

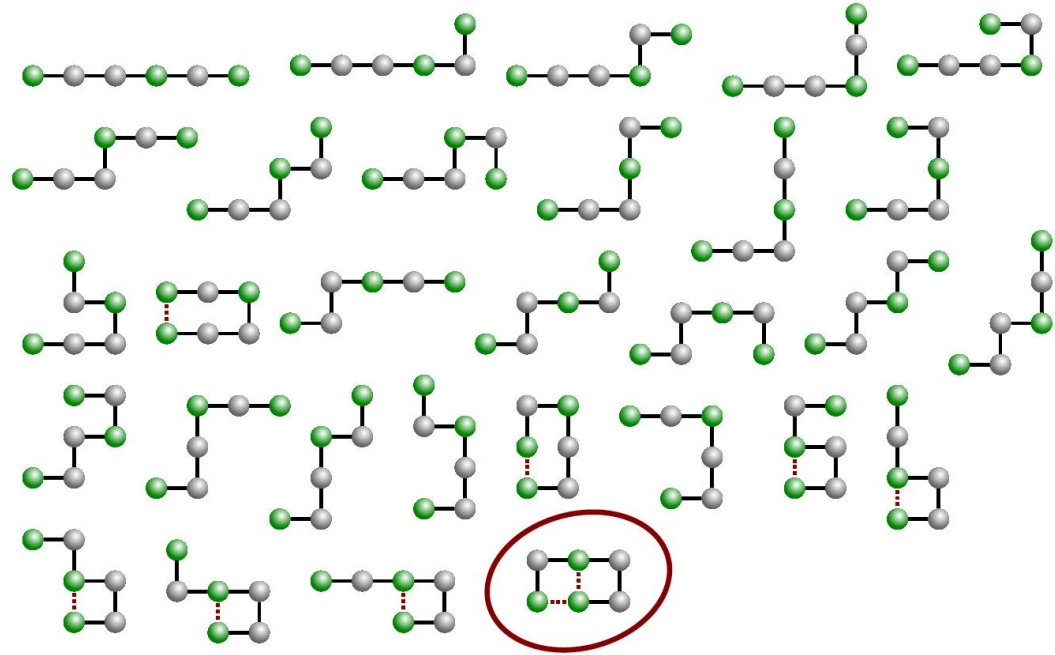
PROTEIN POW(D)ER

TEAM PROTI: JESSE FRANZUA, JOHAN SEMEIJNS,
XAMANIE SEYMONSON



CASE

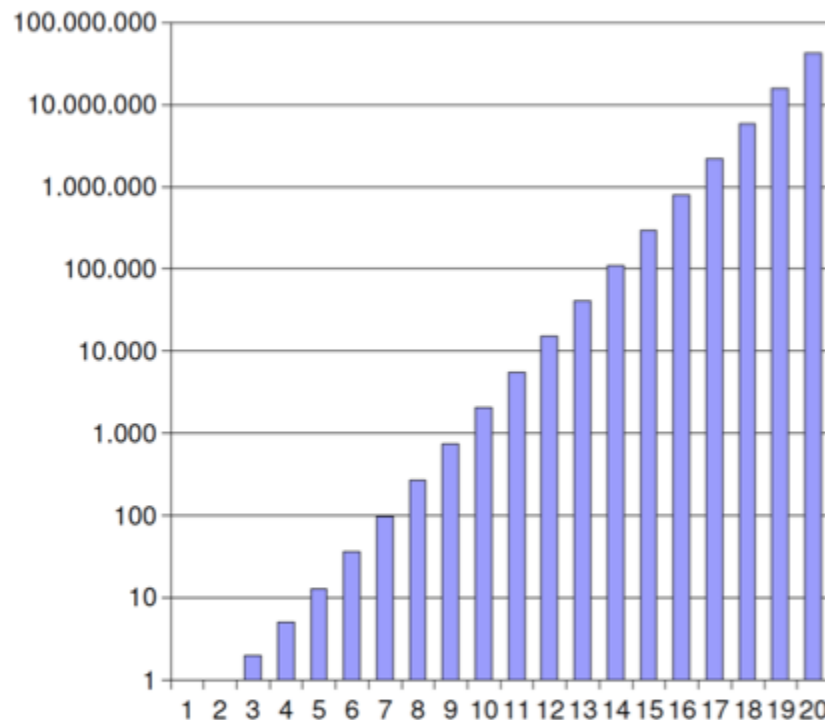
- Start met een eiwit als een string van H, P en later ook C aminos.
- Vouw de protein om zo een stabiele configuratie te vinden.
- Stabiliteit van de configuratie wordt bepaald door een score.
- De score wordt bepaald door H-H, H-C, C-C bruggen die door de vouwing gevormd zijn
 - H-H en H-C bruggen hebben een score -1
 - C-C bruggen hebben een score -5
- Meer bruggen -> lagere score -> stabielere eiwit
- Doel: Vouw het eiwit in de meest stabiele configuratie door de laagste score te vinden.



