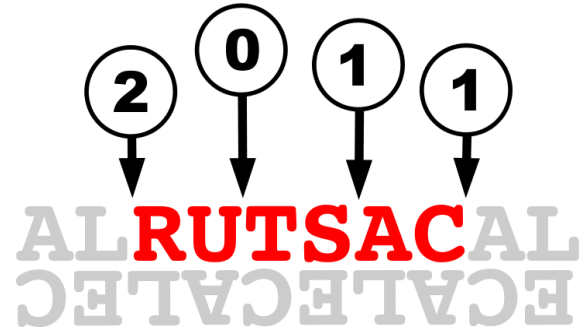


Research Using The Stringmol Artificial Chemistry

Edward Clark, Simon Hickinbotham, Adam Nellis

YCCSA, University of York, Heslington, York YO1 5DD, UK
email: sjh@cs.york.ac.uk

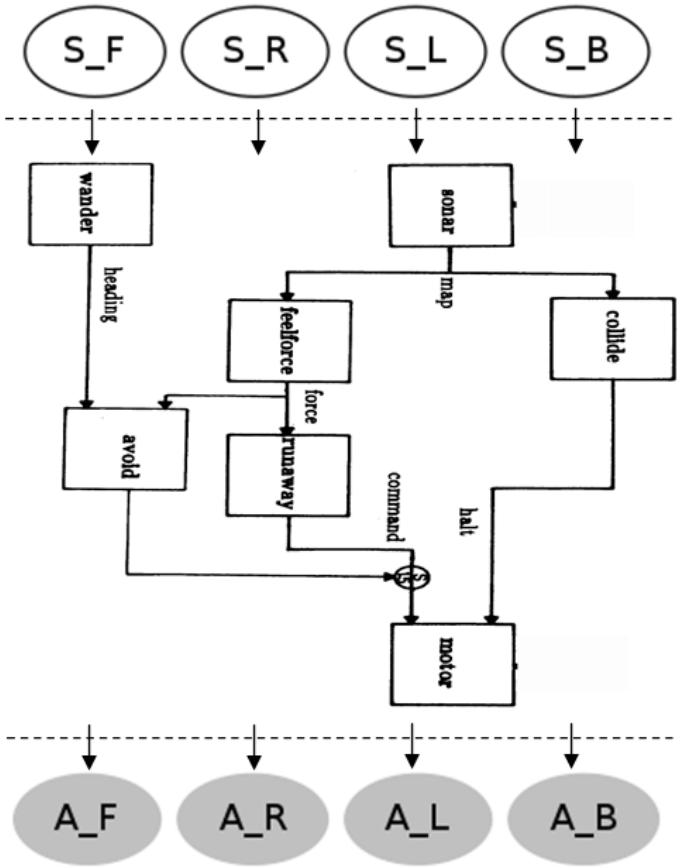
EPSRC grant no. EP/F031033/1

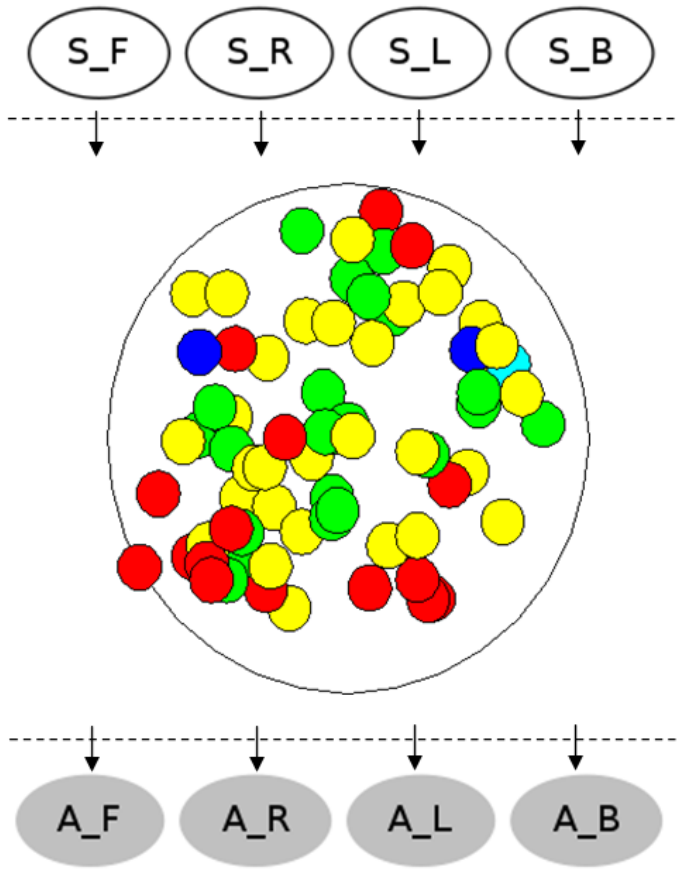
Summary

Artificial Life (ALife) is a bottom-up approach to Artificial Intelligence (AI)
Artificial Chemistry (AChem) is a bottom-up approach to building ALife

- We have devised an artificial chemistry that is encoded such that it can evolve
- A novel combination of a stochastic chemistry model with Instruction-Set based A-Life
- We have implemented a string-based molecular analogue which does
