

Simulování života mikroorganismů v modelovém mikrosvětě

Filip Majer
Gymnázium Jana Keplera

<https://github.com/nat-int/mikrosim>

dokumentace je nedokončená

Obsah

1. Úvod	2
1.1. Cíle	2
1.2. Související práce	2
2. Model	2
2.1. Prostředí	2
2.2. Proteiny	3
2.2.1. Allosterická místa	5
2.2.2. Reakce	5
2.2.3. Genom	6
2.2.4. Speciální funkce	7
2.3. Buňky	7
3. Implementace	7
3.1. Použité technologie	7
4. Návod k použití	7
4.1. Kompilace	7
4.2. Návod k aplikaci	8
4.2.1. okno „Controls“	9
4.2.2. okno „mikrosim“	9
4.2.3. okno „effect blocks“	10
4.2.4. okno „extra info“	10
4.2.5. okno „cell list“	11
4.2.6. okno „protein view“	11
4.2.7. okno „cell view“	11
4.2.8. zobrazení simulace	11
5. Závěr	11

1. Úvod

Život je plný komplexních systémů, ale pro jeho zkoumání je někdy užitečné mít zjednodušený model, který jsme schopni plně propočítat. Naše schopnosti počítání posunuly vpřed počítače, kterým dnes už můžeme dávat i poměrně komplikované modely.

1.1. Cíle

Cílem tohoto projektu je vytvořit grafickou aplikaci modelu mikroorganismů. Model by měl být takový, aby nebyl moc daleko od středoškolského modelu, jak mikroorganismy fungují, ale zároveň byl dostatečně jednoduchý, aby podle něj šlo najednou simulovat více variant organismů i na běžně výkonných počítačích.

1.2. Související práce

Existuje hodně modelů i jejich simulátorů zjednodušeného života. Velká část z nich funguje tak, že se nadefinuje několik typů buněk, kde má každá své specifické chování, ze kterých se poskládá organismus. Ten se pak může replikovat do stejné nebo podobné buněčné konfigurace. Jedním takovým simulátorem je Biogenesis (<https://biogenesis.sourceforge.net/>).

Jiná velká skupina modelů simuluje organismy jako objekty, které podle definovaných pravidel interagují s okolím a podle vstupů provádí různé akce. Vstupy typicky zpracovávají s využitím neuronových sítí. Příkladem pro tuto skupinu je The bibites (<https://thebibites.itch.io/the-bibites>).

Oba tyto přístupy tak modelují spíše mnohobuněčné organismy než jednobuněčné mikroorganismy.

O něco obecnější model má projekt ALIEN (<https://alien-project.org/>), který je založený na simulaci částic, které mohou tvořit vazby a předávat si signály. ALIEN se snaží o jednoduchý model a z něj vyvstávající chování, a tak v něm život není tak blízko realitě, jako bychom chtěli. Kromě toho je napsaný v CUDA, takže vyžaduje počítač s grafickou kartou od společnosti Nvidia.

2. Model

Model je ve dvou dimenzích, aby bylo snadnější jej implementovat a přehledně vykreslovat.

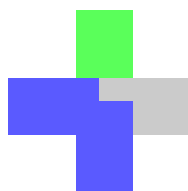
2.1. Prostředí

Mikroorganismy žijí v definovaném prostředí. Jako základ prostředí této simulace byla zvolena tekutina, protože i skutečné mikroorganismy často žijí ve vodě. Standardní způsoby simulování tekutin jsou částicové, eulerovské (kontinuální) a kombinované. V tomto prezentovaném modelu byl použit částicový model z tohoto simulátoru: <https://github.com/SebLague/Fluid-Sim/tree/Episode-01>, především proto, že daný model je snadno implementovatelný, zároveň je pro částicové simulace snažší zabránit tvorbě hmoty z ničeho. Do prezentovaného modelu byl převzat bez korekcí shlukování částic při rychlých ztrátách tlaku, protože chování tekutiny neovlivní moc a jejich vynecháním se model zjednoduší. Snadnost implementace tu je na úkor realismu, tekutina je zde svými vlastnostmi mezi plynem, kapalinou a gelem. Přesto je dostatečně blízko reálné tekutině.

