Genetic algorithms as a search tool in the string landscape

Steven Abel
IPPP Durham

SAA, J.Rizos, JHEP 1408 (2014) 010,1404.7359 hep-th
SAA, Constantin, Harvey, Lukas, *Fortsch.Phys.* 70 (2022) 5, 2200034 • e-Print: 2110.14029

***Background stolen from Gary Shiu's "String Genome Project"**

Motivation …

Motivation …

- String theories typically produce vast theory spaces. e.g 10^500 or even 10^(272000)

  Denef, Douglas; Ashok, Douglas; Denef, Douglas, Greene, Zokowski; Taylor, Wang

Motivation …

- String theories typically produce vast theory spaces. e.g 10^500 or even 10^(272000)
  Denef, Douglas; Ashok, Douglas; Denef, Douglas, Greene, Zokowski; Taylor, Wang

- Finding the “Standard Model” typically requires solving a set of Diophantine conditions (e.g. setting some index to get 3 generations, anomaly cancellation, GSO conditions etc)

Motivation …

- String theories typically produce vast theory spaces. e.g 10^500 or even 10^(272000)
  Denef, Douglas; Ashok, Douglas; Denef, Douglas, Greene, Zokowski; Taylor, Wang
- Finding the "Standard Model" typically requires solving a set of Diophantine conditions (e.g. setting some index to get 3 generations, anomaly cancellation, GSO conditions etc)
- Such tasks are typically computationally hard. Difficulty (probably) increases exponentially with the size of the search space (but the solution can be verified in polynomial time - if in NP). Halverson, Ruehle; Halverson, Plesser, Ruehle, Tian

Motivation …

- String theories typically produce vast theory spaces. e.g 10^500 or even 10^(272000)
  Denef, Douglas; Ashok, Douglas; Denef, Douglas, Greene, Zokowski; Taylor, Wang

- Finding the "Standard Model" typically requires solving a set of Diophantine conditions (e.g. setting some index to get 3 generations, anomaly cancellation, GSO conditions etc)

- Such tasks are typically computationally hard. Difficulty (probably) increases exponentially with the size of the search space (but the solution can be verified in polynomial time - if in NP). Halverson, Ruehle; Halverson, Plesser, Ruehle, Tian

- In recent years machine learning has been intensely studied as a means of learning about the "local string landscape".

Motivation …

- String theories typically produce vast theory spaces. e.g 10^500 or even 10^(272000)
  Denef, Douglas; Ashok, Douglas; Denef, Douglas, Greene, Zokowski; Taylor, Wang

- Finding the "Standard Model" typically requires solving a set of Diophantine conditions (e.g. setting some index to get 3 generations, anomaly cancellation, GSO conditions etc)

- Such tasks are typically computationally hard. Difficulty (probably) increases exponentially with the size of the search space (but the solution can be verified in polynomial time - if in NP). Halverson, Ruehle; Halverson, Plesser, Ruehle, Tian

- In recent years machine learning has been intensely studied as a means of learning about the "local string landscape".

- However there is a venerable technique that doesn't seem to be going away: **Genetic Algorithms** Turing; Barricelli; Fraser, Burnell; Crosby; Bremermann; ***Holland***; Goldberg; Jones

GAs in particle physics …

- Yamaguchi and H. Nakajima (2000)
- Allanach, Grellscheid, Quevedo (2004)
- Akrami, Scott, Edsjo, Conrad and Bergstrom (2009)
- Bl˚aba¨ck, Danielsson and Dibitetto, (2013)
- SAA, Rizos (2014)
- Ruehle (2017)
- SAA, Cerdeno, Robles (2018) ***PMSSM20**
- Cole, Schachner, Shiu (2019)
- AbdusSalam, SAA, Cicoli, Quevedo, Shukla (2020)
- Bena, Bl˚aba¨ck, Grana, Luest (2021)
- SAA, Constantin, Lukas, Harvey (2021)
- Loges, Shiu (2021)
- Cole, Krippendorf, Schachner, Shiu (2021)

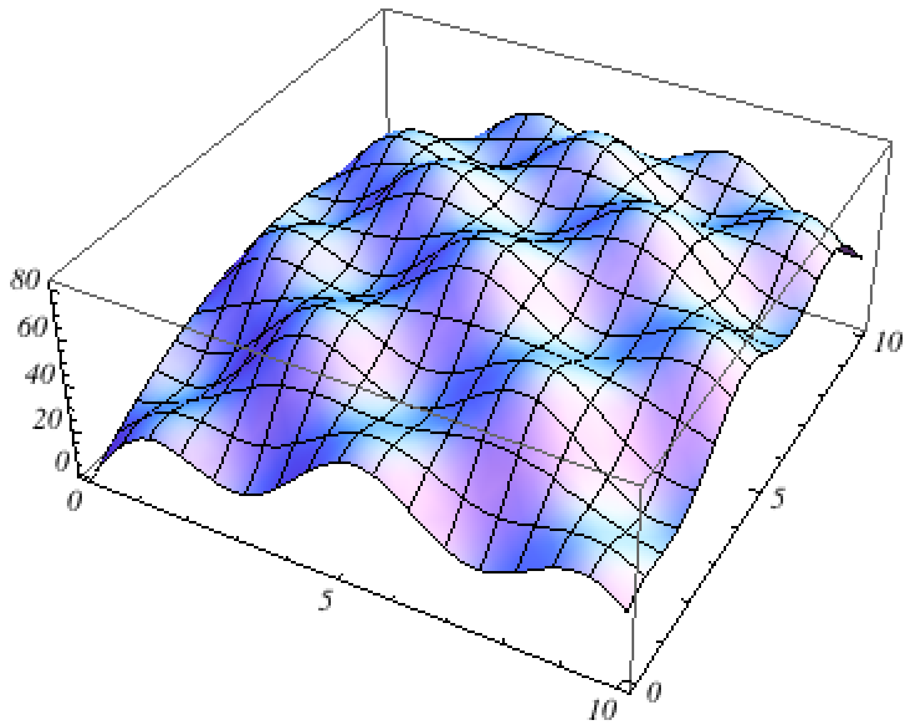
Overview …

- Some GA introduction and background
  - How do they work?
  - Why do they work?
- GA’s on a simple string construction
- GA’s versus Reinforcement Learning

GA's for searching a string sized landscape

GA's for searching a string sized landscape

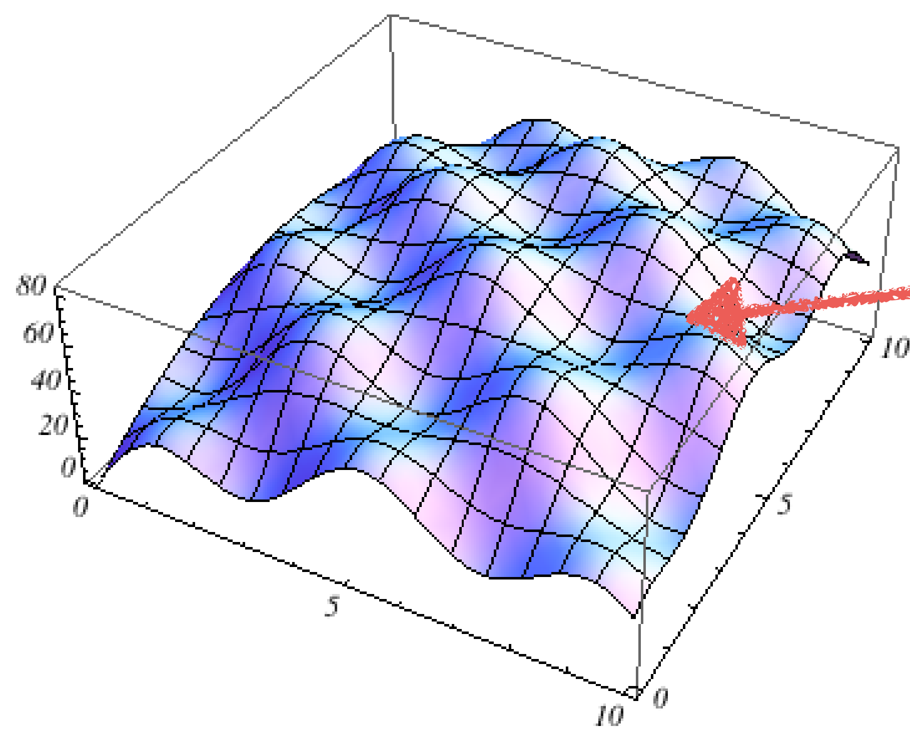
Example landscape task: find global maximum to 250 decimal places without using calculus …



$$f(x, y) = 12\left(\cos\frac{3y}{2}\,\sin\frac{3x}{2} + x + y\right) - x^2 - y^2.$$

GA's for searching a string sized landscape

Example landscape task: find global maximum to 250 decimal places without using calculus …



$$x = a.bcdef\ldots$$
$$y = g.hijkl\ldots$$

$$\Longrightarrow \quad 10^{500}$$

$$f(x,y) = 12\left(\cos\frac{3y}{2}\,\sin\frac{3x}{2} + x + y\right) - x^2 - y^2.$$

Define a "creature" and write out its coordinates => genotype

Terminology: *Genotype* = data. *Phenotype* = *f(x,y)*.

$$x = a.bcdef...$$

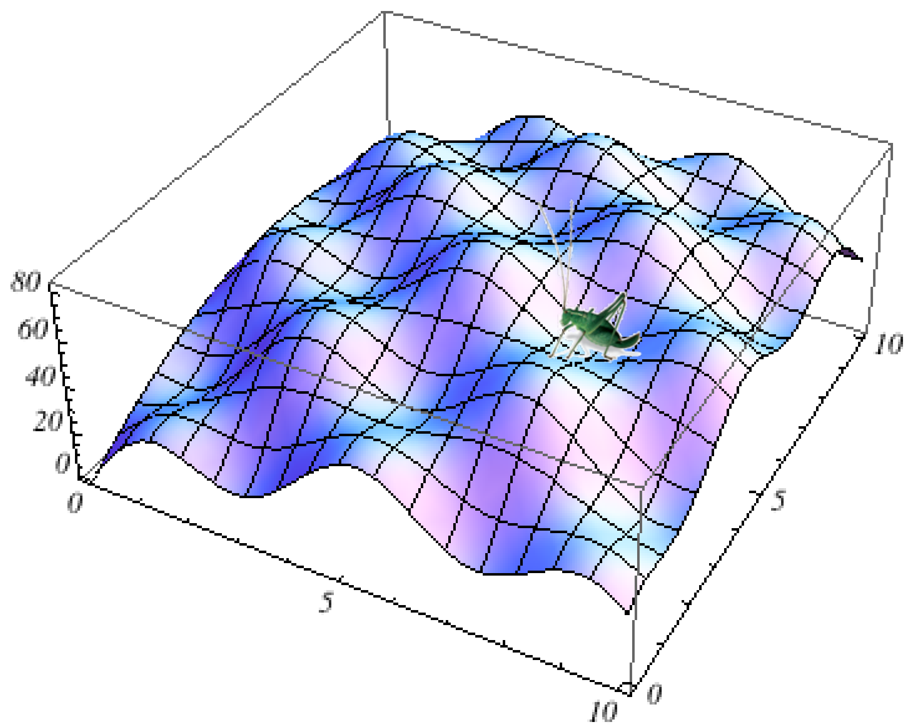
$$y = g.hijkl...$$

Define a “creature” and write out its coordinates => genotype

Terminology: *Genotype* = data. *Phenotype* = $f(x,y)$.

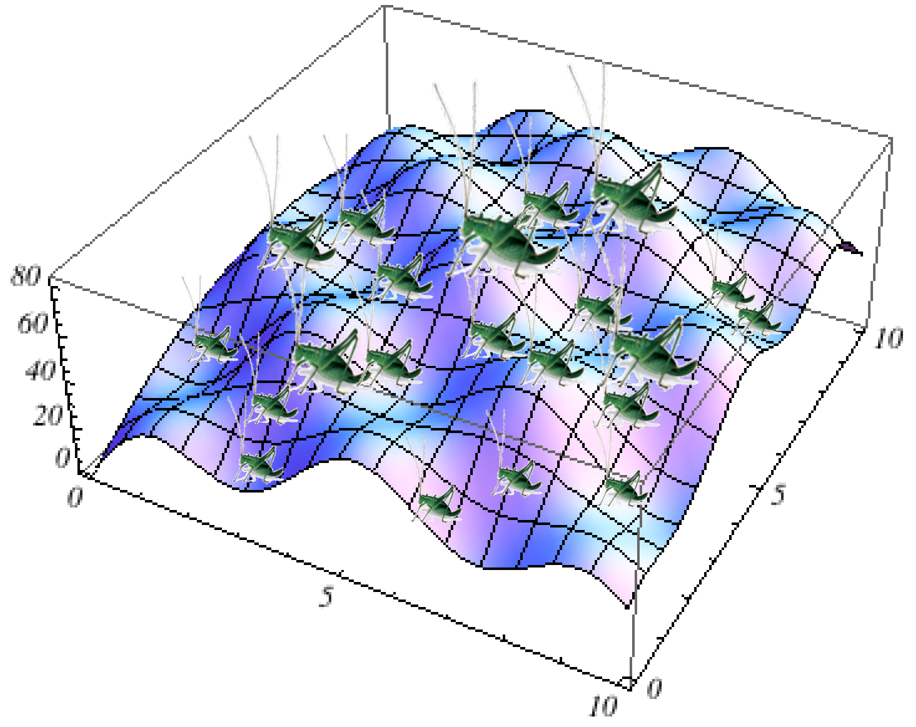
$$x = a.bcdef...$$

$$y = g.hijkl...$$



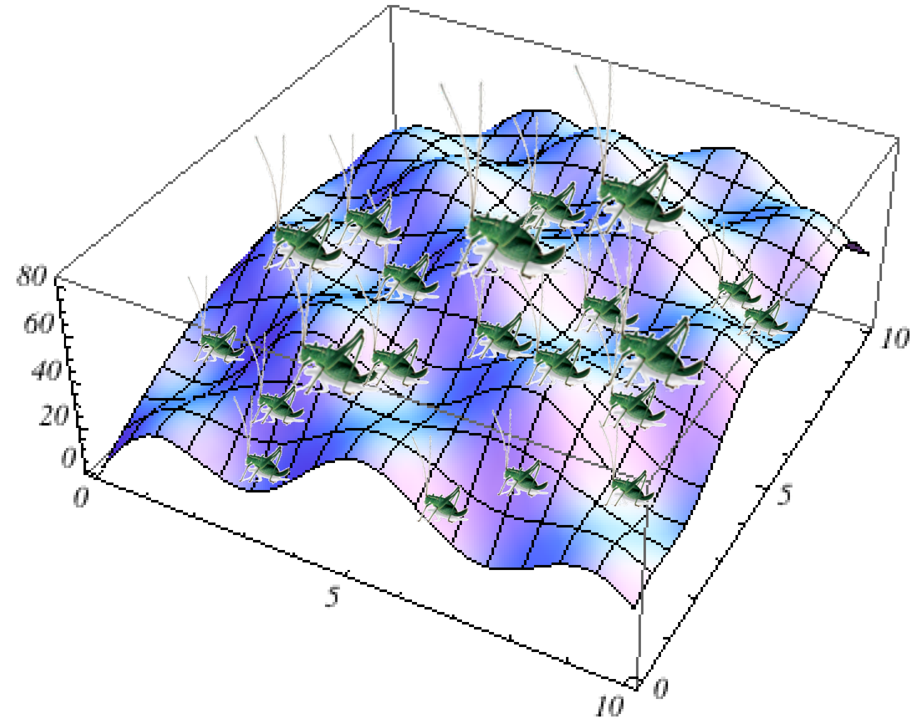
Work with a population of typically ~100 individuals initially sprinkled at random

Step 0: Define fitness function, and work out the fitness *F* of each individual (e.g. $F = f(x,y)$ in this case).



Step 1: Selection: Select pairs for breeding such that the most fit individuals can breed several times, while unfit ones might not breed at all: e.g. "roulette wheel" based on *ranking k,* with $P_1 = \alpha P_{N_{\rm pop}}$:

$$P_k = \frac{2}{(1+\alpha)N_{\rm pop}}\left(1 + \frac{N_{\rm pop}-k}{N_{\rm pop}-1}(\alpha - 1)\right)$$

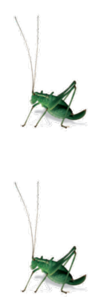


Step 2: breeding: cut and splice genotypes of breeding pairs somehow (not really crucial how) to make an entirely new population of the same size.

$$g.hij\,|\,kl$$

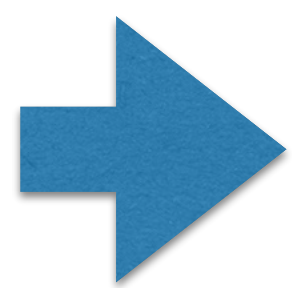
$$a.bcd\,|\,ef$$

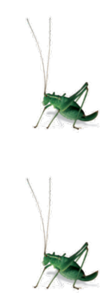
Step 2: breeding: cut and splice genotypes of breeding pairs somehow (not really crucial how) to make an entirely new population of the same size.



$$g.hij \,|\, ef$$

$$a.bcd \,|\, kl$$





Step 3: Mutation of a randomly chosen small percentage of digits (alleles).

$$a.bcdefghij...$$

Step 3: Mutation of a randomly chosen small percentage of digits (alleles).