  - mutation-on-copy
  - specific binding
  - stochastic reaction chemistry

Metabolic model



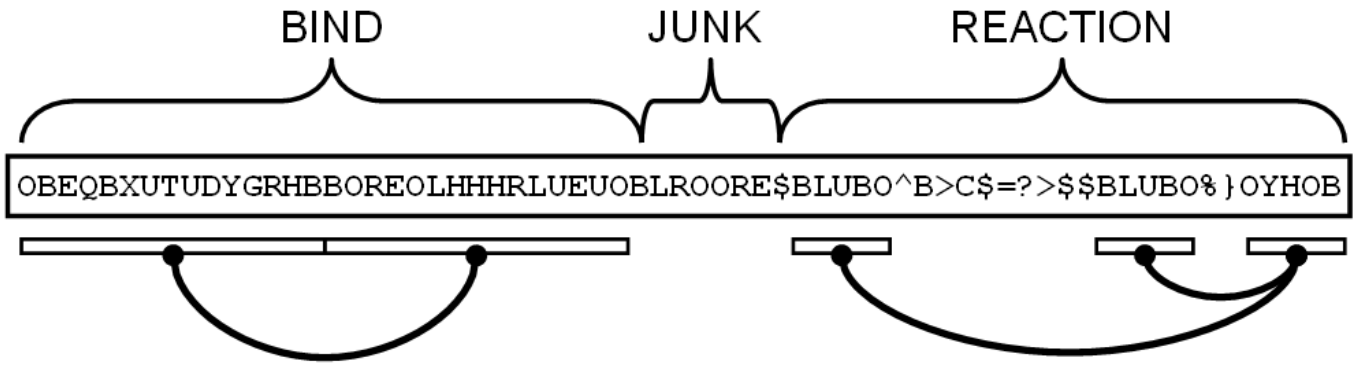


Demo 1: Replicase population

```
''http://www.plazzmid.org/RUTSAC11/
StringmolWeb/StableVersion/index.html''

''http://tinyurl.com/3b997dy''
```

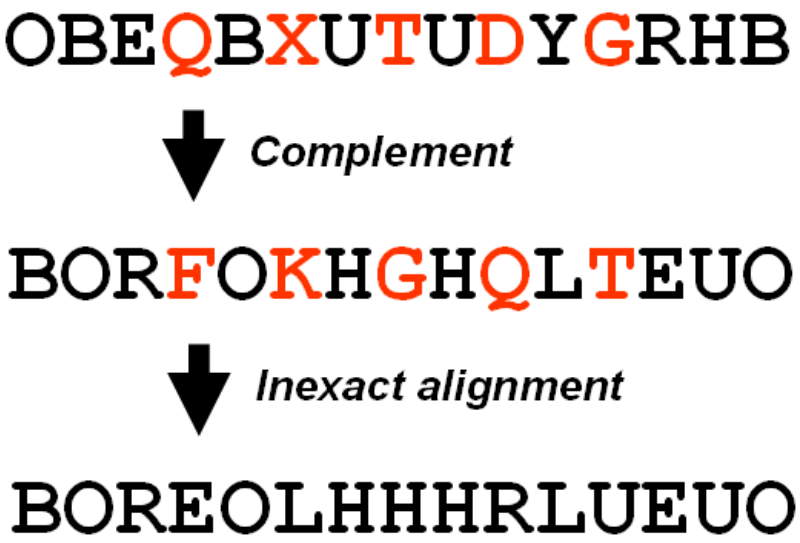
Stringmol



- String-based programming language
- Stochastic chemical “mixer”
- **Probabilistic binding functionality**
- Smith-Waterman-like alignments
- Two symbol types:
  - Templates $T = \{$ ‘**A**’,... ‘**Z**’ $\}$
  - Functional: $\Phi = \{$ ‘**$**’, ‘**>**’, ‘**^**’, ‘**?**’, ‘**=**’, ‘**%**’, ‘**}**’ $\}$
- Four pointer types: **I**nstruction; **F**low; **R**ead; **W**rite