2.1 Model – Prostředí

Tento model ve zkratce funguje tak, že tekutinu rozdělí na částice, kde každá z nich představuje malý objem tekutiny. Částice se zrychlují takovým směrem, aby se hustota (a tím i tlak) blížil požadované hodnotě. Mimo to se rychlosti předávají mezi blízkými částicemi pro efekt viskozity.

Kromě tekutiny jsou v typickém prostředí, kde jednobuněčné mikroorganismy žijí, různé látky, ze kterých získávají energii nebo stavební materiály. Aby se simulace vešla do paměti a zároveň lépe paralelizovala, jsou zde látky modelovány zjednodušeně jako čtveřice podčástí, které jsou červené, zelené, modré a šedé. Látky jsou symetrické vzhledem k rotaci (když se podčásti protočí, tak je to stále stejná látka). Lze je zobrazit jako čtyřcípou hvězdu s cípy obarvenými na jednu ze čtyř barev.



Obrázek 1: „látka“ složená ze zelené, šedé, modré a modré podčástí

Chemické chování látek (tvorba vazeb a vazebná energie) je inspirováno chováním malé organické látky, která obsahuje jeden uhlík se čtyřmi skupinami. Červená podčást se vlastnostmi blíží karboxylové skupině, zelená hydroxylové, modrá jako aminové skupině a šedá neutrálnímu vodíku. Ke každé látce je přiřazena energie (uložená v jejích vazbách), která je určena takto:

$$\sum_i^{\text{podčásti}} E_{z(i)} + c(i) \cdot (f(\text{vedle } i \text{ na jedné straně}) + f(\text{vedle } i \text{ na druhé straně}) + \varphi(\text{naproti } i))$$

E_z dává energii vazby mezi podčástmi a středem látky. f určuje sílu indukčního efektu mezi vedlejšími pozicemi, φ mezi protějšími pozicemi. c dává sílu vlivu indukčního efektu z ostatních podčástí na podčást. Energie látek jsou tak rozmanité a jen občas shodné (ale jsou shodné pro překlopené látky) a můžeme získávat zajímavé výsledky při počítání s jejich rozdíly.

Každá částice (každá kapka tekutiny) v sobě může mít libovolné množství každé látky. V průběhu simulace látky difundují. Při rozptylování dané látky se koncentrace látky z každé částice rovnoměrně rozmístí do všech částic v okruhu se stejným poloměrem, jaký je dosah jejich silových interakcí.

2.2. Proteiny

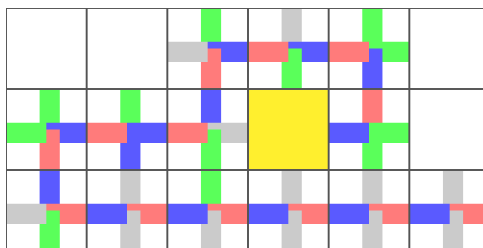
Funkce buněk provádí proteiny. Ty se skládají z aminokyselin, které se zřetěží a složí do tvaru, který jim umožní pracovat. Skládání proteinů je na tuto simulaci moc složité, a tak je zde model opět hodně zjednodušený. V tomto modelu jsou aminokyseliny definovány jako látky vybrané z látek, které obsahují červenou a modrou podčást:



Obrázek 2: látky vybrané jako proteinogení

2.2 Model – Proteiny

Ze seznamu těchto látek pak skládáme protein tak, že za červený konec předchozí látky přidáme další látku tak, aby navazovala modrým koncem (jako translace probíhá od N-terminu k C-terminu za tvorby peptidických vazeb). Protože jsou látky pravoúhlé, skládají se do mřížky. Když existuje více možností, jak by se mohla další látka navázat, vybere se ta, která v řetězci červených a modrých podčástí pokračuje co nejdéle rovně.





Obrázek 3: ukázka proteinu (aa19c6a80100e (více v sekci genom)), žluté pole značí aktivní místo

Ze struktury je určena stabilita proteinu, což je podíl proteinů, která se nerozloží během jednoho kroku simulace:

$$\min(1 - 15\% \cdot e^{-s_c}, 99\%), s_c = \frac{n}{80} + \frac{n_e}{4}$$

kde n je počet polí v mřížce, kde je více látek najednou (protein překrývá sám se sebou - kříží) a n_e je počet míst, kde je více látek najednou a zároveň je vedle prázdného pole. Když se řetězec proteinu překříží, znamená to, že je zabalenější, a tak je náročnější ho rozkládat, takže je stabilnější. Když je překryv vedle prázdného pole, tak musí řetězec postupovat stejnou cestou, a tak se struktura zpevňuje, a tak je o to více stabilnější.