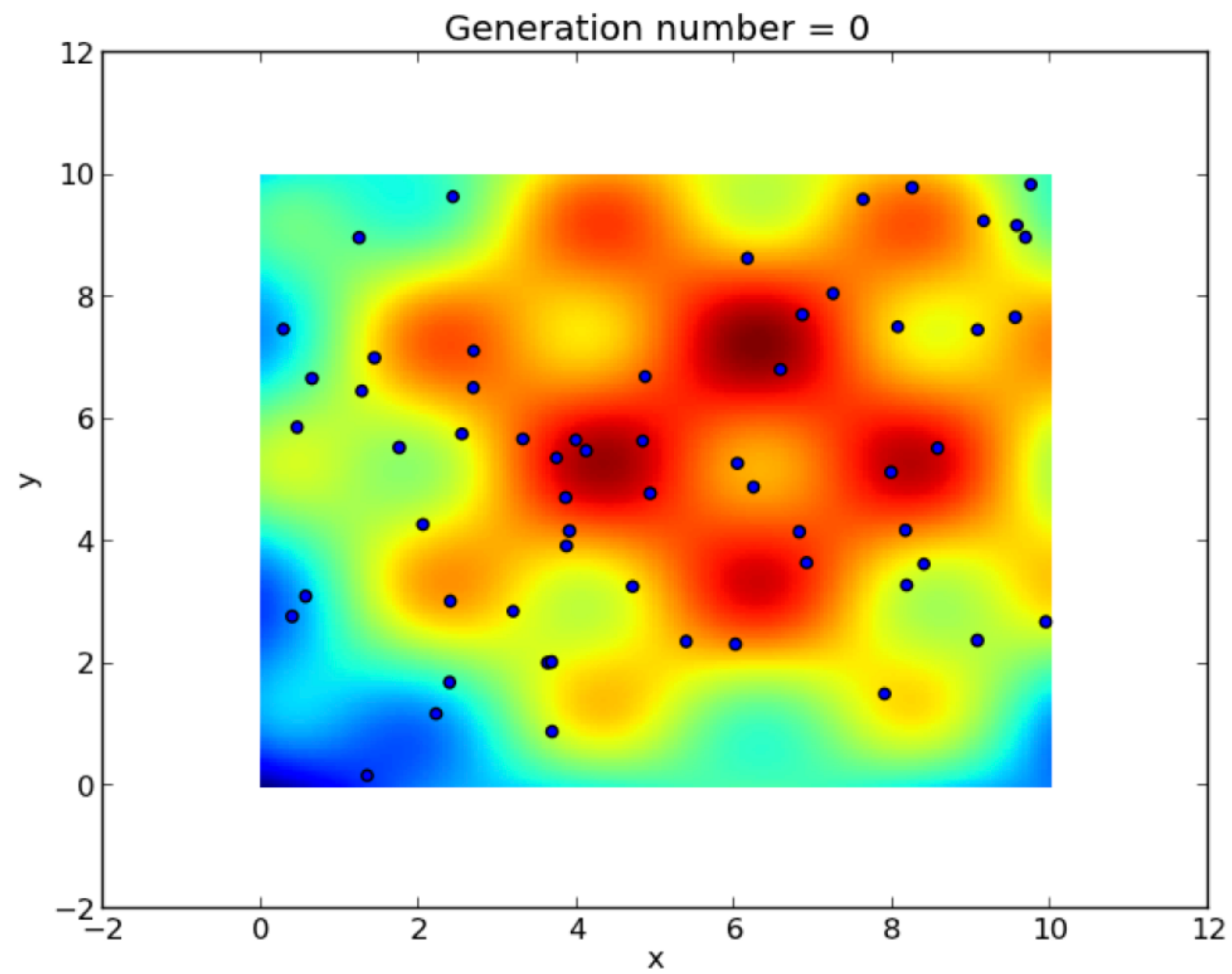
$$a.bcdef'gh'ij...$$

Step 3: Mutation of a randomly chosen small percentage of digits (alleles).

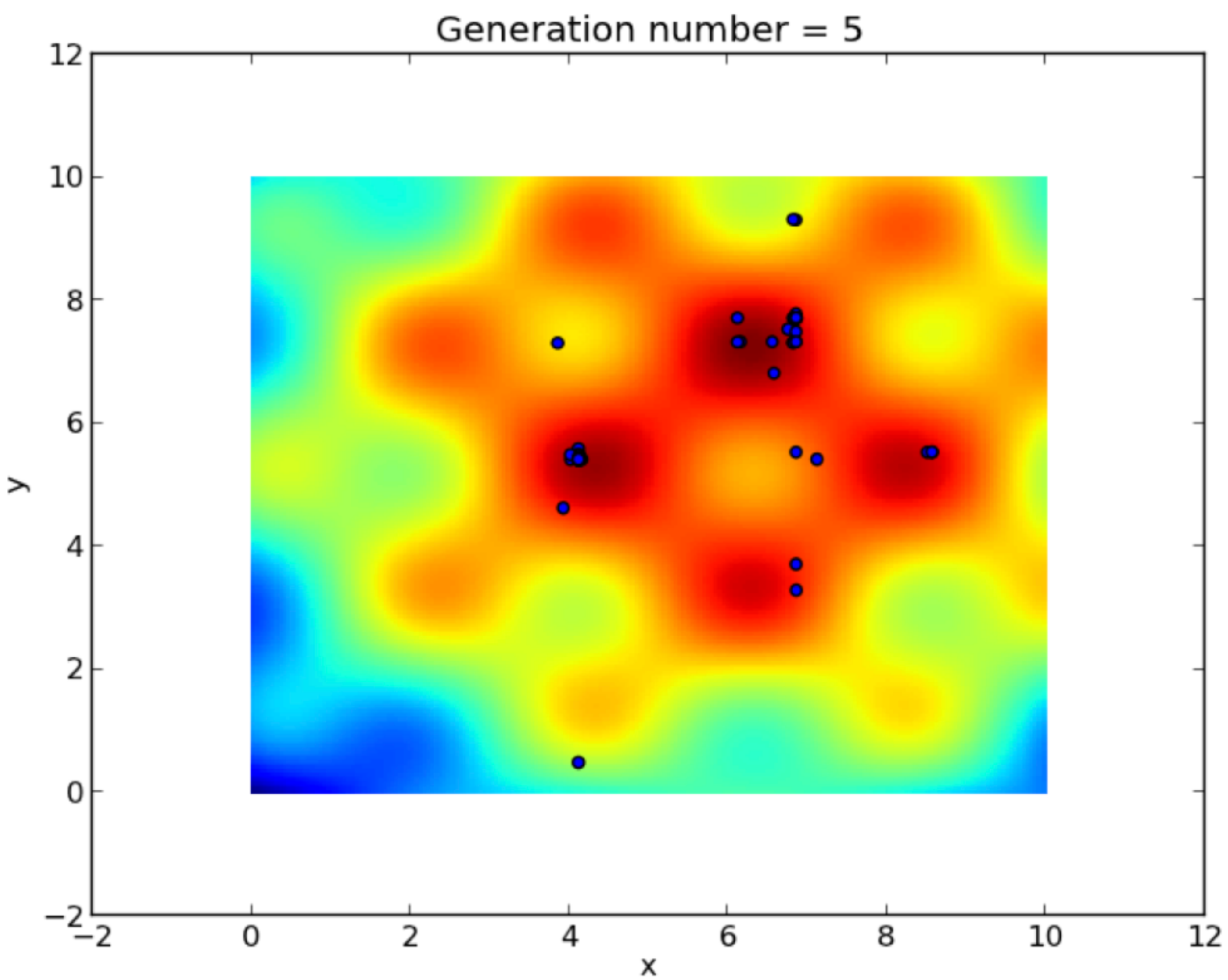
$$a.bcdef'gh'ij...$$

Steps 4 … infinity: rinse and repeat. The population should converge round solutions.

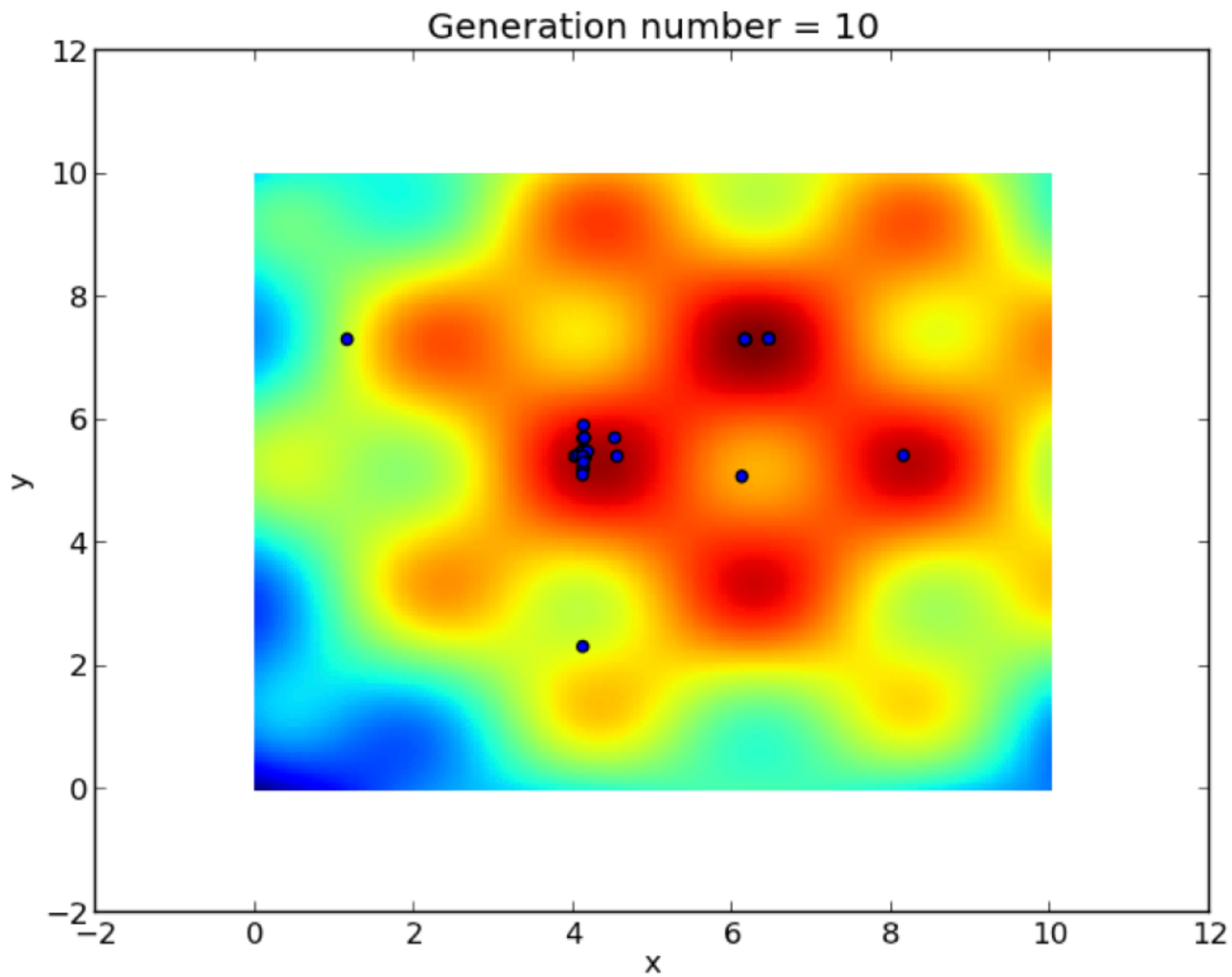
***Summary* — three crucial ingredients: *Selection* (favours the optimisation); *Breeding/crossover* (propagates favourable properties); *Mutation* (prevents stagnation: evolution proceeds by punctuated equilibria)**



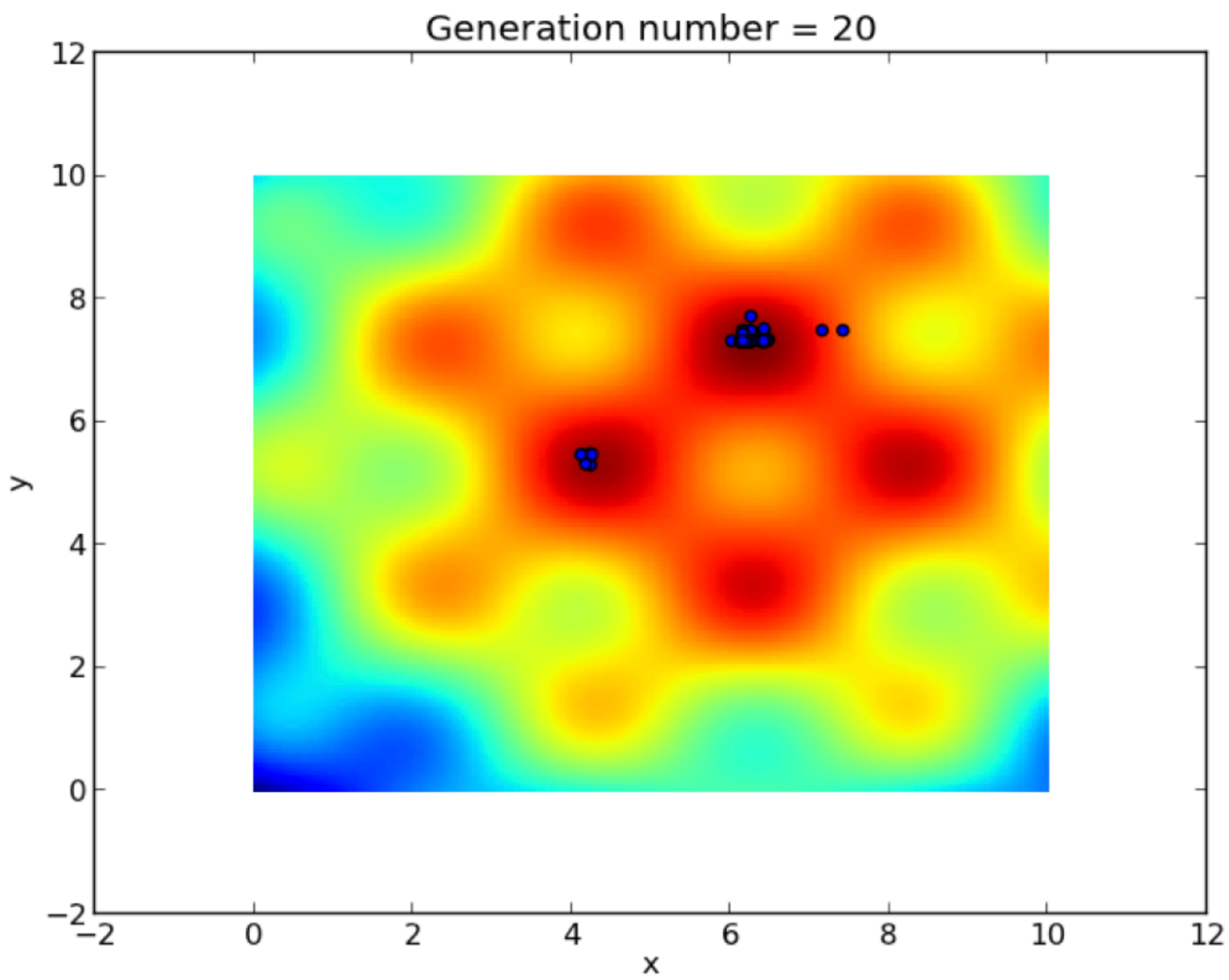
Summary* —** three crucial ingredients: ***Selection (favours the optimisation); ***Breeding/crossover*** (propagates favourable properties); ***Mutation*** (prevents stagnation: evolution proceeds by punctuated equilibria)



Summary* —** three crucial ingredients: ***Selection (favours the optimisation); ***Breeding/crossover*** (propagates favourable properties); ***Mutation*** (prevents stagnation: evolution proceeds by punctuated equilibria)