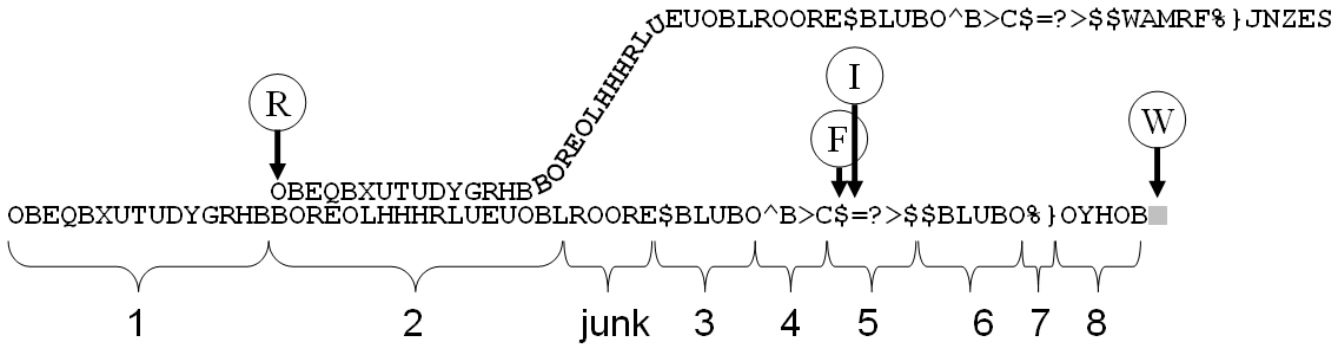
Why the emphasis on soft binding?

- Basis of all interesting DNA interactions
- Transposons, promotors etc. *all* use specific binding
- Therefore *necessary* for phenotypic control of gene expression



Binding molecules

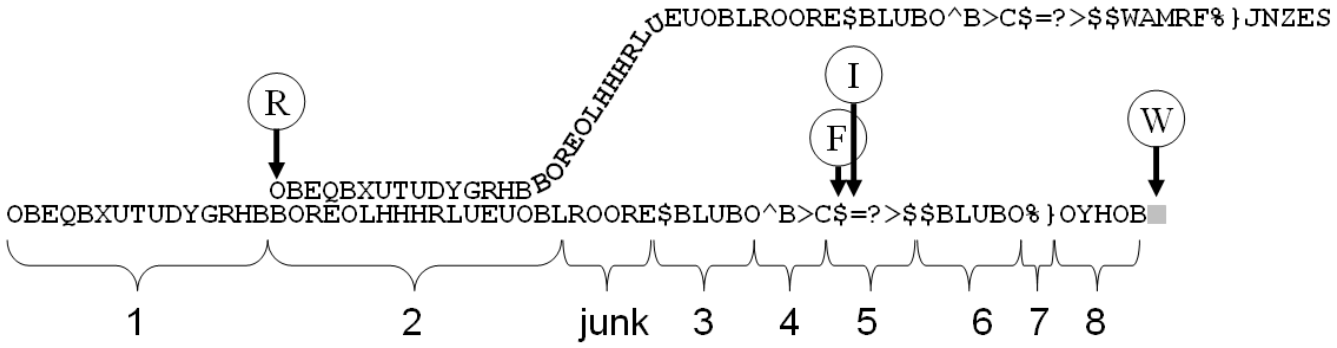
- Bind probability is a function of alignment length and accuracy
- Bind location determines:
  - which molecule is “active”
  - where the reaction-program commences (positions pointers)