V proteinech se nachází aktivní místa, kde dochází k výměnám podskupin. Jako aktivní místa jsou definována prázdná pole, která mají na všech čtyřech sousedních polích právě jednu látku. Do takového místa se může vázat látka, která je doplňková k podčástem, které do aktivního místa míří. K červené a modré podčásti jsou navzájem doplňkové, zelené a šedé podčásti jsou doplňkové samy sobě. Obrázek 3 obsahuje vyznačené místo, do kterého by se vázala látka . Aktivní místa mohou být maximálně 4 (preferují se ty vlevo a poté nahoře).

Aktivní místa, která jsou maximálně 4 pole od sebe (opět přednostně zleva a shora), se zpárují a katalyzují reakci výměny podčástí mezi látkami. Vyměňují se ty podčásti, které jsou k sobě nejbližší. Zbývá aktivní místa katalyzují reakci výměny vedlejších podčástí vázané látky, těch, které jsou nejbližší středu (těžišti) aktivních míst (přednostně nalevo nahoře). Pro Obrázek 3 bude výsledek reakce .

Takto vytvořený model reakce umožňuje změnu podčástí podle struktury proteinu. Při mutaci může reakce, kterou protein provádí, zachovat, mírně pozměnit i radikálně změnit s nezanedbatelnými pravděpodobnostmi. Je snadné ho implementovat a je nenáročný na výpočet. Také dokud nenastanou krajní případy, tak nezáleží na orientaci proteinu. (existuje ale významný krajní případ - jen jedno aktivní místo).

2.2.1. Allosterická místa

Proteinům by neměla chybět možnost chovat se různě podle látek v jejich prostředí (aby buňky mohly měnit chování podle podnětů). Proto je součástí modelu allosterická katalýza, která umožňuje proteinům obsahovat receptor a měnit podle něj svou efektivitu. Allosterická místa jsou podobná aktivním - jsou to prázdná pole, která vedle sebe mají právě tři pole, která obsahují právě jednu látku. To takového místa se mohou vázat 4 různé látky, které mají tři podčásti doplňkové k těm v proteinu. Katalytický efekt je určen jako $\sin(\text{vzdálenost allosterického od nejbližšího aktivního místa})$. Sinus byl vybrán pro svůj průběh, nemění se moc rychle, ale mění se, aby se při posouvání allosterického místa postupně měnil katalytický efekt.

Podle katalytického efektu se určuje celkový efekt katalyzátorů na protein:

$$K_k = \max\left(0, 1 + \left(\sum_i^{\text{kat.}} \max(f_i(c_k), 0)\right) + \left(\min_i^{\text{kat.}} \min(f_i(c_k), 0)\right)\right)$$

(součet $f_i(c_k)$, ale započítaná je jen nejnižší hodnota). kde c_k je koncentrace katalyzující látky v částici kde protein působí. f_i je funkce toho, jak látka ovlivňuje průběh reakce proteinu podle své koncentrace. podle katalytického efektu (k_e) se f_i přelévá mezi

$$f_{\text{aktivační}}(c) = c^2 \quad (\text{pro } k_e = 1)$$

$$f_{\text{potřebný}}(c) = c^2 - 1 \quad (\text{pro } k_e = 0)$$

$$f_{\text{inhibiční}}(c) = -c \quad (\text{pro } k_e = -1)$$

takto:

$$f_i(c) = \text{lerp}(f_{\text{potřebný}}, f_{\text{aktivační}}, k_e) \quad \text{pro } k_e > 0$$

$$f_i(c) = \text{lerp}(f_{\text{potřebný}}, f_{\text{inhibiční}}, \text{smoothstep}(-k_e)) \quad \text{pro } k_e \leq 0$$

Pro aktivátory je přechod mezi funkcemi lineární, pro inhibitory je přidán smoothstep k interpolačnímu koeficientu, aby se urychlil přechod přes bod, kde f_i vychází přibližně jako záporná konstanta.