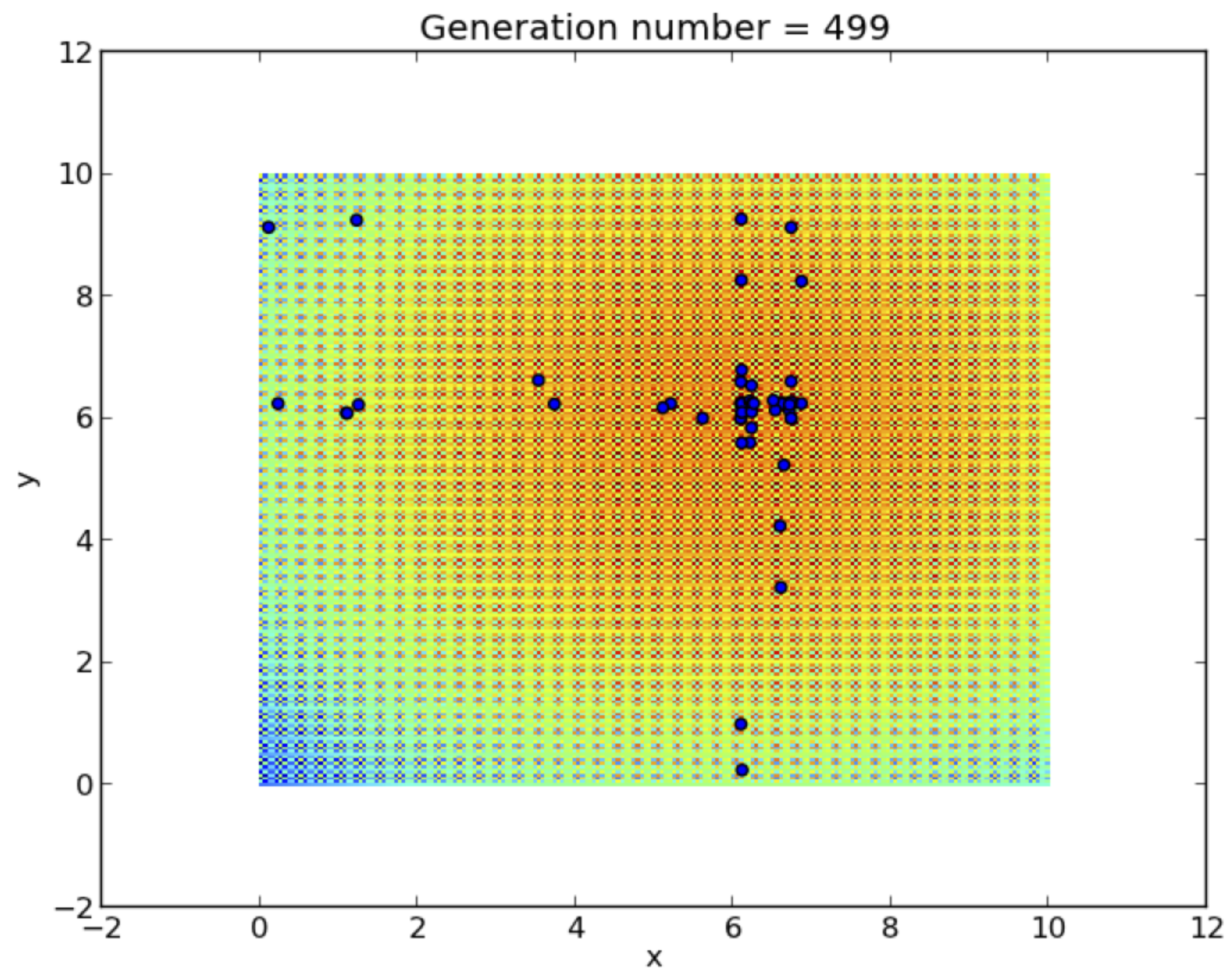


Summary* —** three crucial ingredients: ***Selection (favours the optimisation); ***Breeding/crossover*** (propagates favourable properties); ***Mutation*** (prevents stagnation: evolution proceeds by punctuated equilibria)



Summary* —** three crucial ingredients: ***Selection (favours the optimisation); ***Breeding/crossover*** (propagates favourable properties); ***Mutation*** (prevents stagnation: evolution proceeds by punctuated equilibria)

***Summary* — three crucial ingredients: *Selection* (favours the optimisation); *Breeding/crossover* (propagates favourable properties); *Mutation* (prevents stagnation: evolution proceeds by punctuated equilibria)



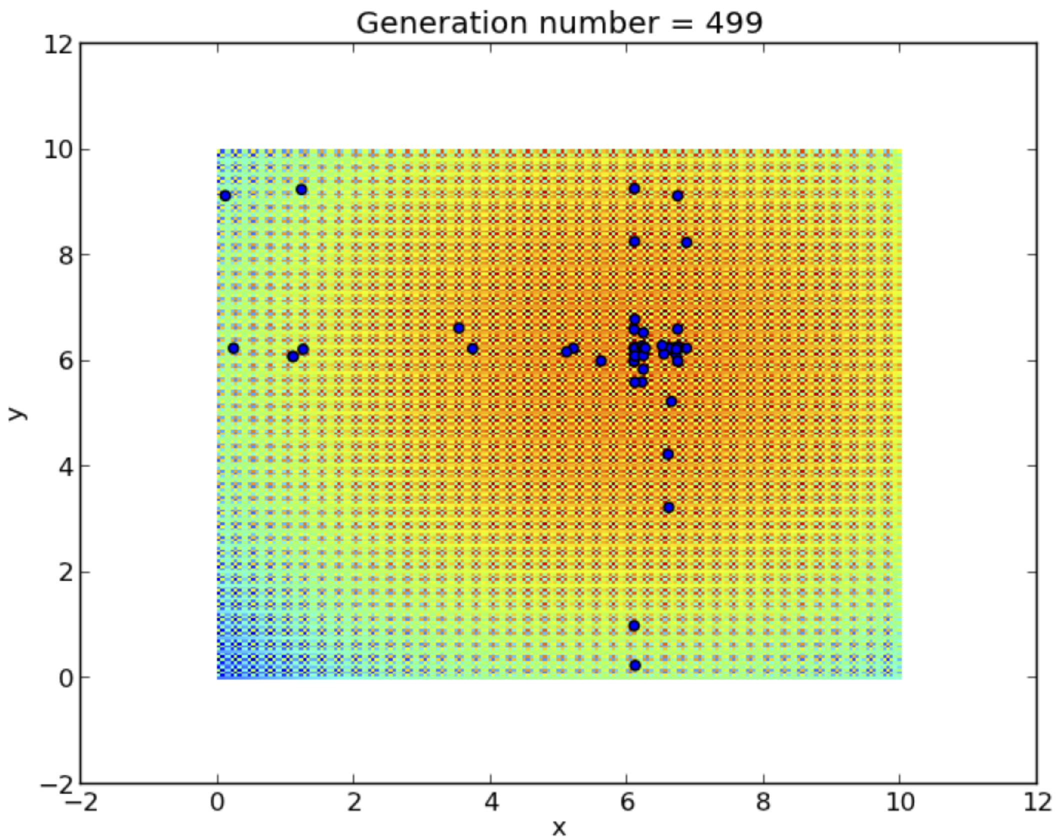
Why do they work?

- Holland proposed a probabilistic explanation for the efficiency of genetic algorithms: based on growth rate of "good" schema *S* , e.g. here $S = 61 * * * 62 * **$
- Holland argues that initial growth of a good schema in the population is exponential
- Selection pushes towards convergence
- Mutation pushes system away from convergence
- Some controversy in 1990s, rehabilitated somewhat by Poli. (Not many good general competing theories)
- Fitness/distance correlation seems to be important
  Holland; David; Jones+Forrest; Collard, Gaspar, Clergue, Escazu

Why do they work?

- Holland proposed a probabilistic explanation for the efficiency of genetic algorithms: based on growth rate of "good" schema *S* , e.g. here $S = 61 * * * 62 * **$
- Holland argues that initial growth of a good schema in the population is exponential
- Selection pushes towards convergence
- Mutation pushes system away from convergence
- Some controversy in 1990s, rehabilitated somewhat by Poli. (Not many good general competing theories)
- Fitness/distance correlation seems to be important
  Holland; David; Jones+Forrest; Collard, Gaspar, Clergue, Escazu

In this example the leading digits of x and y are schemata and get propagated throughout the population



First string example

SAA+Rizos, 2014

- Find a phenomenologically attractive Pati-Salam model.
- We will consider the "fermionic string construction". These are general 4D models in which the world sheet degrees of freedom are fermions. Kawai, Lewellyn, Tye; Antoniadis, Bachas, Kounnas
- PS Models are defined in terms of a set of basis vectors (for the experts) Faraggi, Kounnas, Nooij, Rizos

$$
\begin{aligned}
v_1 = \mathbb{1} &= \{\psi^\mu,\ \chi^{1,\dots,6}, y^{1,\dots,6}, \omega^{1,\dots,6} | \bar{y}^{1,\dots,6}, \bar{\omega}^{1,\dots,6}, \bar{\eta}^{1,2,3}, \bar{\psi}^{1,\dots,5}, \bar{\phi}^{1,\dots,8}\} \\
v_2 = S &= \{\psi^\mu, \chi^{1,\dots,6}\} \\
v_{2+i} = e_i &= \{y^i, \omega^i | \bar{y}^i, \bar{\omega}^i\},\ i = 1, \dots, 6 \\
v_9 = b_1 &= \{\chi^{34}, \chi^{56}, y^{34}, y^{56} | \bar{y}^{34}, \bar{y}^{56}, \bar{\eta}^1, \bar{\psi}^{1,\dots,5}\} \\
v_{10} = b_2 &= \{\chi^{12}, \chi^{56}, y^{12}, y^{56} | \bar{y}^{12}, \bar{y}^{56}, \bar{\eta}^2, \bar{\psi}^{1,\dots,5}\} \\
v_{11} = z_1 &= \{\bar{\phi}^{1,\dots,4}\} \\
v_{12} = z_2 &= \{\bar{\phi}^{5,\dots,8}\} \\
v_{13} = \alpha &= \{\bar{\psi}^{45}, \bar{y}^{1,2}\}\ .
\end{aligned}
$$

- in addition to a set of GSO projection phases $c\begin{bmatrix} v_i \\ v_j \end{bmatrix}, i, j = 1, \dots, n$

Our genotype will be these phases (think of the string construction as a black box that turns these numbers into phenomenological model)

$$c_{ij} = \begin{array}{cc} & \begin{array}{ccccccccccccc} \mathbb{1} & S & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & b_1 & b_2 & z_1 & z_2 & \alpha \end{array} \\ \begin{array}{c} \mathbb{1} \\ S \\ e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ b_1 \\ b_2 \\ z_1 \\ z_2 \\ \alpha \end{array} & \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & \ell_{26} & \ell_{27} & \ell_{28} & \ell_{29} & \ell_{30} & \ell_6 & 0 & \ell_{14} & \ell_{20} & \ell_{41} \\ 1 & 1 & \ell_{26} & 0 & \ell_{31} & \ell_{32} & \ell_{33} & \ell_{34} & \ell_7 & 0 & \ell_{15} & \ell_{21} & \ell_{42} \\ 1 & 1 & \ell_{27} & \ell_{31} & 0 & \ell_{35} & \ell_{36} & \ell_{37} & 0 & \ell_{10} & \ell_{16} & \ell_{22} & \ell_{43} \\ 1 & 1 & \ell_{28} & \ell_{32} & \ell_{35} & 0 & \ell_{38} & \ell_{39} & 0 & \ell_{11} & \ell_{17} & \ell_{23} & \ell_{44} \\ 1 & 1 & \ell_{29} & \ell_{33} & \ell_{36} & \ell_{38} & 0 & \ell_{40} & \ell_8 & \ell_{12} & \ell_{18} & \ell_{24} & \ell_{45} \\ 1 & 1 & \ell_{30} & \ell_{34} & \ell_{37} & \ell_{39} & \ell_{40} & 0 & \ell_9 & \ell_{13} & \ell_{19} & \ell_{25} & \ell_{46} \\ 0 & 0 & \ell_6 & \ell_7 & 0 & 0 & \ell_8 & \ell_9 & 1 & 0 & \ell_2 & \ell_4 & \ell_{47} \\ 0 & 0 & 0 & 0 & \ell_{10} & \ell_{11} & \ell_{12} & \ell_{13} & 0 & 1 & \ell_3 & \ell_5 & \ell_{48} \\ 1 & 1 & \ell_{14} & \ell_{15} & \ell_{16} & \ell_{17} & \ell_{18} & \ell_{19} & \ell_2 & \ell_3 & 1 & \ell_1 & \ell_{49} \\ 1 & 1 & \ell_{20} & \ell_{21} & \ell_{22} & \ell_{23} & \ell_{24} & \ell_{25} & \ell_4 & \ell_5 & \ell_1 & 1 & \ell_{50} \\ 1 & 1 & \ell_{41} & \ell_{42} & \ell_{43} & \ell_{44} & \ell_{45} & \ell_{46} & \ell_{47}+1 & \ell_{48}+1 & \ell_{49}+1 & \ell_{50} & \ell_{51} \end{pmatrix} \end{array} \mod 2$$

$$c\begin{bmatrix} v_i \\ v_j \end{bmatrix}, i, j = 1, \ldots, n$$

Our genotype will be these phases (think of the string construction as a black box that turns these numbers into phenomenological model)

$$c\begin{bmatrix} v_i \\ v_j \end{bmatrix}, i, j = 1, \ldots, n$$

$$c_{ij} = \begin{array}{cc} & \begin{array}{ccccccccccccc} \mathbb{1} & S & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & b_1 & b_2 & z_1 & z_2 & \alpha \end{array} \\ \begin{array}{c} \mathbb{1} \\ S \\ e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ b_1 \\ b_2 \\ z_1 \\ z_2 \\ \alpha \end{array} & \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & \ell_{26} & \ell_{27} & \ell_{28} & \ell_{29} & \ell_{30} & \ell_6 & 0 & \ell_{14} & \ell_{20} & \ell_{41} \\ 1 & 1 & \ell_{26} & 0 & \ell_{31} & \ell_{32} & \ell_{33} & \ell_{34} & \ell_7 & 0 & \ell_{15} & \ell_{21} & \ell_{42} \\ 1 & 1 & \ell_{27} & \ell_{31} & 0 & \ell_{35} & \ell_{36} & \ell_{37} & 0 & \ell_{10} & \ell_{16} & \ell_{22} & \ell_{43} \\ 1 & 1 & \ell_{28} & \ell_{32} & \ell_{35} & 0 & \ell_{38} & \ell_{39} & 0 & \ell_{11} & \ell_{17} & \ell_{23} & \ell_{44} \\ 1 & 1 & \ell_{29} & \ell_{33} & \ell_{36} & \ell_{38} & 0 & \ell_{40} & \ell_8 & \ell_{12} & \ell_{18} & \ell_{24} & \ell_{45} \\ 1 & 1 & \ell_{30} & \ell_{34} & \ell_{37} & \ell_{39} & \ell_{40} & 0 & \ell_9 & \ell_{13} & \ell_{19} & \ell_{25} & \ell_{46} \\ 0 & 0 & \ell_6 & \ell_7 & 0 & 0 & \ell_8 & \ell_9 & 1 & 0 & \ell_2 & \ell_4 & \ell_{47} \\ 0 & 0 & 0 & 0 & \ell_{10} & \ell_{11} & \ell_{12} & \ell_{13} & 0 & 1 & \ell_3 & \ell_5 & \ell_{48} \\ 1 & 1 & \ell_{14} & \ell_{15} & \ell_{16} & \ell_{17} & \ell_{18} & \ell_{19} & \ell_2 & \ell_3 & 1 & \ell_1 & \ell_{49} \\ 1 & 1 & \ell_{20} & \ell_{21} & \ell_{22} & \ell_{23} & \ell_{24} & \ell_{25} & \ell_4 & \ell_5 & \ell_1 & 1 & \ell_{50} \\ 1 & 1 & \ell_{41} & \ell_{42} & \ell_{43} & \ell_{44} & \ell_{45} & \ell_{46} & \ell_{47}+1 & \ell_{48}+1 & \ell_{49}+1 & \ell_{50} & \ell_{51} \end{pmatrix} \end{array} \mod 2$$

51 independent phases in these models: hence search space is $2^{51} = 2 \times 10^{15}$

This search space is (just about) searchable deterministically so we can compare the two methods. Assel, Christodoulides, Faraggi, Kounnas, Rizos

The phases determine the characteristics of the models

(a) 3 complete family generations, $n_g = 3$

(b) Existence of PS breaking Higgs, $k_R \geq 1$

(c) Existence of SM Higgs doublets, $n_h \geq 1$

(d) Absence of exotic fractional charge states, $n_e = 0$

(e) Existence of top Yukawa coupling

This search space is (just about) searchable deterministically so we can compare the two methods. Assel, Christodoulides, Faraggi, Kounnas, Rizos

The phases determine the characteristics of the models

(a) 3 complete family generations, $n_g = 3$

(b) Existence of PS breaking Higgs, $k_R \geq 1$

(c) Existence of SM Higgs doublets, $n_h \geq 1$

(d) Absence of exotic fractional charge states, $n_e = 0$

(e) Existence of top Yukawa coupling

- a)+b)+c) = 1 : 10,000

This search space is (just about) searchable deterministically so we can compare the two methods. Assel, Christodoulides, Faraggi, Kounnas, Rizos

The phases determine the characteristics of the models

(a) 3 complete family generations, $n_g = 3$

(b) Existence of PS breaking Higgs, $k_R \geq 1$

(c) Existence of SM Higgs doublets, $n_h \geq 1$

(d) Absence of exotic fractional charge states, $n_e = 0$

(e) Existence of top Yukawa coupling

- a)+b)+c) = 1 : 10,000
- a)+b)+c)+d) = 1 : 2,500,000

This search space is (just about) searchable deterministically so we can compare the two methods. Assel, Christodoulides, Faraggi, Kounnas, Rizos