Pointers

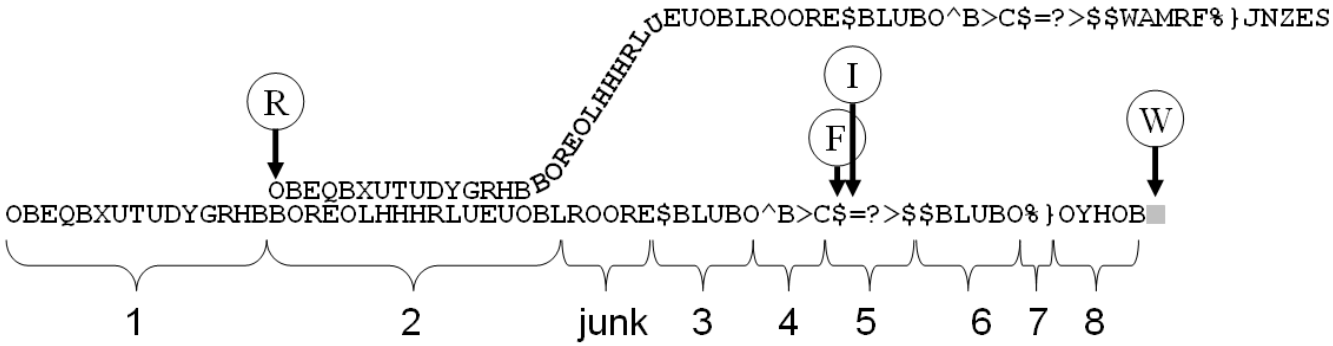
Pointers run the “code” of a reaction:
Four pointer types:

- I: Instruction points at the next code to be executed; increments
- F: Flow moves around via ’**$**’ operator; other pointers can then follow
- R: Read as part of the ‘**=**’ operator the code source
- W: Write as part of the ‘**=**’ operator the code sink



Mutation On Copy

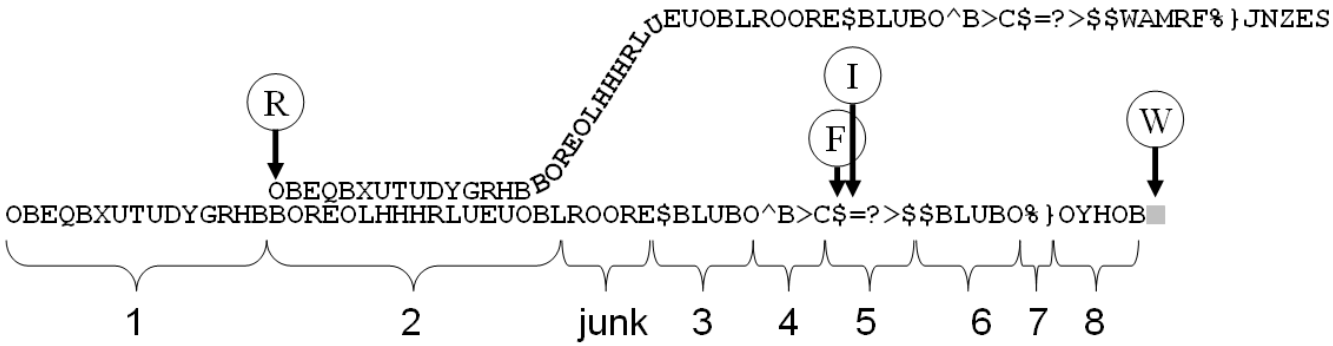
- The '=' instruction copies from the Read pointer to the Write pointer
- This has a small chance of *error* ($p = 0.0001$)
- mutation to "next door" symbols on a pre-arranged sequence
- Mutation rate *per molecule* is a function of string length
- The *basis* of all changes in the system - triggers *cascades* of mutation



Worksheet

- Let's now go to the website, and build our own molecules.