2.2.2. Reakce

Když protein působí v nějaké částici, katalyzuje v ní reakci. Předpokládá se, že bez pomoci proteinu běží reakce zanedbatelně.

O reakci jsou známy reaktanty a produkty, takže lze získat i změnu entalpie:

$$\Delta H = \sum_i^{\text{produkty}} E(i) - \sum_i^{\text{reaktanty}} E(i)$$

2.2 Model – Proteiny

kde E je již zmíněná energie látky. Když se zanedbá $\Delta S \cdot T$ (předpokladem pro to je, že se entropie reakcí moc nezmění), tak z

$$\Delta G = \Delta H - \Delta S \cdot T \rightarrow \Delta G = \Delta H$$

$$K = e^{-\frac{\Delta G}{RT}} \rightarrow K = e^{-\frac{\Delta H}{RT}}$$

jde spočítat rovnovážnou konstantu reakce. K té by se měl blížit poměr součinů koncentrací produktů a reaktantů, s rychlostí určenou silou enzymatické katalýzy. Během kroku, kdy protein provádí danou reakci, se spočítá aktuální poměr (K'') a během kroku se změní na

$$K' = \text{lerp}(K'', K, c_r \cdot K_k \cdot c_p)$$



kde c_r je konstanta rychlosti všech reakcí, K_k je efekt katalyzátorů a c_p je koncentrace proteinu. Nechť je δ změna koncentrací v tomto kroku. Pak

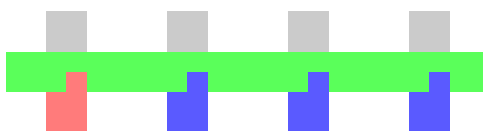
$$K' = \frac{\prod_i^{\text{produkty}} (c_{pi} + \delta)}{\prod_i^{\text{reaktanty}} (c_{ri} - \delta)}$$

$$K' \cdot \left(\prod_i^{\text{reaktanty}} (c_{ri} - \delta) \right) - \prod_i^{\text{produkty}} (c_{pi} + \delta) = 0$$

což je po roznásobení rovnice maximálně čtvrtého stupně v δ (protože maximum jsou 4 aktivní místa) a tak je řešitelná. Z řešení se vybere takové δ , které je v absolutní hodnotě nejmenší a zároveň nedostane žádnou koncentraci do záporné hodnoty. Kdyby mělo přesáhnout do záporu, tak se delta zmenší.

2.2.3. Genom

Jako návod pro sestavení proteinu slouží genom. Ten je tvořen z látek  a , které jsou přes zelené podčásti spojené do řady (v tomto modelu jsou jen 2 báze):



Obrázek 4: ukázka úseku genetického kódu (FTTT/8)

tento genetický kód začíná nalevo. Protože je aminokyselin více, než bází, tak se genetický kód dělí na kodony, které tu mají délku 4, aby mohly kódovat všech 14 zdejších aminokyselin. Genetický kód se přepisuje pomocí F a T, kde F značí červenou bázi a T modrou. Pro přehlednější zápis je i zápis po kodonech. Číslo v hexadecimálním zápisu, tedy číslice nebo malé písmeno z rozsahu a-f značí jeden kodon. Každé číslo převedeme na kodon tak, že do genetického kódu přidáme F v případě lichého a T v případě sudého čísla. Poté číslo vydělíme dvěma a tento postup ještě třikrát zopakujeme. Přiřazení aminokyselin ke kodonům ukazuje okno extra info v aplikaci (v dokumentaci je i jeho obrázek).