The phases determine the characteristics of the models

(a) 3 complete family generations, $n_g = 3$

(b) Existence of PS breaking Higgs, $k_R \geq 1$

(c) Existence of SM Higgs doublets, $n_h \geq 1$

(d) Absence of exotic fractional charge states, $n_e = 0$

(e) Existence of top Yukawa coupling

- a)+b)+c) = 1 : 10,000
- a)+b)+c)+d) = 1 : 2,500,000
- a)+b)+c)+d)+e) = 1 : 10,000,000,000

This search space is (just about) searchable deterministically so we can compare the two methods. Assel, Christodoulides, Faraggi, Kounnas, Rizos

The phases determine the characteristics of the models

(a) 3 complete family generations, $n_g = 3$

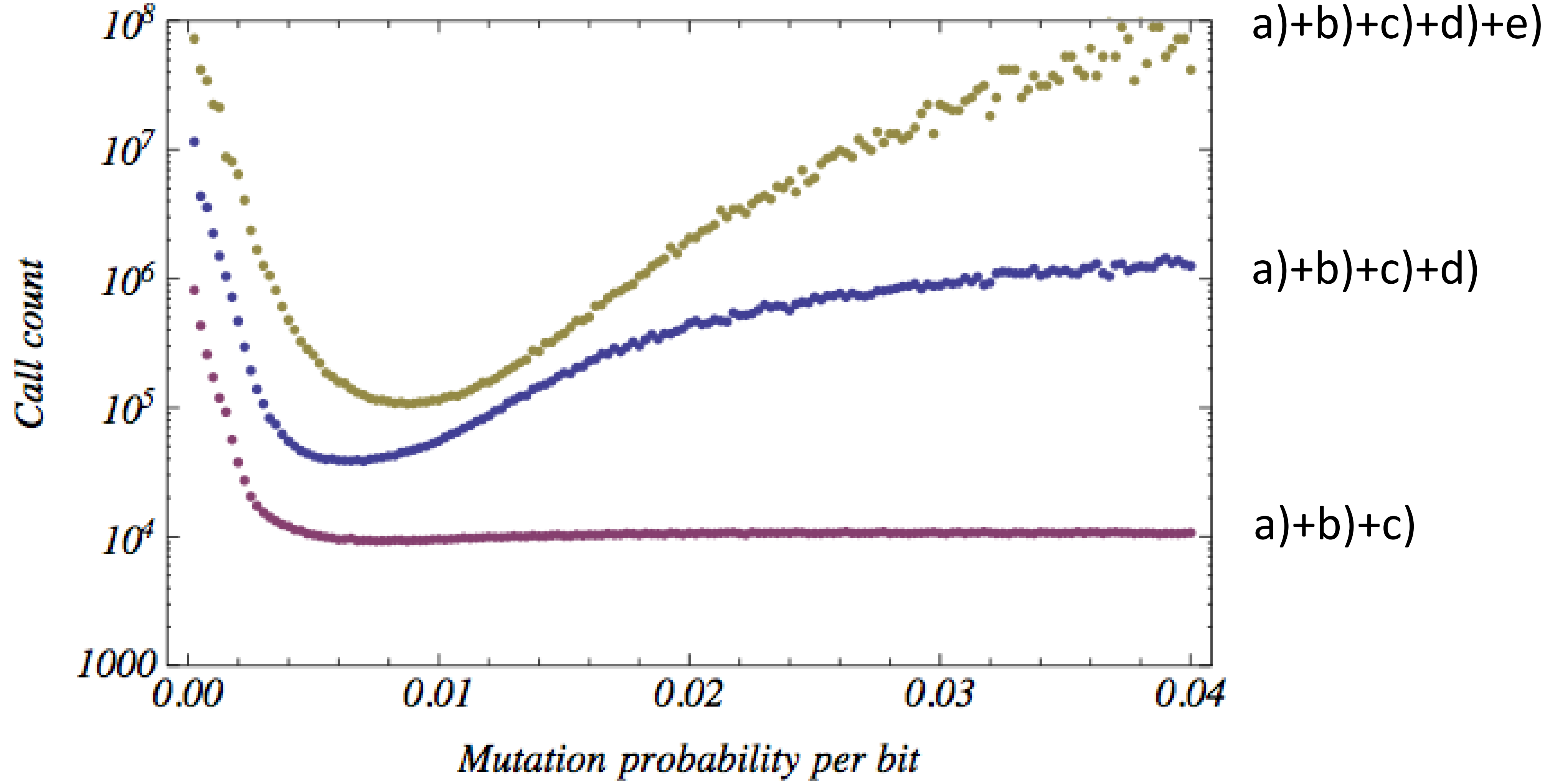
(b) Existence of PS breaking Higgs, $k_R \geq 1$

(c) Existence of SM Higgs doublets, $n_h \geq 1$

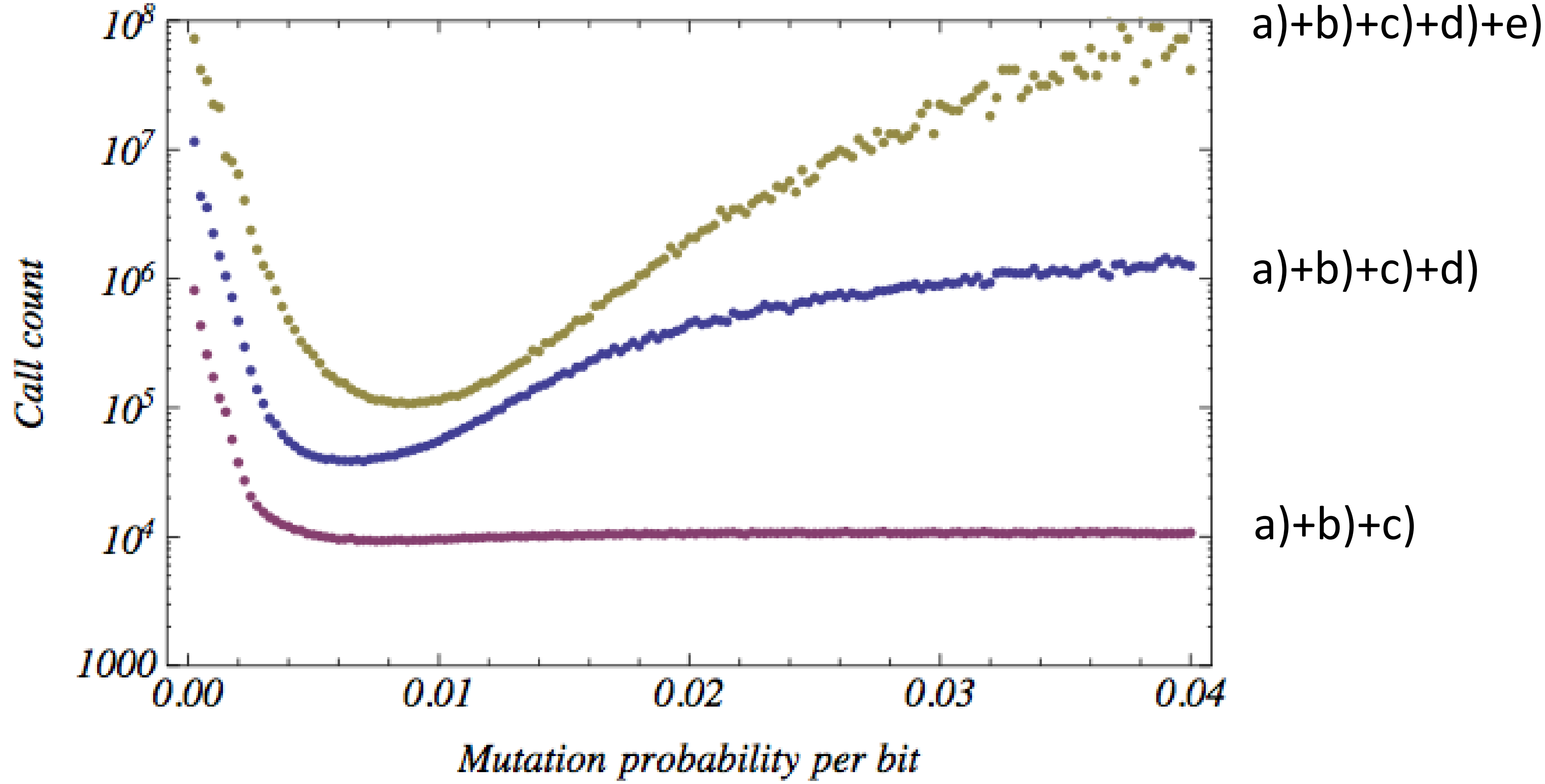
(d) Absence of exotic fractional charge states, $n_e = 0$

(e) Existence of top Yukawa coupling

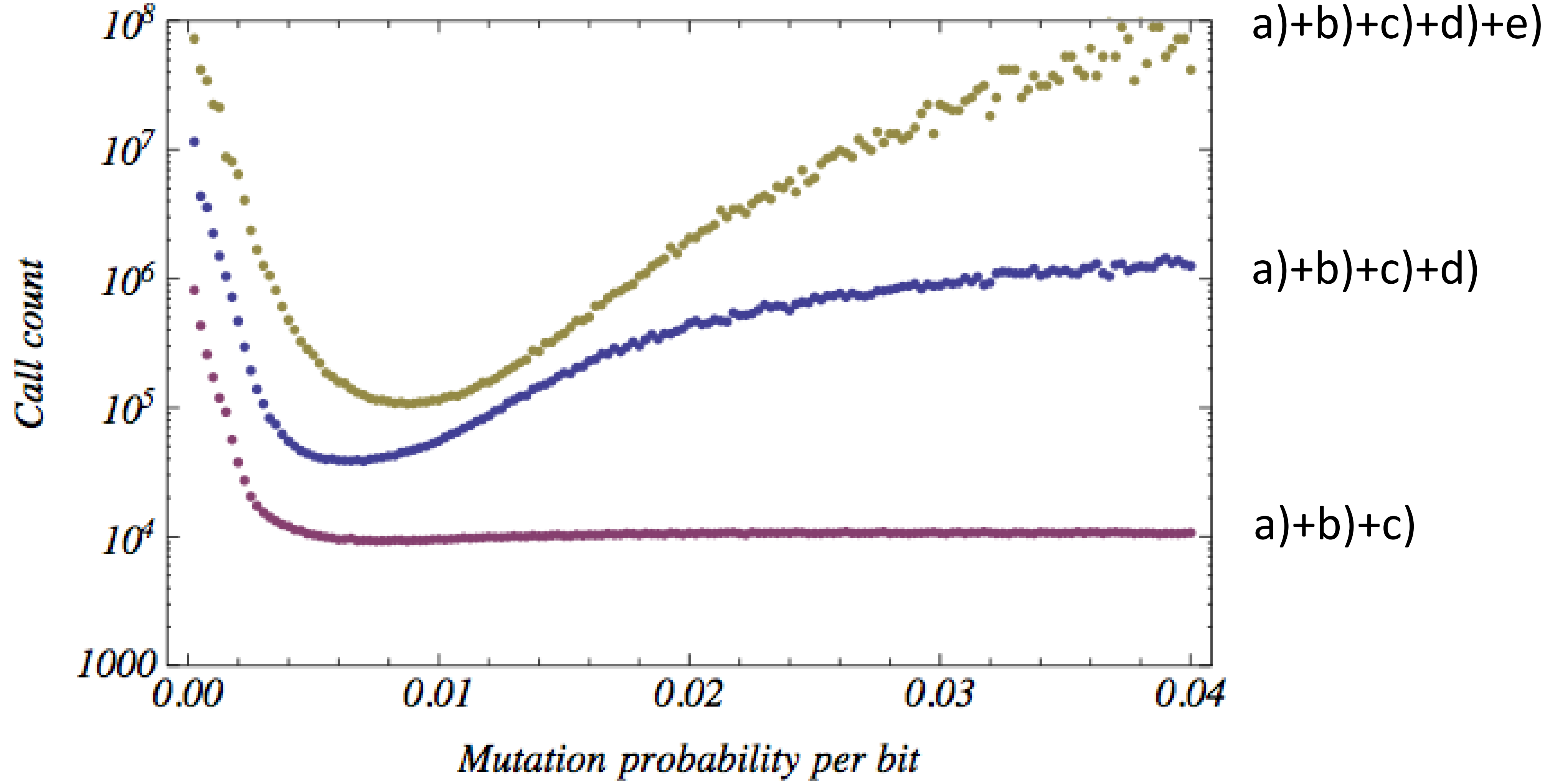
- a)+b)+c) = 1 : 10,000
- a)+b)+c)+d) = 1 : 2,500,000
- a)+b)+c)+d)+e) = 1 : 10,000,000,000
- deterministically we would expect to have to construct 10 billion models to find an example of the latter



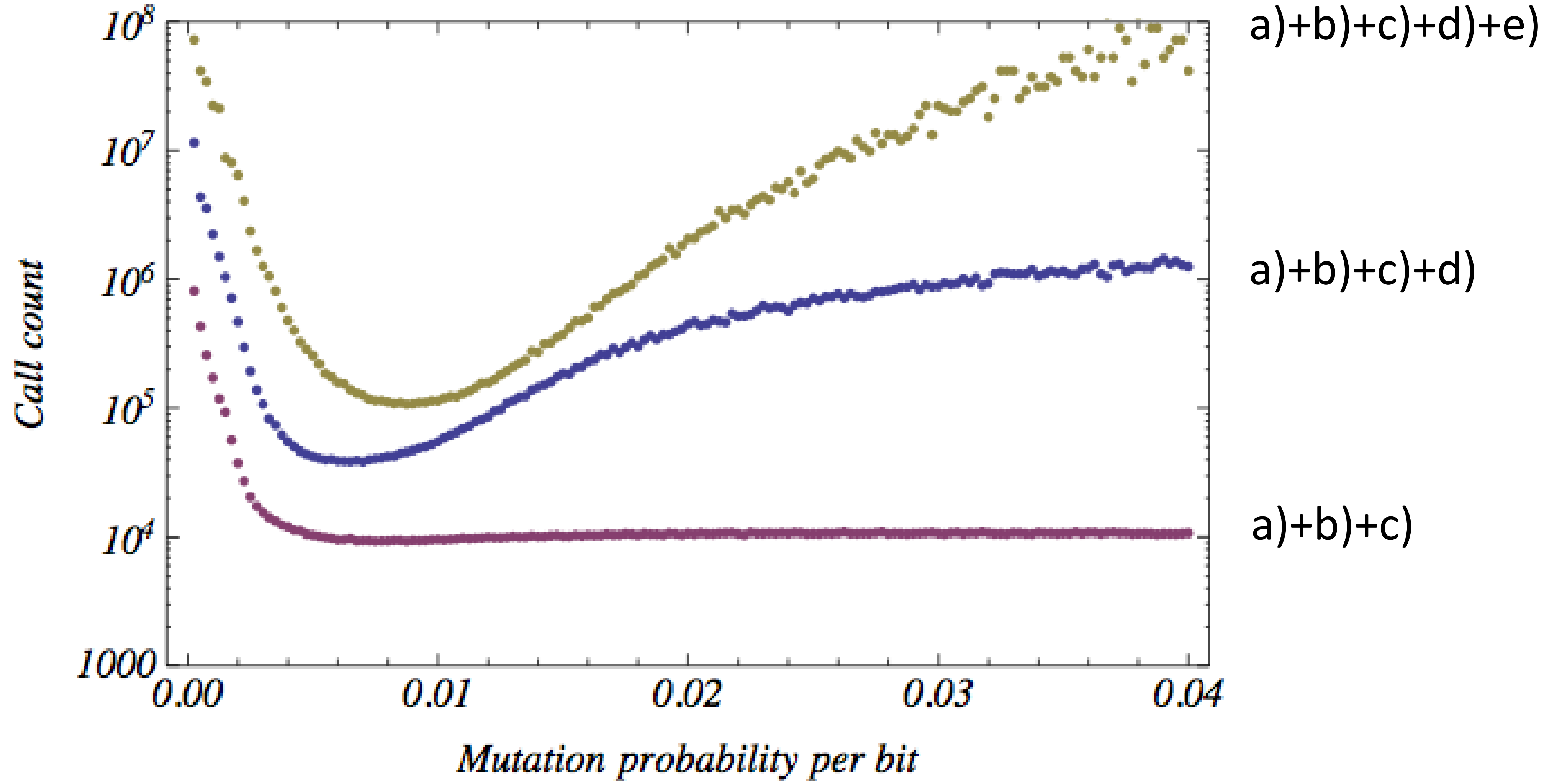
10^8
10^7
10^6
10^5
10^4
1000
0.00
0.01
0.02
0.03
0.04
Call count
Mutation probability per bit
a)+b)+c)+d)+e)
a)+b)+c)+d)
a)+b)+c)