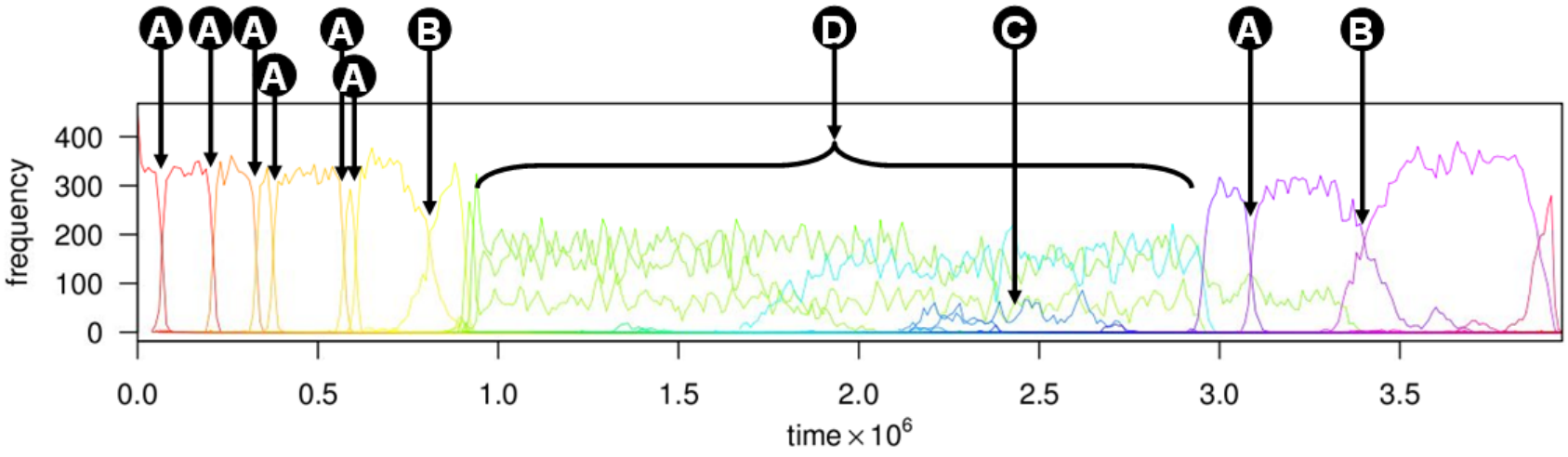
''http://www.plazzmid.org/RUTSAC11/StringmolWeb/StableVersion/index.html''



Setting up a single trial

- 400 identical “seed” replicases
- Stochastic chemical simulation
- Limited energy per time-step
- Constant decay rate for all molecules
- Survival via (inexact) copying
- Run until no stringmols remain in the system

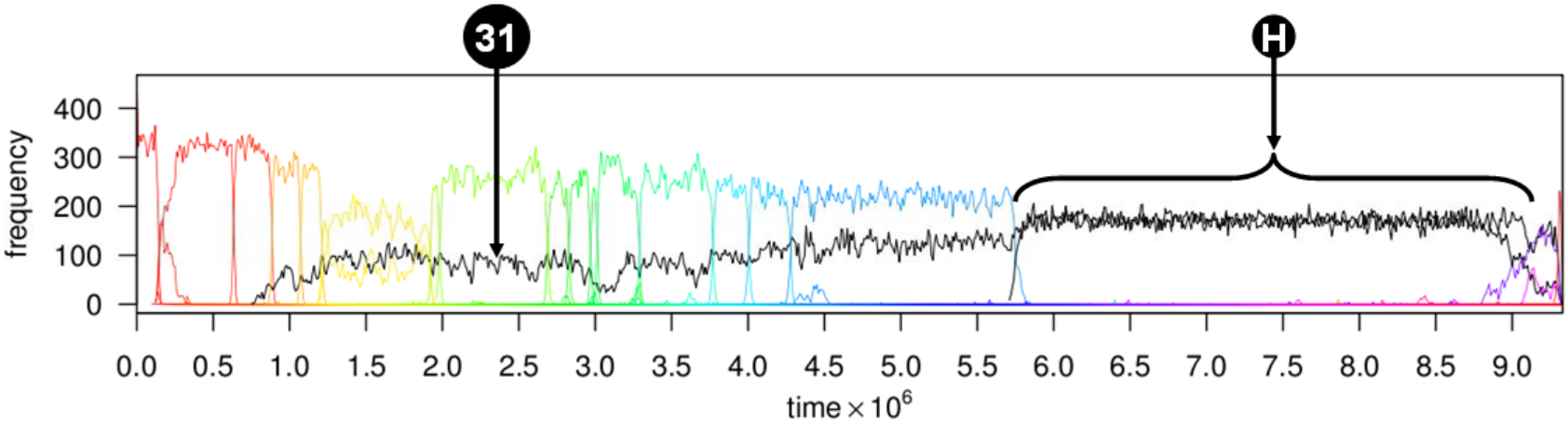
Single trial example...