4.1 Návod k použití — Kompilace

Vše lze dohromady systému Ubuntu nainstalovat následujícím bash skriptem, ale pravděpodobně je v repozitářích několik z těchto věcí v moc starých verzích:

```
sudo apt install build-essential cmake
sudo apt install vulkan-tools libvulkan-dev spirv-tools
sudo apt install libglfw3-dev libglm-dev libboost-dev

git clone https://github.com/GPUOpen-LibrariesAndSDKs/VulkanMemoryAllocator.git
cd VulkanMemoryAllocator
cmake -S. -Bbuild -DCMAKE_BUILD_TYPE=Release
cmake --build build
sudo cmake --build build -t install
```

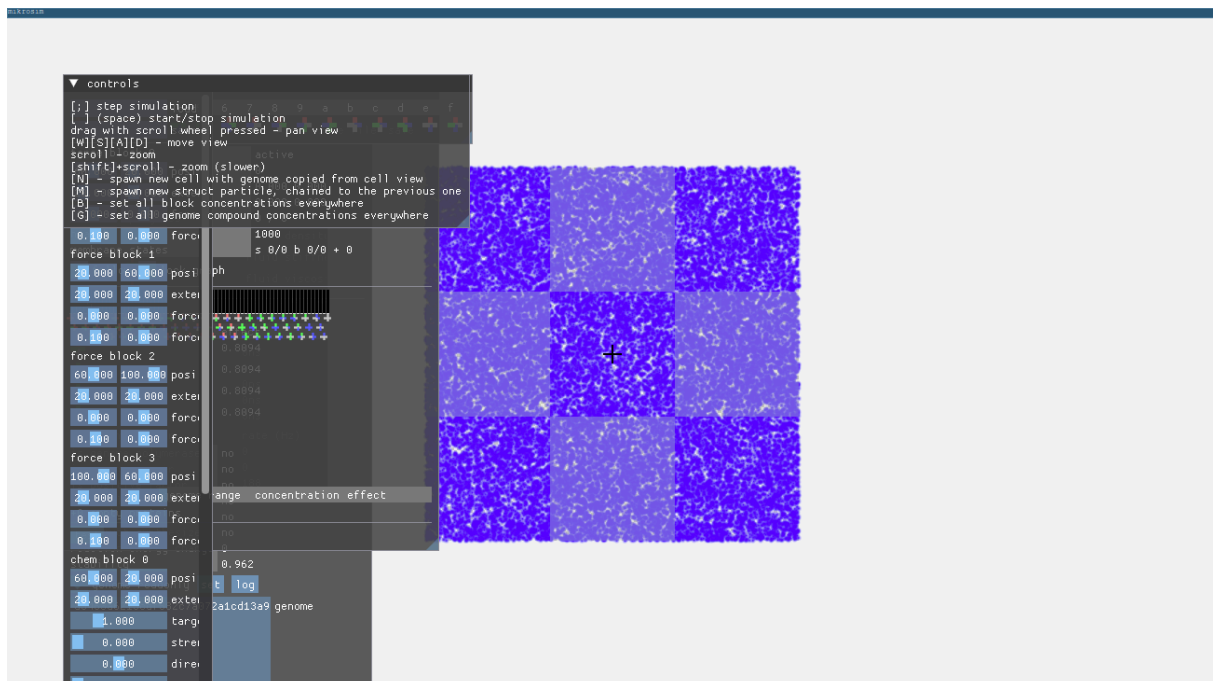
Tento projekt pak jde naklonovat a zkompileovat následujícím bash skriptem:

```
git clone --recurse-submodules https://github.com/nat-int/mikrosim.git
cd mikrosim
cmake -S. -Bbuild -DCMAKE_BUILD_TYPE=Release -DVULKAN_VALIDATION=OFF \
  -DCOMPILER_SHADERS=OFF
cmake --build build
```

Výsledný spustitelný soubor pak je `./out/mikrosim`, případně `./out/mikrosim.exe`, (relativně k cestě po příkazu `cd mikrosim`). Je ho potřeba spouštět ze složky `./out/`.

4.2. Návod k aplikaci

Po prvním spuštění vypadá aplikace přibližně takto:



Obrázek 6: aplikace po spuštění

Obsahuje několik podoken, se kterými se dá levým tlačítkem myši pohybovat a připínat k hlavnímu oknu nebo do sebe navzájem. Mimo podoken je vidět zobrazení simulace (modrý čtverec za okny).