Hoe groot is de state space?

Zonder constraints in een tweedimensionaal vlak is de state space 4^{n-1} met n de lengte van de string.

De uitdaging: ontwerp een algoritme dat alleen kijkt naar structuren die een goede score kunnen halen.

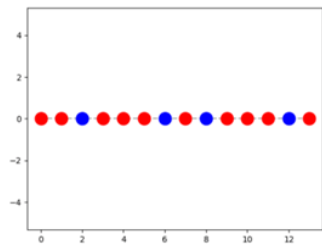


Het aantal -niet symmetrische- valide structuren uitgezet tegen de lengte van de structuur.

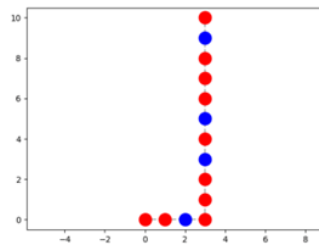
UITGEVOERDE ALGORITMEN

- Monte Carlo (2D + 3D)
- Hill Climber (2D + 3D)
- Simulated Annealing en Simulated Annealing met toegevoegde zogenaamde “re-annealing” (2D + 3D)
- Breadth First Branch and Bound (2D + 3D)
- Tree (2D)
- Firefly (2D)
- Dead-end Elimination (2D)
- Genetic (2D)

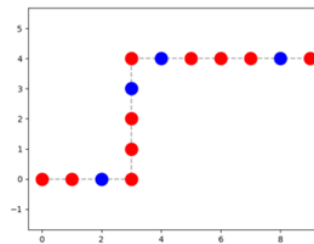
Monte Carlo



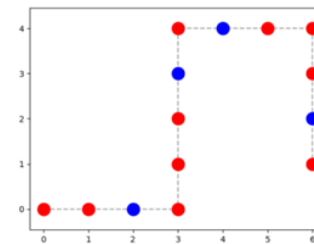
(a)



(b)



(c)



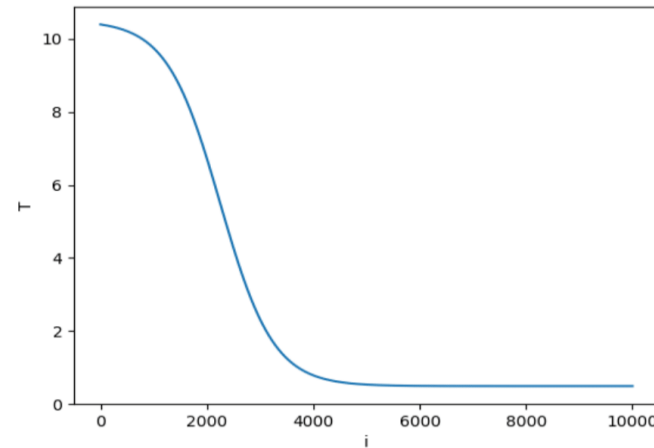
(d)

Start met volledige, ongevouwen structuur (a).

Willekeurige vouwing worden uitgevoerd (b, c, d).

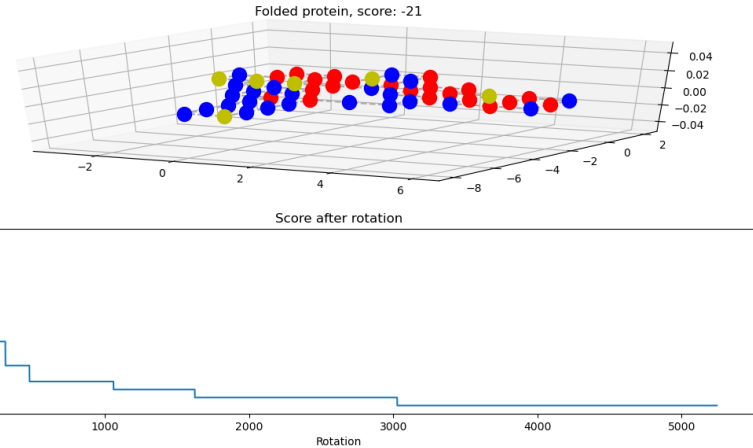
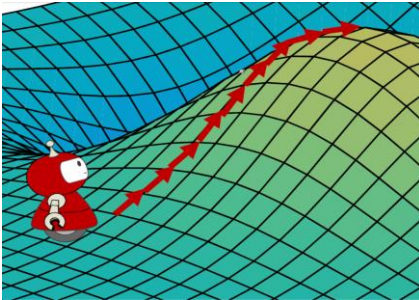
$$p = \exp\left(-\frac{score_{new} - score_{old}}{T}\right)$$