Epochs of different dominant molecules are evident

- **A:** Characteristic sweep
- **B:** Slow sweep
- **C:** Subpopulations
- **D:** Multi-species hypercycles

Observations of an individual trial



- **31:** persists for $9x10^6$ time steps
- **H:** Hypercycle emerges
- Partners in this hypercycle do not self-copy

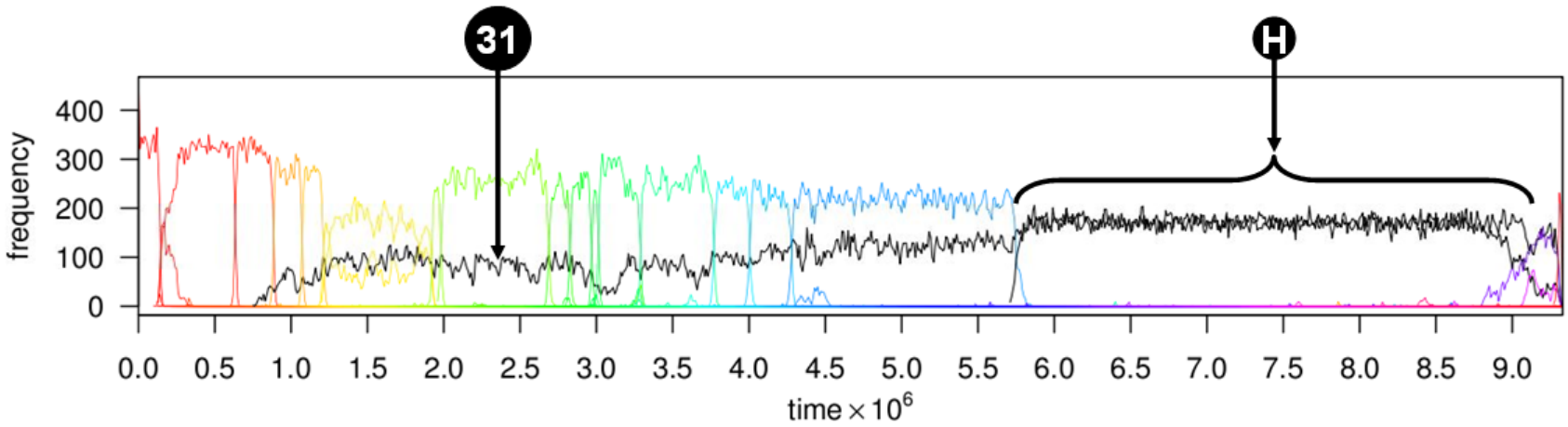
Origin of the macro mutation

- A mutation in the functional region causes a double-length molecule to be created

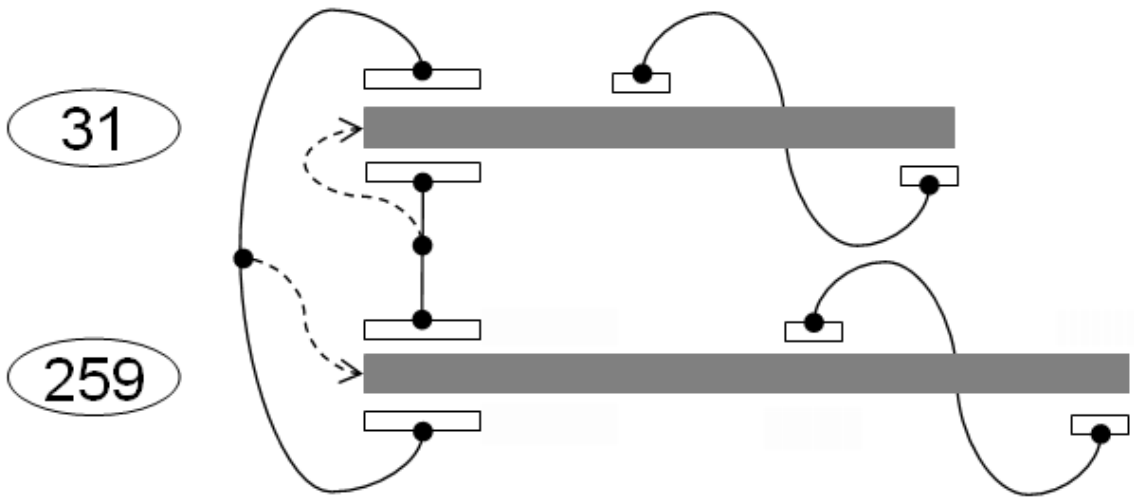
```
030 OBEQBXUUUDYG...P^B>C$=?>$$BLUBO%}OYHOOBEQBXUUUDYGRHBBOSEOLHHHRLUEUOBLROORE$BLUBO^B>C$=?>$$
Bind site:                              |--------------|
009                          OBEQBXUUUDYGRHBBOSEOLHHHRLUEUOBLROORE$BLUBO^B>C$=?>$$BLUBO%}OYHOB
Product:                                |--------------------------------------------|
031                                     BBOSEOLHHHRLUEUOBLROORE$BLUBO^B>C$=?>$$BLUBO%}OYHOB
```