4.2 Návod k použití — Návod k aplikaci

4.2.1. okno „Controls“

```
▼ controls
[;] step simulation
[ ] (space) start/stop simulation
drag with scroll wheel pressed - pan view
[W][S][A][D] - move view
scroll - zoom
[shift]+scroll - zoom (slower)
[N] - spawn new cell with genome copied from cell view
[M] - spawn new struct particle, chained to the previous one
[B] - set all block concentrations everywhere
[G] - set all genome compound concentrations everywhere
```

Okno „Controls“ obsahuje seznam kláves/akcí pro interakci se zobrazením simulace s popisem jejich efektu.

4.2.2. okno „mikrosim“

The 'mikrosim' window displays simulation controls and performance data. At the top, it shows '60 FPS' and a 'stopped' status with a 'start/stop' button. Below are several sliders for parameters like 'steps per frame', 'shown compund', 'set concentrations command target', 'fluid density', 'fluid stiffness', 'fluid viscosity', and 'particle draw size'. A section for 'event' lists various simulation events with their 'timestamp' and 'delta' values. At the bottom, a table shows the 'time' and 'rate (Hz)' for different components of the simulation.

	time	rate (Hz)
sim (cpu)	18us	52834
simu (gpu)	9ms	109
render (gpu)	5ms	185

Okno mikrosim začíná počtem snímků v poslední vteřině, poté text říkající, zda simulace běží a tlačítko, které simulaci spustí, nebo zastaví.

Na druhém řádku je posuvník, na kterém se nastavuje počet kroků simulace, které se provedou mezi každými dvěma snímky (když simulace běží).

Na třetím řádku je posuvník, na kterém se nastavuje látka, která se právě zobrazuje. Pod ním je obrázek dané látky a energie uložená v jejích vazbách.

Další posuvník nastavuje koncentraci, na kterou klávesy G a B nastavují (když se stiskne klávesa B když žádné okno nemá „focus“, tak se koncentrace všech proteinogenních látek nastaví na tuto hodnotu).

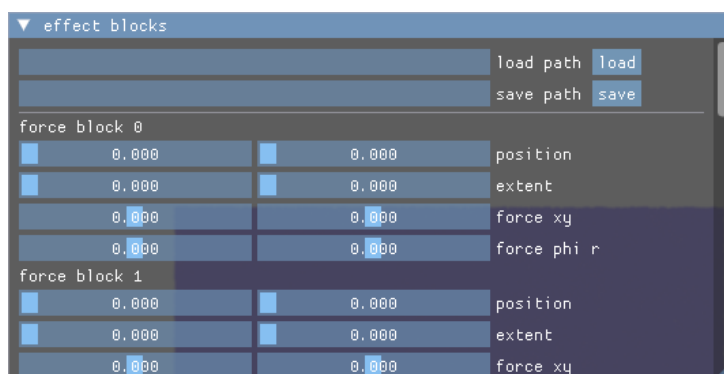
Následující 3 posuvníky nastavují vlastnosti tekutiny - hustotu, které se snaží dosáhnout, koeficient síly vyrovnávání hustoty a viskozitu.

Poslední posuvník nastavuje velikost částic při jejich vykreslování.

Pod posuvníky jsou tabulky s časy jednotlivých částí výpočtů simulace a vykreslování.

4.2 Návod k použití — Návod k aplikaci

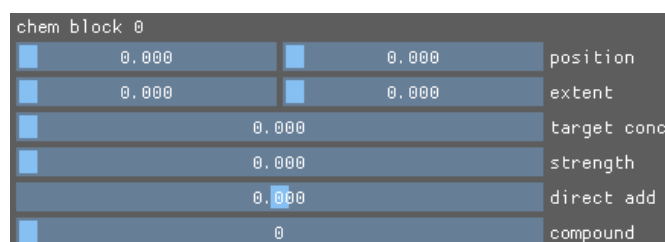
4.2.3. okno „effect blocks“



Toto okno obsahuje nastavení „efektových bloků“, kterými se dá ovlivňovat simulace. Existují jich 2 typy: silové („force blocks“) a chemické („chem blocks“). Silové bloky umožňují působit silou na částice v obdélníkové oblasti simulace. Chemické umožňují ovlivňovat koncentrace nějaké látky v obdélníkové oblasti simulace. Od obou typů jsou dostupné 4 bloky. Bloky se vykreslují přes zobrazení simulace s vysokou průhledností, silové bloky červeně a chemické zeleně.