$$T = \frac{10}{1 + \exp\left(\frac{i-9N/40}{N/20}\right)} + 0.5$$



Hill Climber

- Configuratie wordt alleen aangenomen als er een verbetering is
- Bij verslechtering wordt de configuratie terug gezet



Simulated annealing

- Verschil met HC: Configuratie kan ook worden aangenomen bij verslechtering van de scc

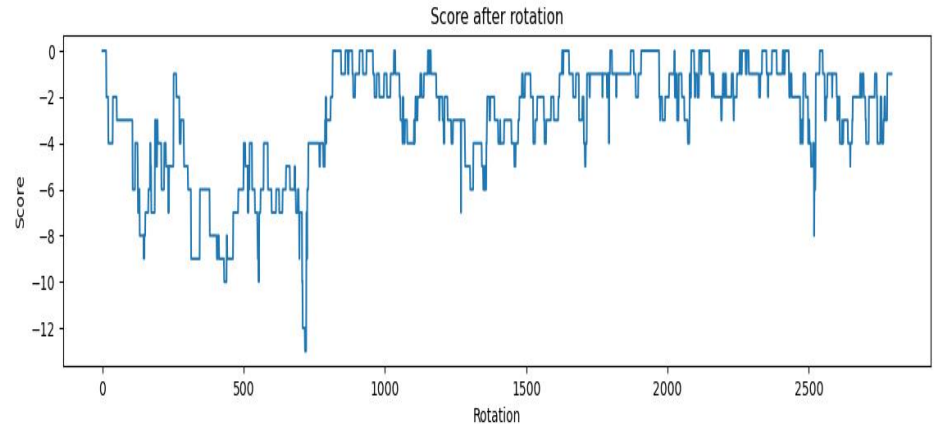
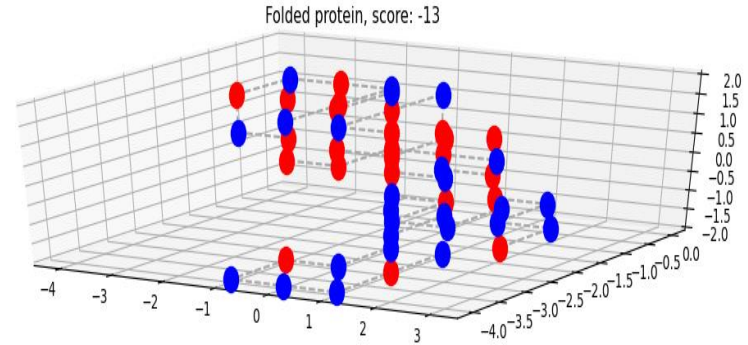
start_temp -> zelf te bepalen waarde

Temperatuur = start_temp * (0.997^{iteratie})

Acceptatie_kans = $2^{-[(\text{oude_score} - \text{nieuwe_score}) / \text{temperatuur}]}$

Treshhold is random waarde tussen 0 en 1

Verslechtering wordt aangenomen als
acceptatie_kans > treshhold



Branch and Bound

Score: score van de partiële conformatie met lengte n .

Gemiddelde score: gemiddelde van de scores van conformaties met lengte n .

