

# Experimentatie

## Statespace

Berekening:

Keten	Lower-bound	Upper-bound
HH <b>PH</b> HH <b>PH</b> HH <b>PH</b>		
H <b>PH</b> PPH <b>PH</b> PP <b>PH</b> HHPP <b>PH</b>		
PP <b>PH</b> PPH <b>PH</b> PPPPHHHHHH <b>PH</b> HHPPPPH <b>PH</b> PP		
HH <b>PH</b> PP <b>PH</b> PHHHHH <b>PH</b> PP <b>PH</b> PPPPHPP <b>PH</b> PPPPHPPH <b>PH</b> HHHH <b>PH</b> PH <b>PH</b> PHH		
CPP <b>CH</b> PP <b>CH</b> PPCP <b>PH</b> HHHHHHCC <b>CH</b> PPCP <b>CH</b> PPH <b>CH</b>		
H <b>CH</b> PP <b>CH</b> PP <b>CH</b> CH <b>PH</b> PP <b>PH</b> PPPPHPP <b>PH</b> PPPPHPP <b>PH</b> HHHCC <b>CH</b> CH <b>CH</b> CHH		
H <b>CH</b> PP <b>PH</b> PH <b>CH</b> HHHH <b>CH</b> PPHPP <b>PH</b> PPPPCP <b>PH</b> PP <b>PH</b> PPH <b>PH</b> HHHH <b>CH</b> PH <b>PH</b> PHH		

## Score

Berekening Upper-bound:

- $(\text{aantal H} - 1) * 1 + (\text{aantal C} - 1) * 5$

Keten	Lower-bound	Upper-bound
HHPHHHPHPPHHHPH	0	-9
HPHPPHHHPHPPHPPHPPH	0	-9
PPPHHPPHHPPPPPHHHHHHHPPHHPPPPHHPPHPP	0	-15
HHPHPPHPPHHHHHPHPPHPPHPPPPHPPHPPPHHHHPHPPHPPH	0	-23
CPPCHPPCHPPCPPHHHHHHCCPCHPPCPCHPPHPC	0	-55
HCPHPCPHPCCHCHPPPHPPHPPPPHPCPHPPHPPHHCCCHCHCHHH	0	-63
HCPHPPHCHHHHPCPPHPPHPPPPCPPHPPHPPHHHCHPHPHPPH	0	-45

## Experiment

Met dit experiment willen we kijken hoe de beginscore van een streng invloed heeft op de score die uiteindelijk kan worden behaald met het 'hillclimber' algoritme. Om deze vraag te beantwoorden, gebruiken we de drie verschillende startpunten waarmee we onze hillclimber kunnen runnen, namelijk:

- Straight (recht gevouwen)
- Random (random gevouwen)
- Depth (eerst gerund met depth-first)

Welke runs maken wij:

- Elk algoritme runnen wij met de volgende strengs:
  - Streng 1 (alleen P & H, laagste score upper-bound)
  - Streng 4 (alleen P & H, hoogste score upper-bound)
  - Streng 8 (P, H & C, laagste score upper-bound)
  - Streng 7 (P, H & C, hoogste score upper-bound)
- Er is voor deze vier strengs gekozen omdat we graag voor beide soorten strengs (zonder en met cysteïne) het verschil wilden laten zien tussen de streng met de hoogste en de streng met de laagste upper-bound.
- Voor elke combinatie berekenen we de gemiddelde score per iteratie over **5** runs.
- Andere instellingen (voor iedere run gelijk):
  - Dimension: 3D
  - Iteraties : 1000
  - Maximum runtime depth: 15 seconden
- Verwachtingen beginscore:
  - Een recht gevouwen streng heeft altijd een score van 0 omdat de verschillende aminozuren per definitie niet naast elkaar liggen.
  - Een random gevouwen streng heeft altijd een andere score omdat het random is.
  - Een streng vanuit depth-first algoritme heeft altijd dezelfde beginscore per streng. Dit komt omdat we de depth-first eerst 15 seconden laten runnen en dan met de beste streng vanuit daar verder gaan. Omdat hij steeds even lang de tijd krijgt, komt de beste streng altijd neer op dezelfde score.
- Verwachtingen eindscore:
  - We verwachten dat een streng met een lagere beginscore meer en sneller vooruit zal gaan tijdens de hillclimber, maar dat de uiteindelijke score voor alle drie de startpunten ongeveer gelijk zal zijn.
- Resultaten:
  - De plots van de resultaten staan in de map 'resultaten', met op de x-as het aantal iteraties en op de y-as de score.
  - Zie hieronder een tabel voor uitleg welke plotnamen bij welke soort run horen.

Naam	Instellingen
plot1.1	Random, keten 1
plot1.2	Random, keten 4
plot1.3	Random, keten 8
plot1.4	Random, keten 7
plot2.1	Straight, keten 1
plot2.2	Straight, keten 4
plot2.3	Straight, keten 8
plot2.4	Straight, keten 7
plot3.1	Depth, keten 1
plot3.2	Depth, keten 4
plot3.3	Depth, keten 8
plot3.4	Depth, keten 7