- Longer alignment with centre of species 030.
- Species 009 is the template, species 030 is active
- First binding site on 009 is not copied - species 031 is created

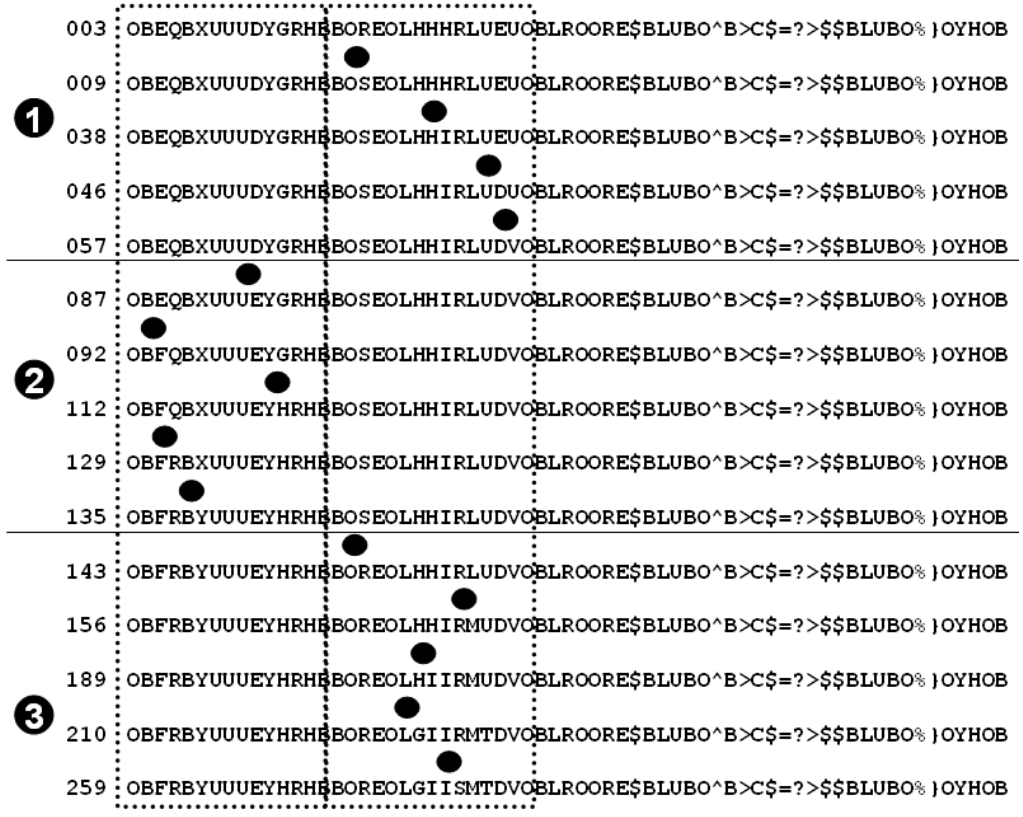
Repeated mutation $\rightarrow$ rich behaviour