Laagste score: laagste score behaald door een conformatie van lengte n .

```
if score < laagste_score:
    aannemen
if score > gemiddelde_score:
    verwijderen(p = 0.99)
if gemiddelde_score > score > laagste_score:
    verwijderen(p = 0.98)
```


Okke's methode

Partiele score: score van gedeeltelijk geplaatste conformatie

Mogelijke score: $-2 \cdot \#H - 10 \cdot \#C$

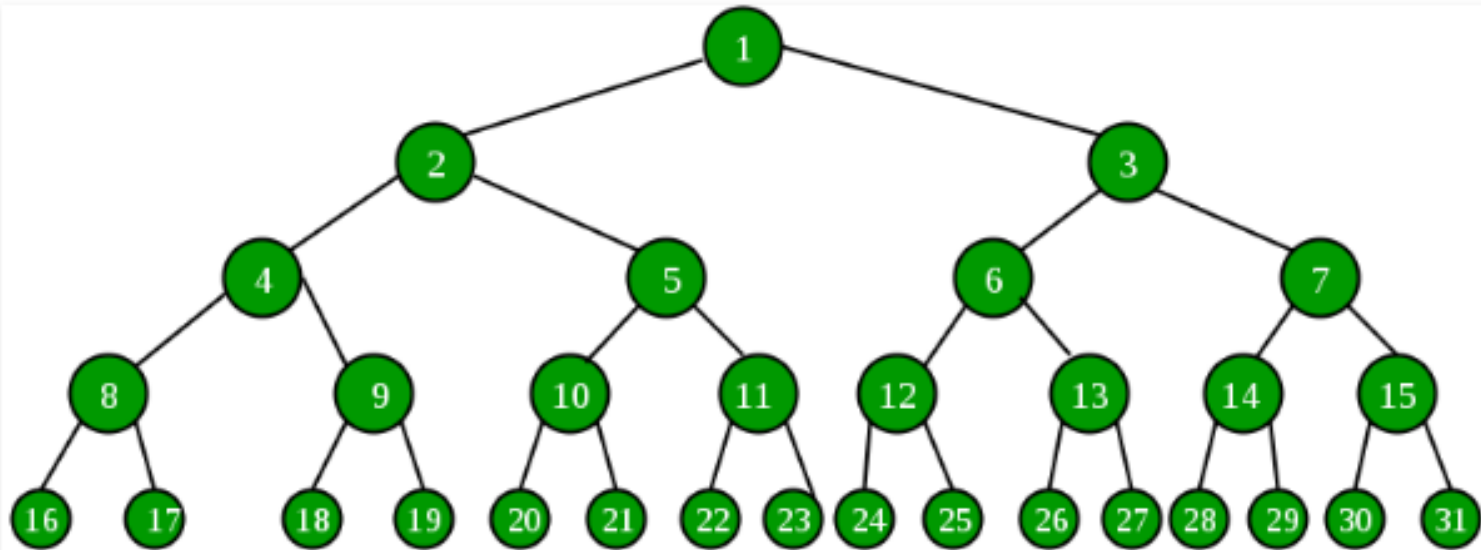
```
if partiele_score + mogelijke_score > laagste_score:  
    verwijderen
```

Tree

Elke node: links, rechts, rechtdoor

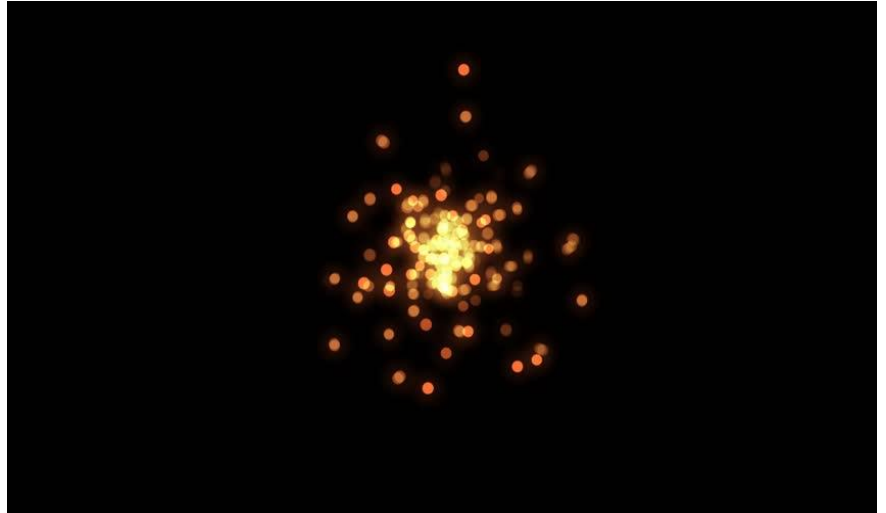
3^{n-2} mogelijke conformaties

Vind de laagste score



Firefly

```
maak_zwerm()  
for i in range(N):  
    bepaal_score()  
    aanpassen()
```



Genetic

Populatie van willekeurige conformaties.