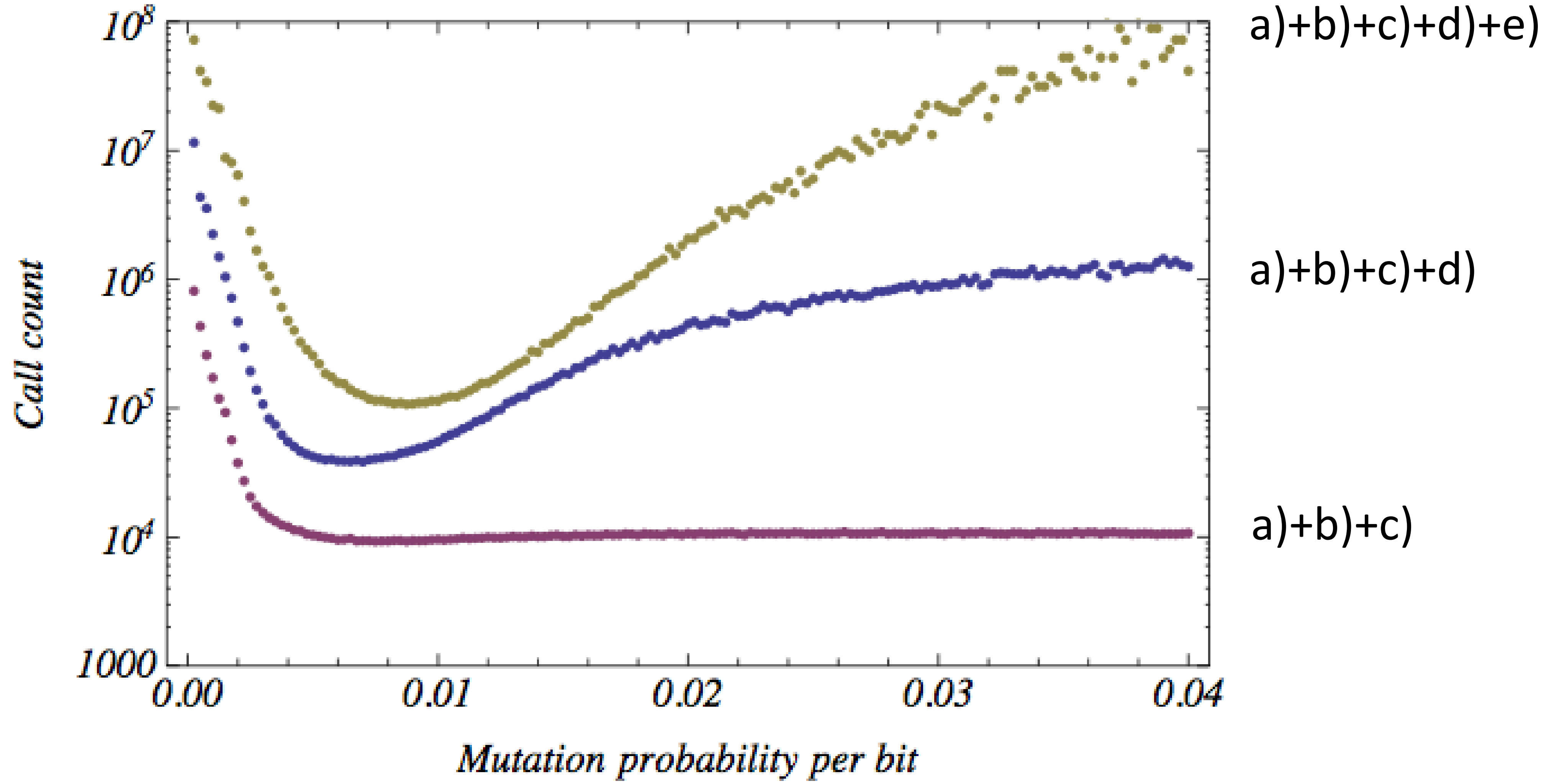
- Optimum mutation rate => genetic algorithm is working as expected



- Optimum mutation rate => genetic algorithm is working as expected
- GA's do not confer much advantage when the search is "easy"



- Optimum mutation rate => genetic algorithm is working as expected
- GA's do not confer much advantage when the search is "easy"
- They work best when there are many criteria and the search is difficult



- Optimum mutation rate => genetic algorithm is working as expected
- GA's do not confer much advantage when the search is "easy"
- They work best when there are many criteria and the search is difficult

GAs versus reinforcement learning

SAA, Constantin, Lukas, Harvey

(c.f. recent work by Cole, Schachner, Shiu; Cole, Krippendorf, Schachner, Shiu; Loges, Shiu)

- First comparison of GA versus RL in string context (NB techniques both work with “environment”)
- Consider kind of string construction with a much larger space that has been the subject of intensive ML scrutiny: “Monad bundles on Complete Intersection Calabi Yaus”. Distler, Greene; Kachru; Anderson; Anderson, He, Lukas; Anderson, Gray, He, Lukas; He, Lee, Lukas
- Considered the following two kinds of CICY (*bi-cubic* and *triple trilinear* respectively) with configuration matrices, where indices are h11, h21, and Euler number (for the experts):

$$\left[\begin{array}{c|c} \mathbb{P}^2 & 3 \\ \mathbb{P}^2 & 3 \end{array}\right]^{2,83}_{-162}, \qquad \left[\begin{array}{c|ccc} \mathbb{P}^2 & 1 & 1 & 1 \\ \mathbb{P}^2 & 1 & 1 & 1 \\ \mathbb{P}^2 & 1 & 1 & 1 \end{array}\right]^{3,48}_{-90}$$

- Models constructed by monad bundles on the CICY defining the E8xE8 background: constructed from two line-bundle sums, *B* and *C: in the end boils down to matrix of integers (where k=1,..,h11):*

$$(b_1^k, \ldots, b_{r_B}^k, c_1^k, \ldots, c_{r_C}^k)$$

- As well as various consistency conditions (e.g. anomaly cancellation), all the phenomenological properties (e.g. number of generations) determined by these numbers via (several) index theorems. Constantin, Lukas, Harvey

- Search for “perfect-models” (aka “terminal states”): require SM-like theories (i.e. SO(10) GUT from broken E8, with 3 generations).

- *So what is the size of search space?* If we take $b_{\rm min} \leq b_i^k \leq b_{\rm max}$, $c_{\rm min} \leq c_a^k \leq c_{\rm max}$,

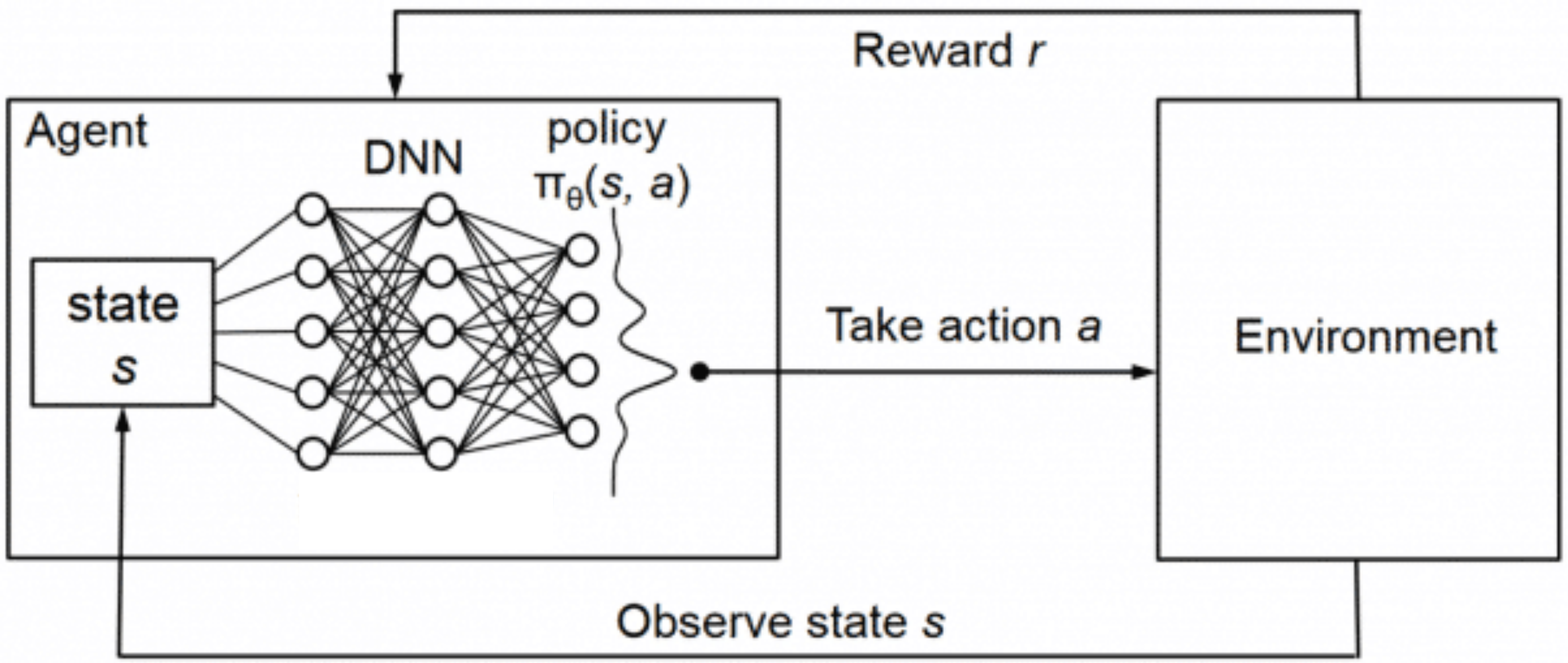
- Allowing say 10 values per entry, that is $10^{h^{1,1}(r_B+r_C-1)}$
  with say h11=3 it becomes huge very quickly!

- For the GA we simply encode these integers as a single binary string and operate as before. Used quite large population = 250.

- In both RL and GA we use the same function to stand for the reward / fitness, based on the number of criteria that are satisfied.

Reinforcement learning vs GAs for these models

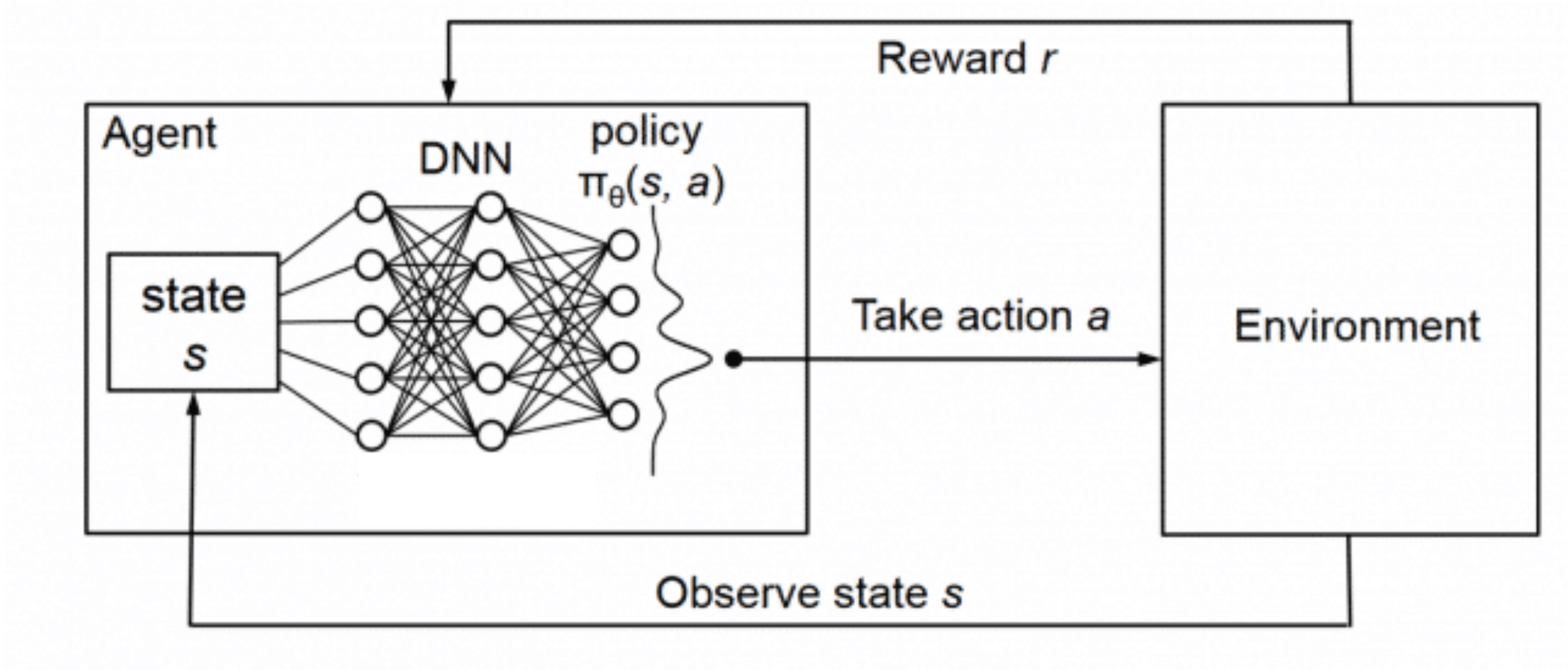
These models were already shown to be amenable to RL. Constantin, Lukas, Harvey



Reinforcement learning vs GAs for these models

These models were already shown to be amenable to RL. Constantin, Lukas, Harvey

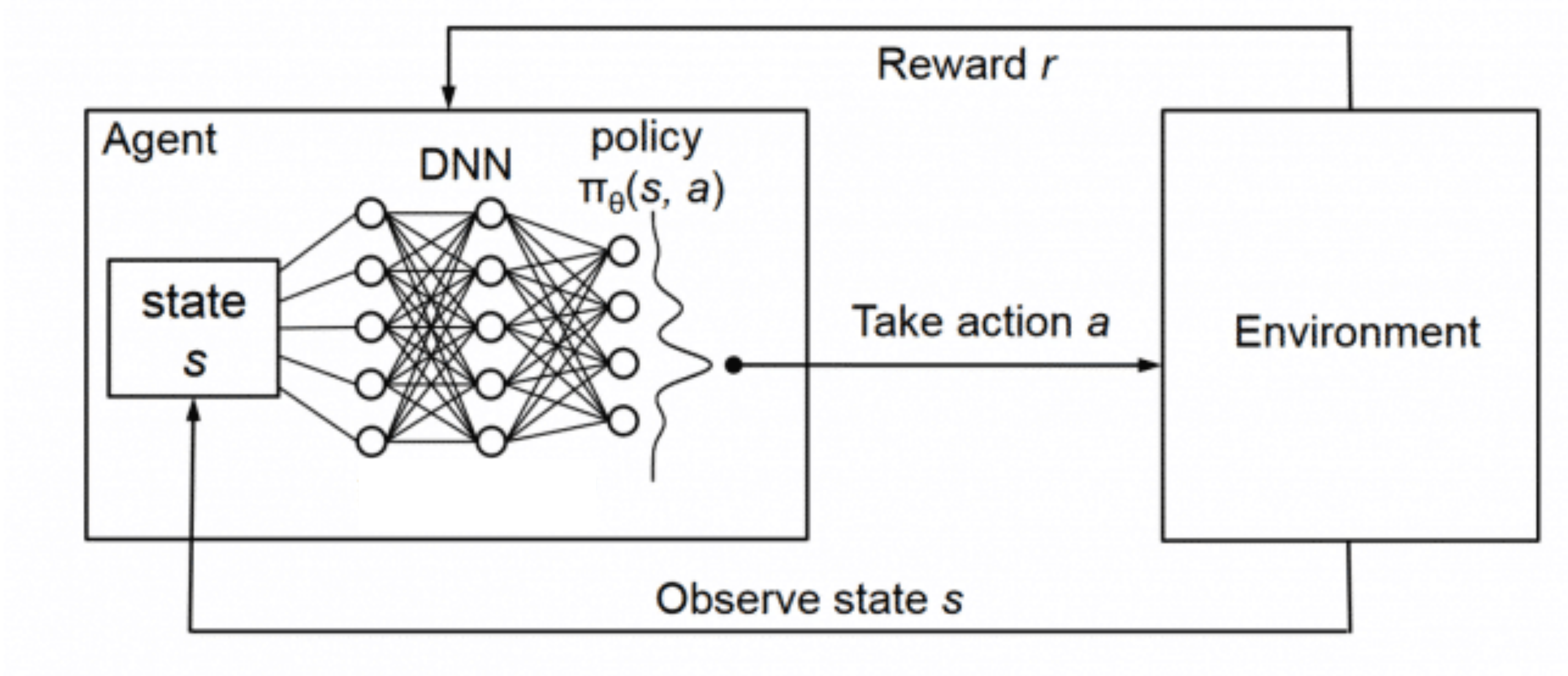
using REINFORCE …



Reinforcement learning vs GAs for these models

These models were already shown to be amenable to RL. Constantin, Lukas, Harvey

using REINFORCE …



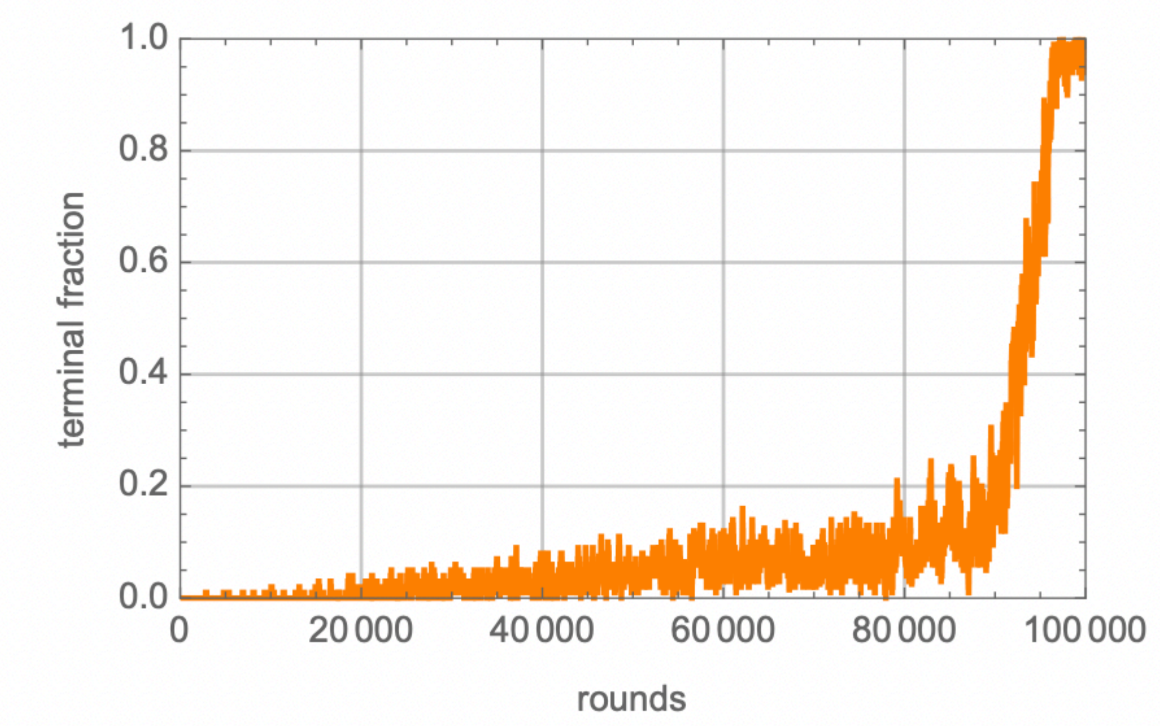
Reinforcement learning vs GAs for these models