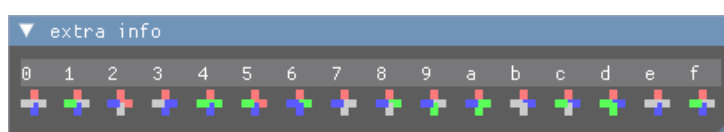
Na začátku jsou dvě textová pole s tlačítky - první umožňuje načíst nastavení všech bloků ze souboru, jehož cesta je napsána v textovém poli. Druhý umožňuje do souboru v textovém poli nastavení všech bloků uložit.

Pak následují nastavení silových bloků. Každé začíná pozicí a velikostí v osách x a y. Na třetím řádku posuvníků je nastavení homogení síly v osách x a y. Na čtvrtém řádku je nastavení síly kolem středu bloku (proti směru hodinových ručiček) a síla ke středu bloku. Síly jsou malé a tak chvilku trvá, než zapůsobí.



Poté následují nastavení chemických bloků. Stejně jako u silových bloků začínají dvěma řádky nastavení pozice a velikosti. Poté následuje cílová koncentrace, ke které se koncentrace přibližuje, poté síla, která určuje rychlost tohoto přibližování. Posuvník „direct add“ udává koncentraci, která se přičítá ve všech částicím v oblasti. Nakonec je posuvník určující látku, kterou blok zrovna ovlivňuje (čísla látek jsou stejná jako v okně „mikrosim“).

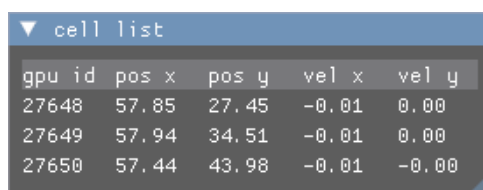
4.2.4. okno „extra info“



Okno extra info obsahuje tabulku pro převod mezi označením v genomu a látkou v proteinu.

4.2 Návod k použití — Návod k aplikaci

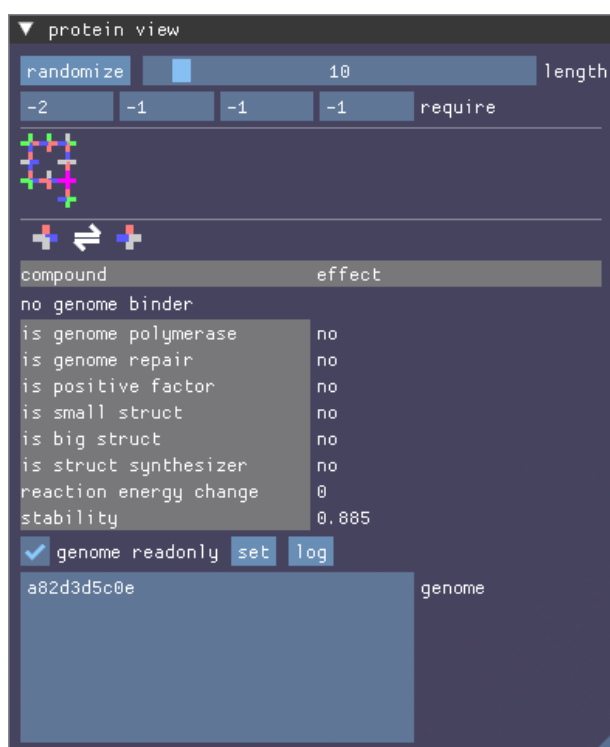
4.2.5. okno „cell list“



gpu id	pos x	pos y	vel x	vel y
27648	57.85	27.45	-0.01	0.00
27649	57.94	34.51	-0.01	0.00
27650	57.44	43.98	-0.01	-0.00


Okno cell list obsahuje seznam všech živých buněk s jejich identifikačním číslem, pozicí a rychlostí.


4.2.6. okno „protein view“



randomize ☐ 10 length

-2 -1 -1 -1 require





compound	effect
no genome binder	
is genome polymerase	no
is genome repair	no
is positive factor	no
is small struct	no
is big struct	no
is struct synthesizer	no
reaction energy change	0
stability	0.885

☒ genome readonly

a82d3d5c0e genome

4.2.7. okno „cell view“

4.2.8. zobrazení simulace

5. Závěr