De vijftig beste conformaties maken kinderen.

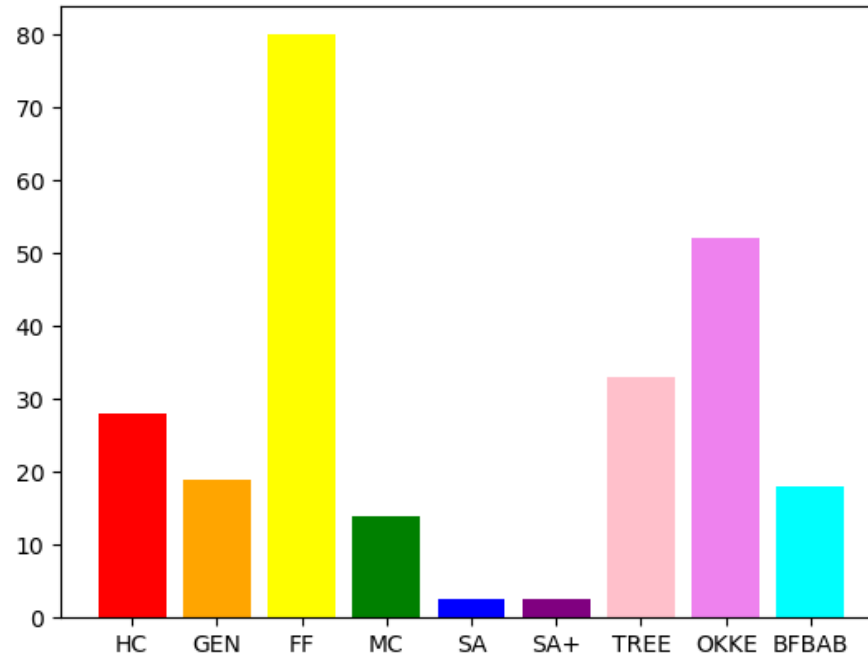
Kinderen vervangen de zwakke conformaties

```
populatie = genereer_conformaties
for i in range(N):
    ouder1, ouder2 = kies_ouders(populatie)
    kind1, kind2 = crossover(ouder1, ouder2)
    mutatie1, mutatie2 = mutatie(kind1, kind2)
    swap(mutatie1, mutatie2, populatie)
```

Resultaten: scores

Length	Type	Optimal	HC (28s)	Gen (19s)	FF (80s)	MC (14s)	SA (2.5s)	SA+ (2.5s)	Tree (33s)	Okke (52s)	BFBaB (18s)
8	HP	-3	-3	-3	-3	-3	-3	-3	-3	-3	-3
14	HP	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6
20	HP	-9	-6	-9	-7	-8	-7	-7	NR	-9	-9
36	HP	-14	-10	-13	-8	-12	-8	-10	NR	NR	-12
50	HP	-21	-16	-18	-14	-17	-11	-14	NR	NR	-19
36	HPC	?	-17	-24	-15	-24	-15	-17	NR	NR	-21
36	HPC	?	-25	-35	-24	-33	-22	-23	NR	NR	-35
50	HPC	?	-26	-29	-21	-26	-20	-22	NR	NR	-28
50	HPC	?	-22	-32	-17	-29	-17	-22	NR	NR	-33

Resultaten: runtime



Conclusie

- Op dit moment laten de resultaten zien dat algoritmes met de minste tijd en het minste geheugen tot een optimum kunnen komen. Deze algoritmes kunnen we in de toekomst in de richting van native states uitbreiden.

Toekomst plannen

- Kritiek op het simuleren van eiwitstructuren is dat er wat stabiliteit betreft een beste vouwing gevonden kan worden maar dat dit niet noodzakelijkerwijs de vouwing is die daadwerkelijk in de natuur voorkomt. Een aanpassing van het huidige programma kan de gevonden oplossingen rangschikken naar mate van eenvoud.
- Bij strings met lengte 48 zijn er 1000 tot 1.5 miljoen optimale vouwingen (Yue et al., 1995). Een relevante richting voor de simulatie van eiwitstructuren is het benaderen van 'native states', ofwel: hoe de structuren in de natuur voorkomen.
- In ons model van het vouwen van de eiwitstructuur is met name bij de iteratieve algoritmes een uitgelezen kans om de gevonden optima te rangschikken. Hierbij is een optimum dat met minder vouwingen tot stand komt dichterbij een 'native state' en zal deze hoger in de ranglijst komen te staan.