These models were already shown to be amenable to RL. Constantin, Lukas, Harvey

We find good performance after a long training time: typical run on the (6,2) bi-cubic with

$$b_{\rm min} = -3,\; c_{\rm min} = 0 \qquad b_{\rm max} = 4,\; c_{\rm max} = 7$$

for which the search space is $\simeq 4.4 \times 10^{12}$



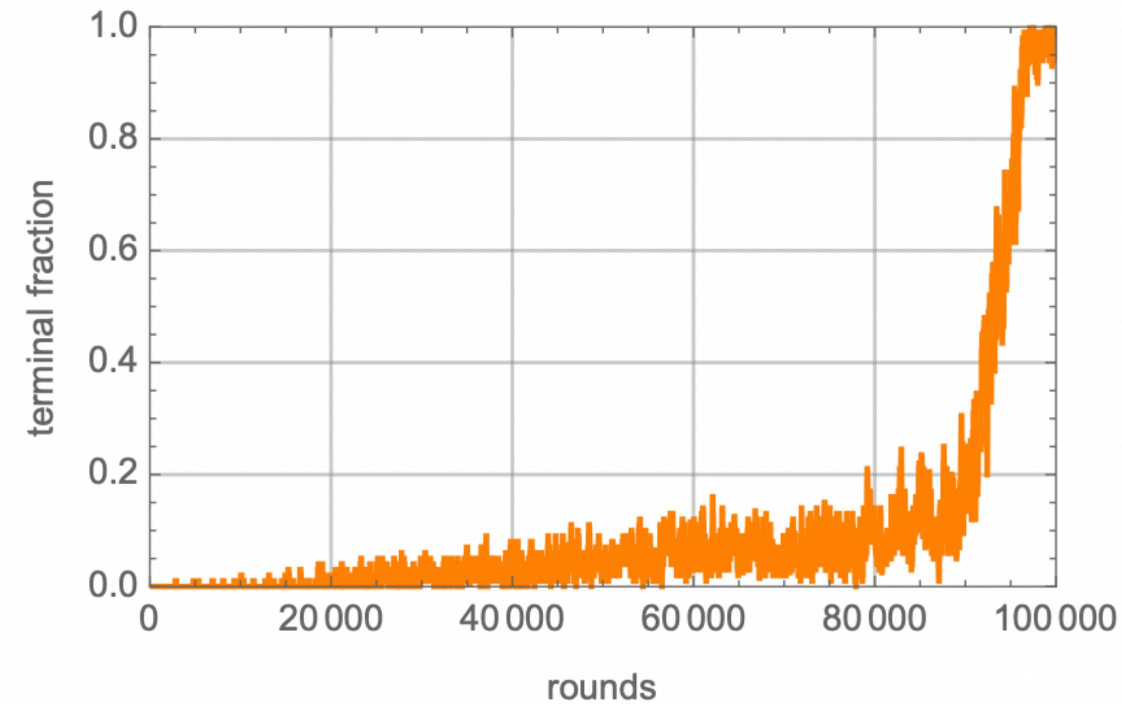
Reinforcement learning vs GAs for these models

These models were already shown to be amenable to RL. Constantin, Lukas, Harvey

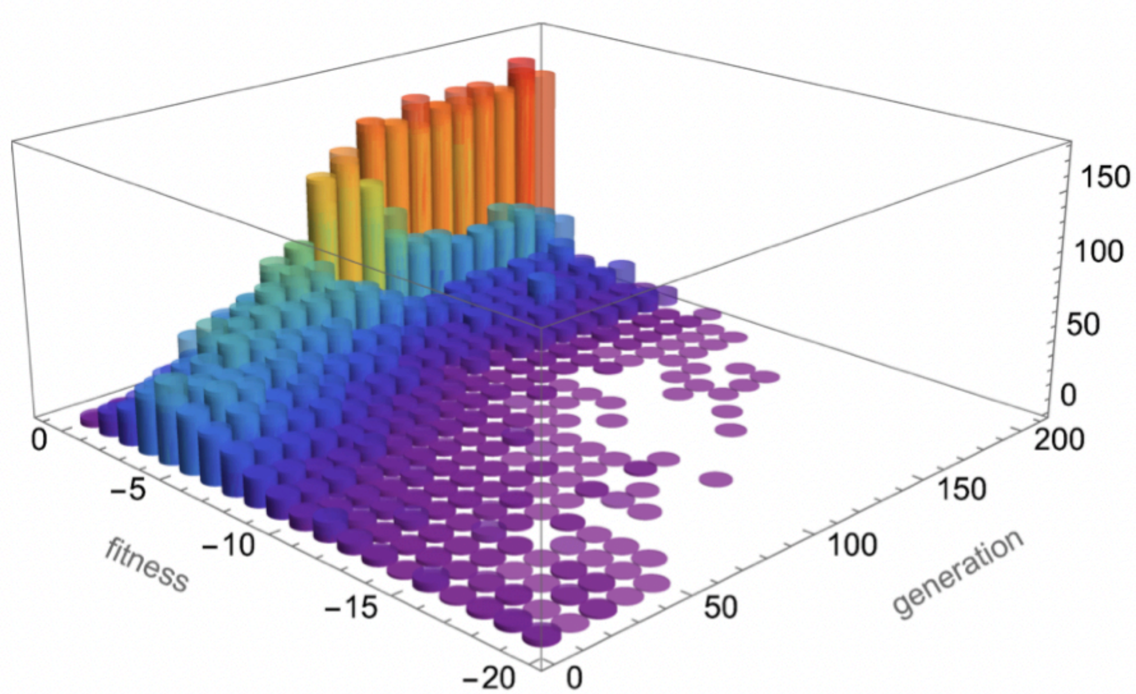
We find good performance after a long training time: typical run on the (6,2) bi-cubic with

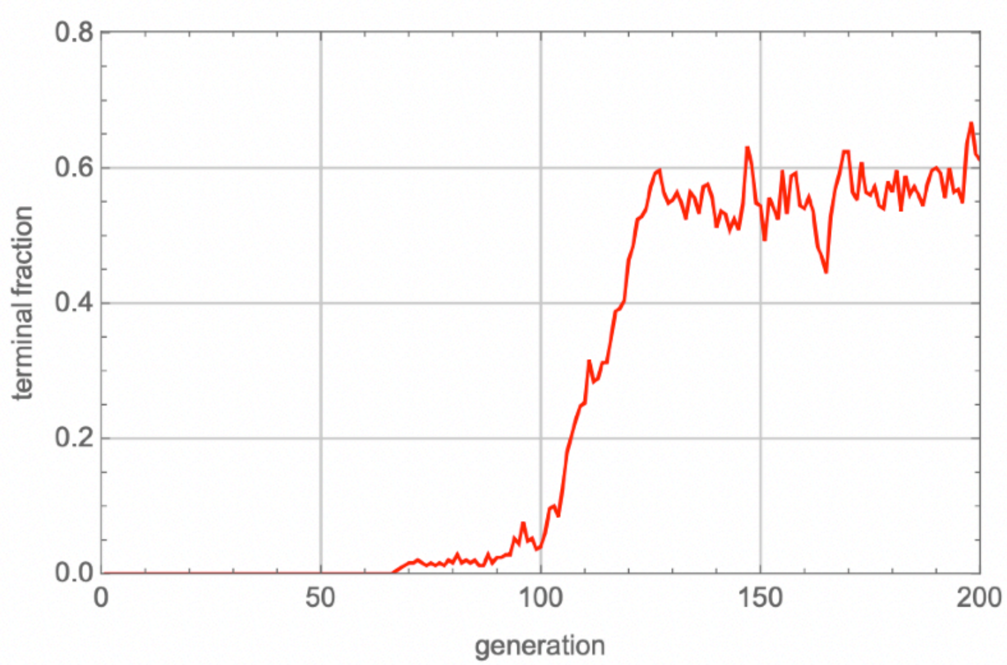
$$b_{\text{min}} = -3,\ c_{\text{min}} = 0 \qquad b_{\text{max}} = 4,\ c_{\text{max}} = 7,$$

for which the search space is $\simeq 4.4 \times 10^{12}$



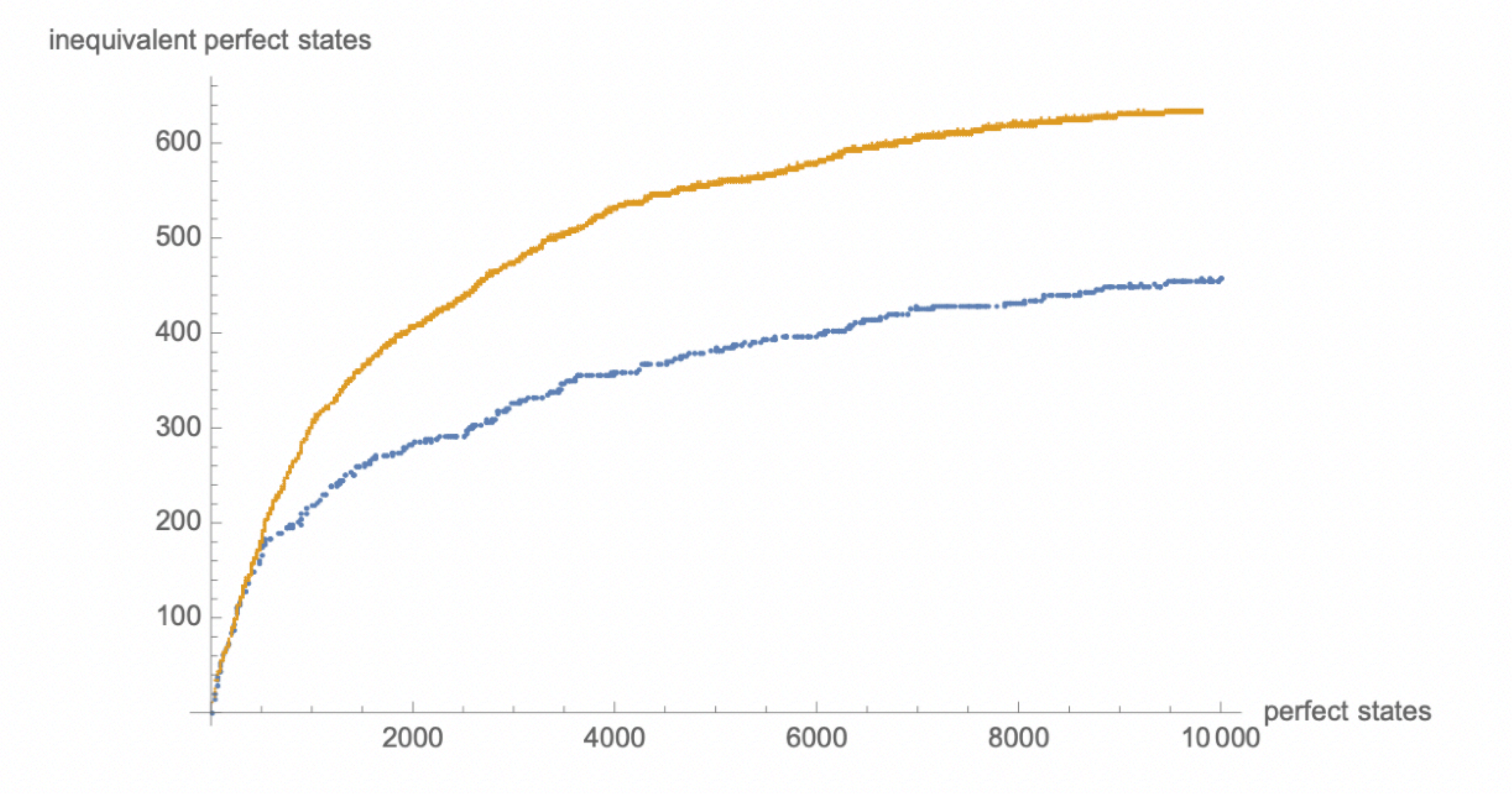
The GA is much faster to the first solutions! Note only 50K states visited:





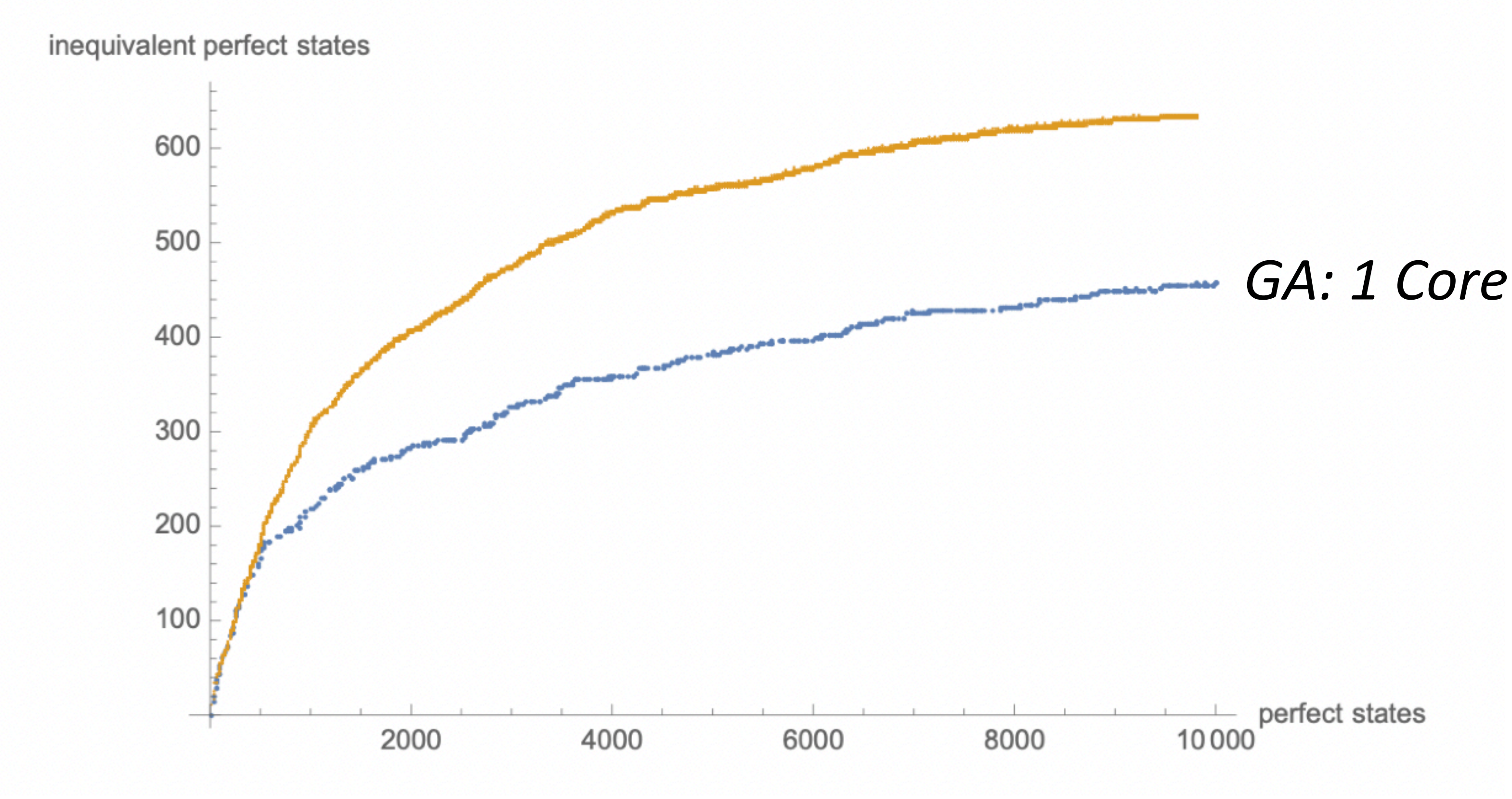
Reinforcement learning vs GAs for these models

