérium elhelyeztem egy  $1000 \mu\text{M}$  maximális értékű sugár irányba Gauss görbe alakú,  $x=y=5000$  középpontú ligandum eloszlást. Fontos, hogy kellően nagy gradiens alakuljon ki kellően nagy távon, hogy érzékelhető legyen, de nehogy véletlen egy run alatt túlfuthasson a baktérium. A kezdeti állapotot a 0,0 pozíciónak állítottam, illetve az ottani ligandum koncentrációval szimuláltam a reakciókat egy hosszabb ideig, hogy beálljon az egyensúlyi állapot.

---

<sup>3</sup>Hasraütés alapján

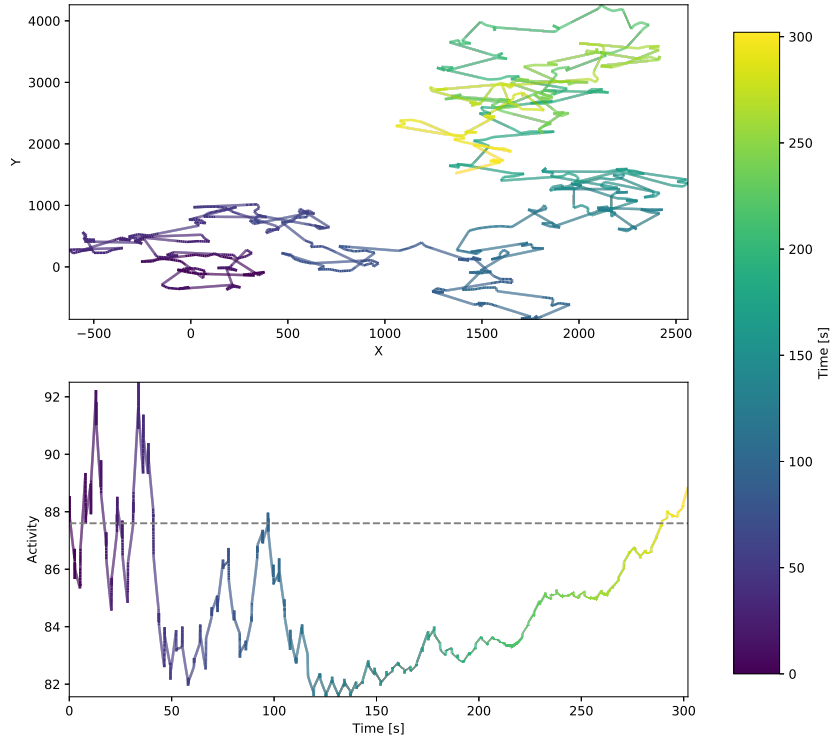


Figure 4: A baktérium szimulált útvonala (5000,5000) középpontú 3000 szórású, 1000  $\mu\text{M}$  csúcsú Gauss görbe ligandum koncentrációban, illetve közben az aktivitás időbeli lefolyása. Az összevethetőség érdekében a szín az időt jelöli. A kezdeti aktivitást szaggatott szürke vonal jelzi. Fontos megjegyezni, hogy a kezdeti aktivitás nem a korábban tapasztalt 110 körüli érték. Utólag vizsgálat alapján a nyugalmi állapot nem volt kellően hosszan szimulálva, így még nem állt be.

A 4. ábrán megfigyelhető, hogy a modell ténylegesen végez egyfajta run és tumbling fázisokat, ám a gradiens irányba haladás nem tökéletes, pedig ez egy válogatottan jó példa. Összességében ennyire leegyszerűsítve a gradiens irányba való haladás nem tűnik túl működőképesnek. Ettől függetlenül szépen látható, hogy az aktivitás ténylegesen csökken, amikor közeledünk a táplálék felé, illetve növekszik, amikor távolodunk tőle. Csupán a mozgás így nem megfelelően reagál, habár érzékelhető, hogy itt már valamivel hosszabbak a runok, viszont a run irányát az előző tumble végiránya határozza meg, így eléggé véletlenszerű.

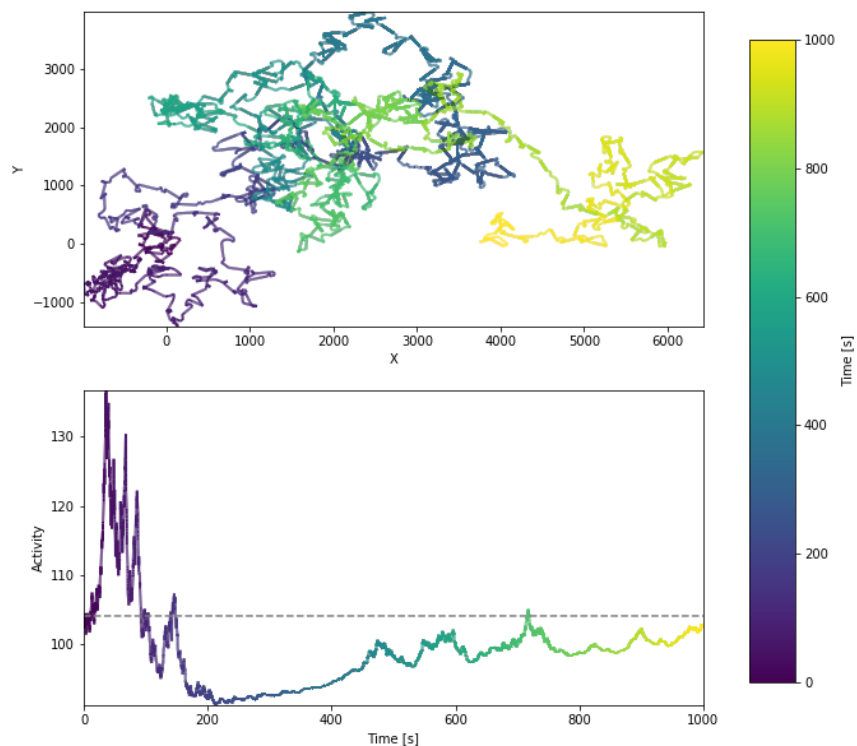


Figure 5: 4. ábrának megfelelően a szimuláció újrafuttatva hosszabb steady-state várakozás után, illetve hosszabb ideig szimulálva

## References

- [1] N. Barkai, S. Leibler, *Robustness in simple biochemical networks*, 1997
- [2] U. Alon, L. Camarena, M. G. Surette, B. Aguera y Arcas, *Response regulator output in bacterial chemotaxis*, 1998
- [3] [http://pearl.elte.hu/andras/sysbio3/2019/40\\_chemotaxis.pdf](http://pearl.elte.hu/andras/sysbio3/2019/40_chemotaxis.pdf)
- [4] <https://github.com/Balazzs/chemotaxis>