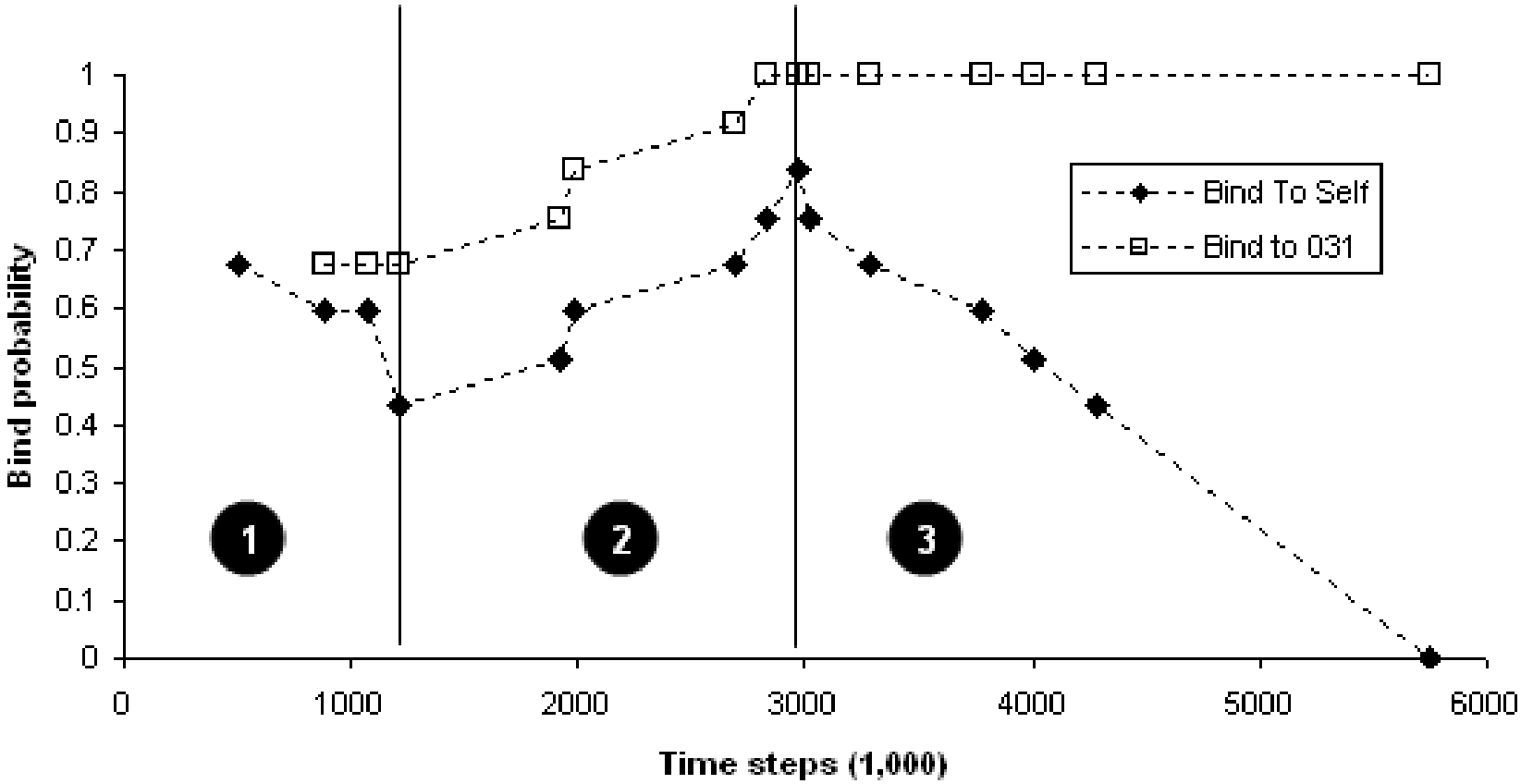
Reactions in hypercycle partners



Tracking the dominant mutations...



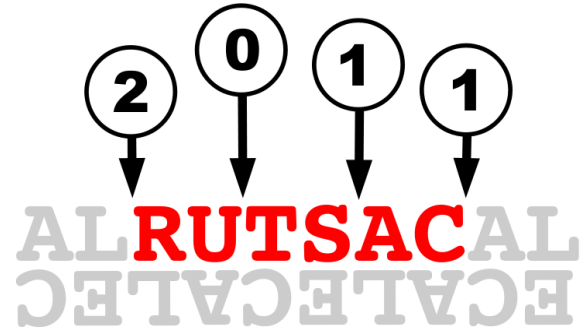
Evolution of binding



Summary

Artificial Life (ALife) is a bottom-up approach to Artificial Intelligence (AI)
Artificial Chemistry (AChem) is a bottom-up approach to building ALife

- We have devised an artificial chemistry that is encoded such that it can evolve
- A novel combination of a stochastic chemistry model with Instruction-Set based A-Life
- We have implemented a string-based molecular analogue which does
  - mutation-on-copy
  - specific binding
  - stochastic reaction chemistry



Research Using The Stringmol Artificial Chemistry

Edward Clark, Simon Hickinbotham, Adam Nellis

YCCSA, University of York, Heslington, York YO1 5DD, UK
email: sjh@cs.york.ac.uk

EPSRC grant no. EP/F031033/1