Redundancy: The methods behave differently. GA's tend to produce a lot of redundancy (equivalent perfect states) due to convergence, but are still more efficient:



Reinforcement learning vs GAs for these models

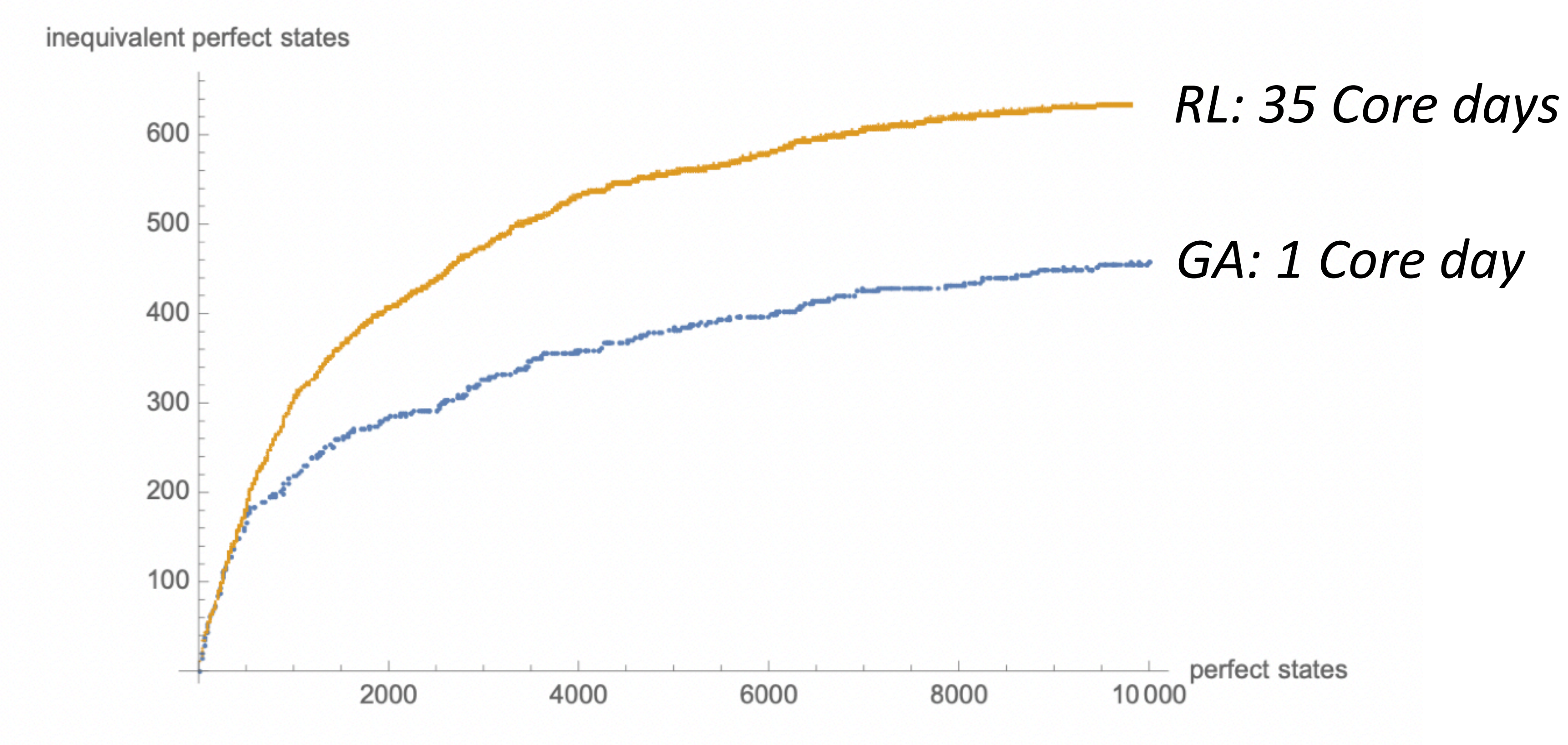
Redundancy: The methods behave differently. GA's tend to produce a lot of redundancy (equivalent perfect states) due to convergence, but are still more efficient:



GA: 1 Core day

Reinforcement learning vs GAs for these models

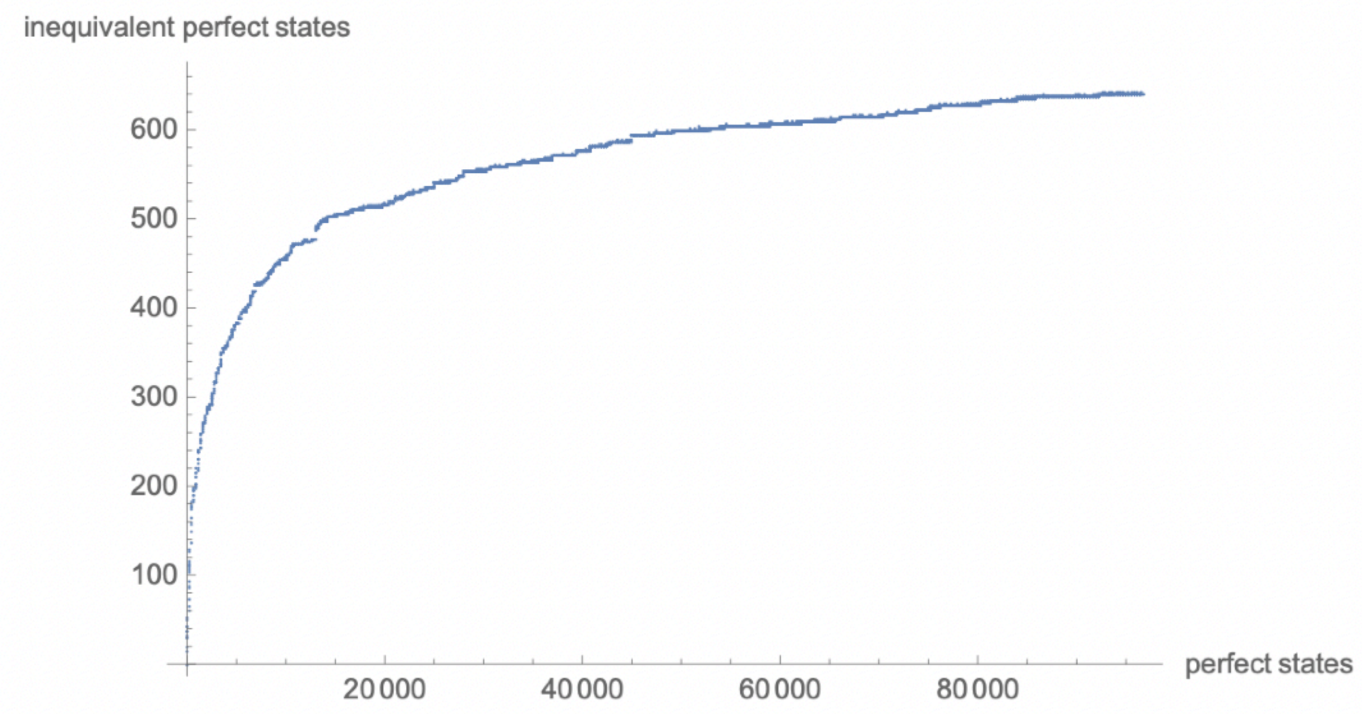
Redundancy: The methods behave differently. GA's tend to produce a lot of redundancy (equivalent perfect states) due to convergence, but are still more efficient:



Reinforcement learning vs GAs for these models

Saturation: after 35 core days the RL produced 643 inequivalent perfect states. After 10 core days the GA saturated at 639 inequivalent perfect states.

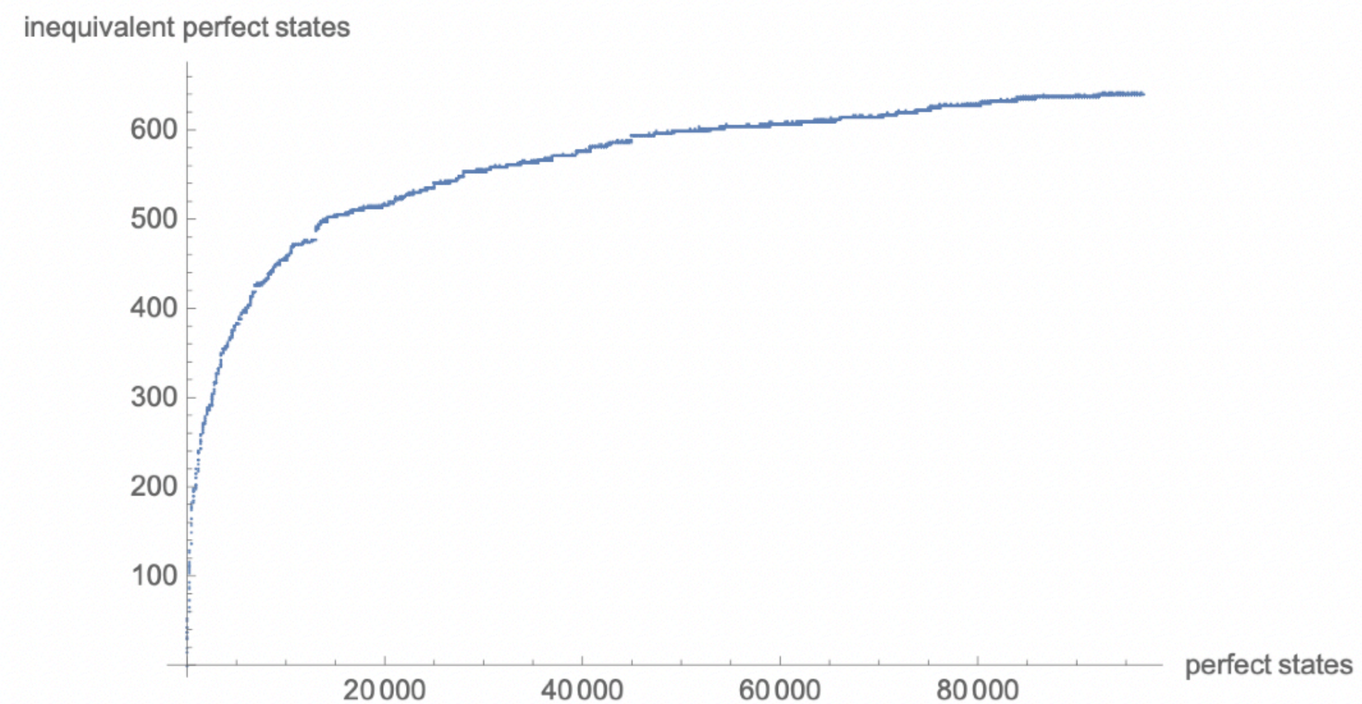
About 50 models in complement (i.e. 689 models in total)



Reinforcement learning vs GAs for these models

Saturation: after 35 core days the RL produced 643 inequivalent perfect states. After 10 core days the GA saturated at 639 inequivalent perfect states.

About 50 models in complement (i.e. 689 models in total)

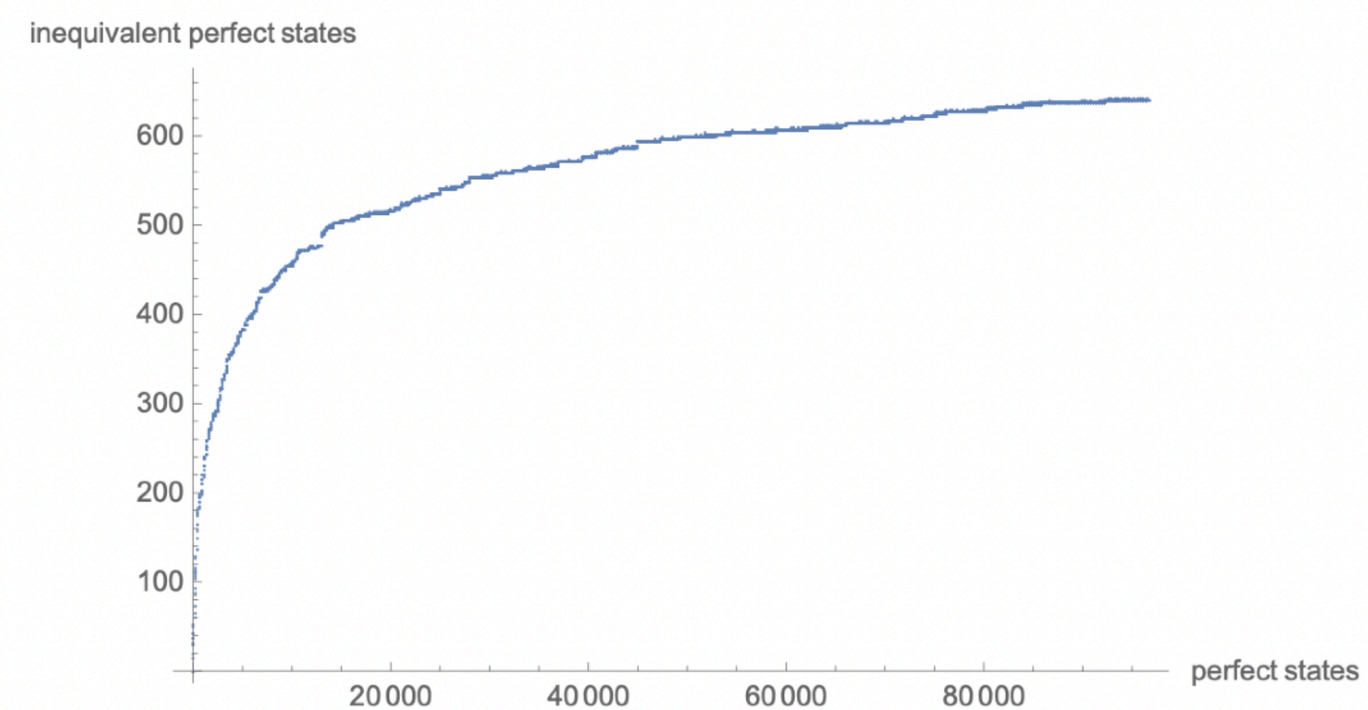


NB: at the beginning they cover different regions (Sammon mapping), so an important side-effect is that we have evidence almost all possibilities are saturated, for this choice of hyper-parameters .

Reinforcement learning vs GAs for these models

Saturation: after 35 core days the RL produced 643 inequivalent perfect states. After 10 core days the GA saturated at 639 inequivalent perfect states.

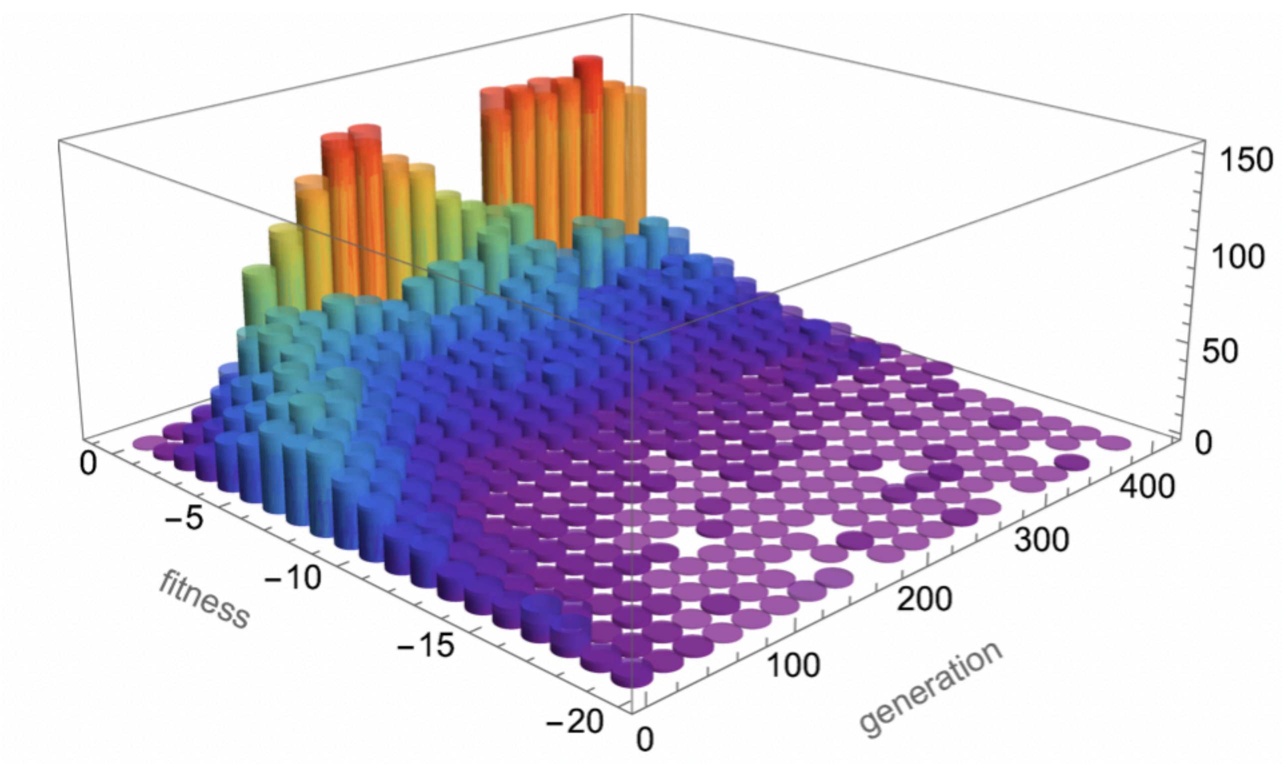
About 50 models in complement (i.e. 689 models in total)

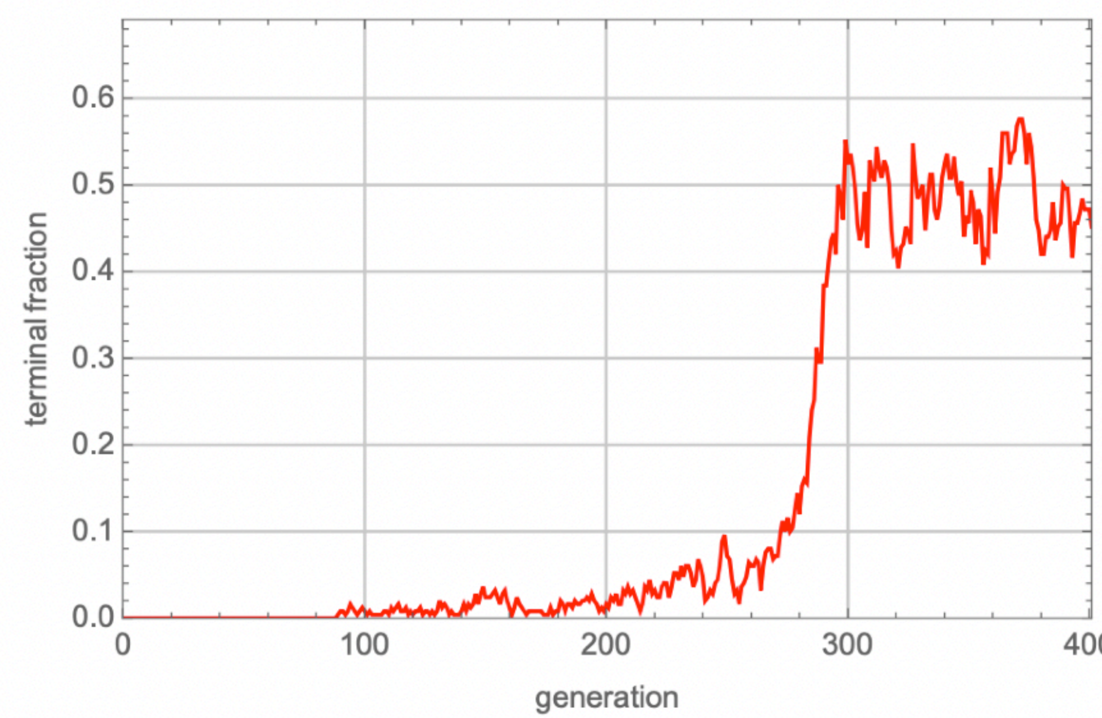


NB: at the beginning they cover different regions (Sammon mapping), so an important side-effect is that we have evidence almost all possibilities are saturated, for this choice of hyper-parameters .

Reinforcement learning vs GAs for these models

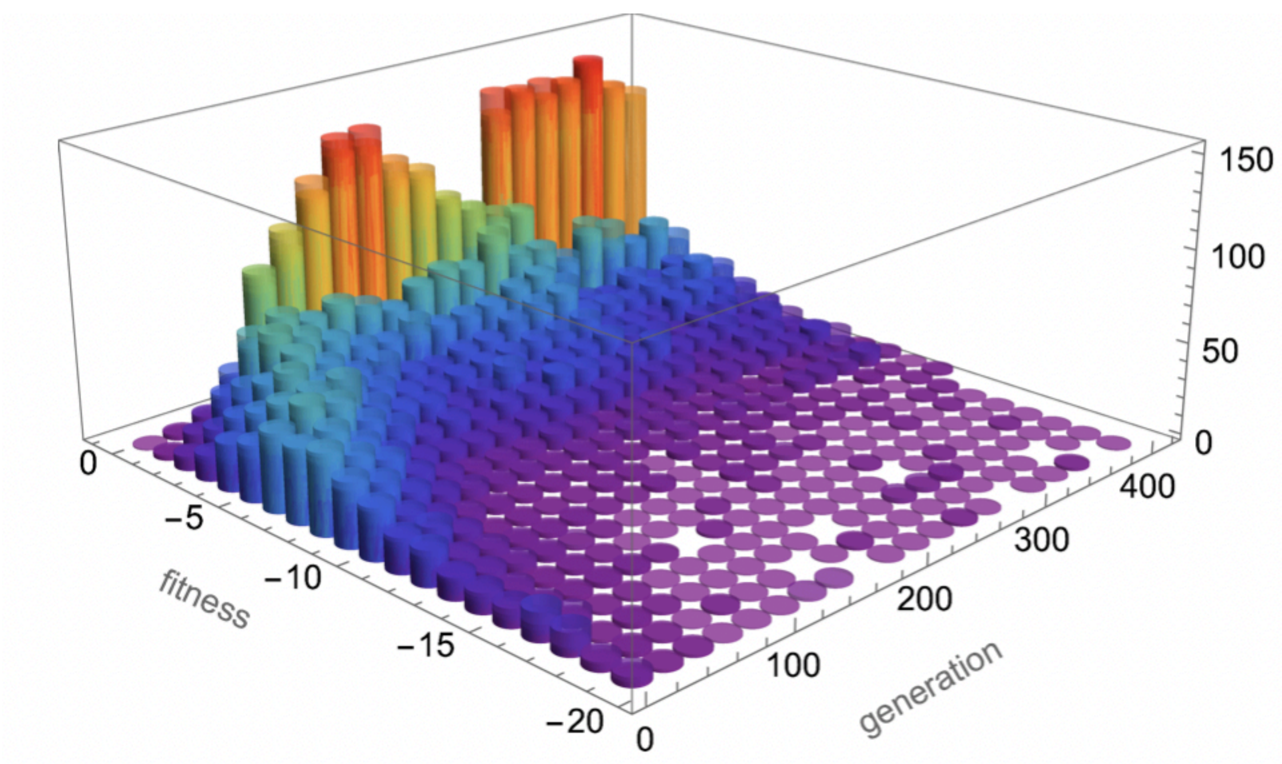
(6,2) triple trilinear. Keep same domains of defining integers, but now h11=3 gives search space $8^{21} \simeq 10^{19}$ is seven orders of magnitude larger!

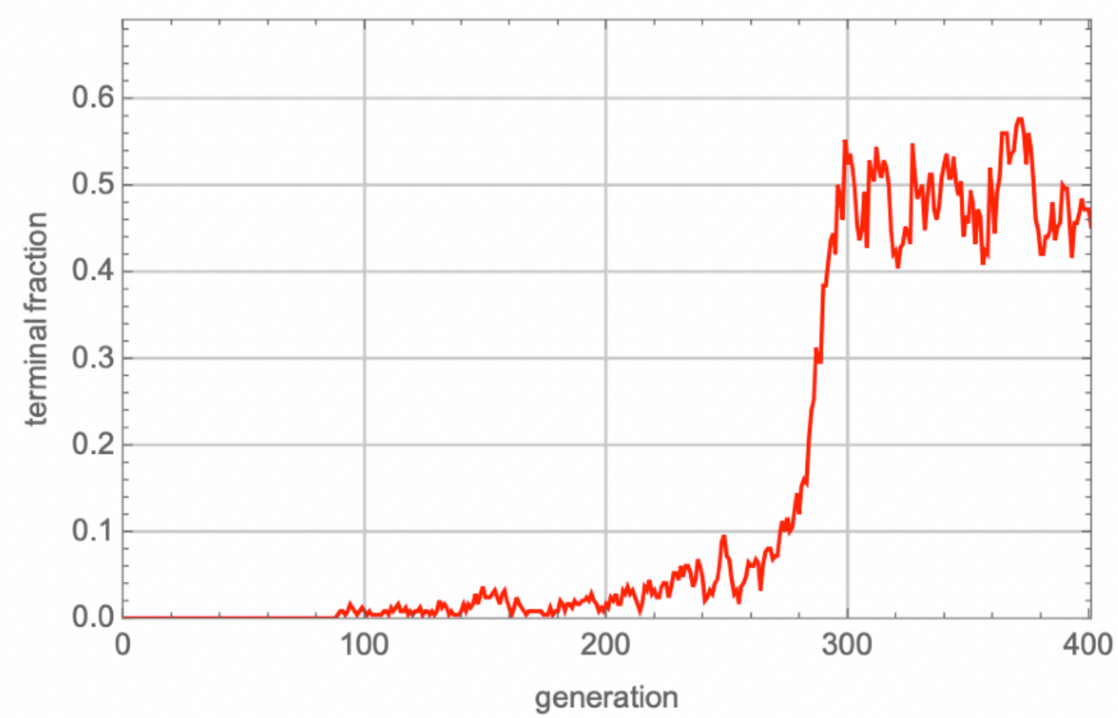




Reinforcement learning vs GAs for these models

(6,2) triple trilinear. Keep same domains of defining integers, but now h11=3 gives search space $8^{21} \simeq 10^{19}$ is seven orders of magnitude larger!



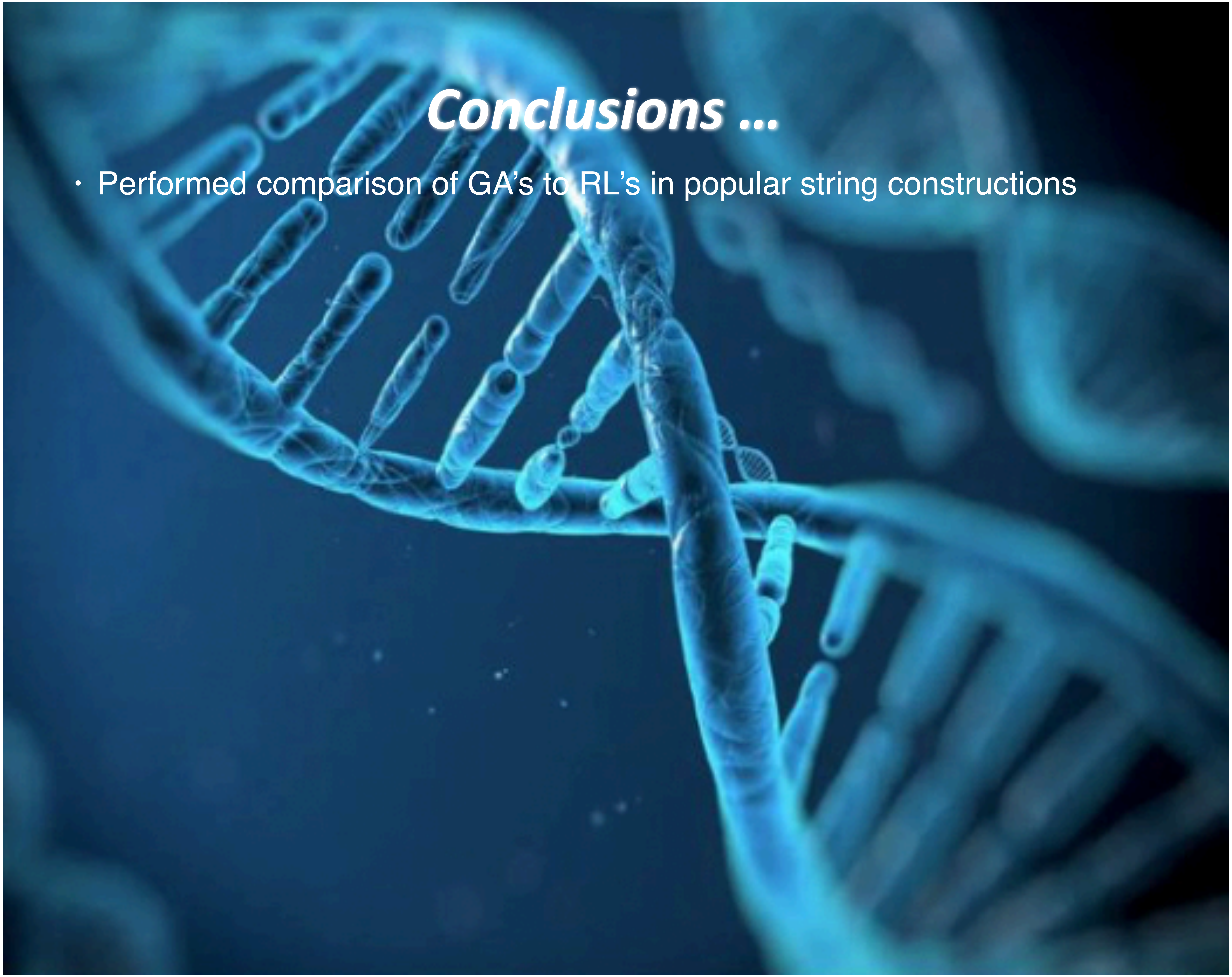


GA in a given run takes only twice as many generations to reach the saturated fitness.


Conclusions …

Conclusions …

- Performed comparison of GA's to RL's in popular string constructions



Conclusions …

- Performed comparison of GA’s to RL’s in popular string constructions
- GA’s remain a strikingly effective search tool for finding favourable string vacua

Conclusions …

- Performed comparison of GA’s to RL’s in popular string constructions
- GA’s remain a strikingly effective search tool for finding favourable string vacua
- Note we did not yet work hard to optimise the GA in the RL/GA study. (Just single point cross-over, not particularly optimising mutation rate, no creep mutation etc)

Conclusions …

- Performed comparison of GA's to RL's in popular string constructions
- GA's remain a strikingly effective search tool for finding favourable string vacua
- Note we did not yet work hard to optimise the GA in the RL/GA study. (Just single point cross-over, not particularly optimising mutation rate, no creep mutation etc)
- Results suggest estimates of string-landscape size are less meaningful than fitness-distance correlation (i.e. SM is not a needle in a haystack)

Conclusions …

- Performed comparison of GA's to RL's in popular string constructions
- GA's remain a strikingly effective search tool for finding favourable string vacua
- Note we did not yet work hard to optimise the GA in the RL/GA study. (Just single point cross-over, not particularly optimising mutation rate, no creep mutation etc)
- Results suggest estimates of string-landscape size are less meaningful than fitness-distance correlation (i.e. SM is not a needle in a haystack)
- A combined GA+RL approach can indicate when a portion of the landscape has been comprehensively searched for SM models. This is not possible with either method alone or by doing any kind of scan (for NP-hard).

Conclusions …

- Performed comparison of GA’s to RL’s in popular string constructions
- GA’s remain a strikingly effective search tool for finding favourable string vacua
- Note we did not yet work hard to optimise the GA in the RL/GA study. (Just single point cross-over, not particularly optimising mutation rate, no creep mutation etc)
- Results suggest estimates of string-landscape size are less meaningful than fitness-distance correlation (i.e. SM is not a needle in a haystack)
- A combined GA+RL approach can indicate when a portion of the landscape has been comprehensively searched for SM models. This is not possible with either method alone or by doing any kind of scan (for NP-hard).
- Search difficulty does not seem to be increasing proportionally to difficulty (scaling behaviour remains to be determined)

Conclusions …

- Performed comparison of GA's to RL's in popular string constructions
- GA's remain a strikingly effective search tool for finding favourable string vacua
- Note we did not yet work hard to optimise the GA in the RL/GA study. (Just single point cross-over, not particularly optimising mutation rate, no creep mutation etc)
- Results suggest estimates of string-landscape size are less meaningful than fitness-distance correlation (i.e. SM is not a needle in a haystack)
- A combined GA+RL approach can indicate when a portion of the landscape has been comprehensively searched for SM models. This is not possible with either method alone or by doing any kind of scan (for NP-hard).
- Search difficulty does not seem to be increasing proportionally to difficulty (scaling behaviour remains to be determined)