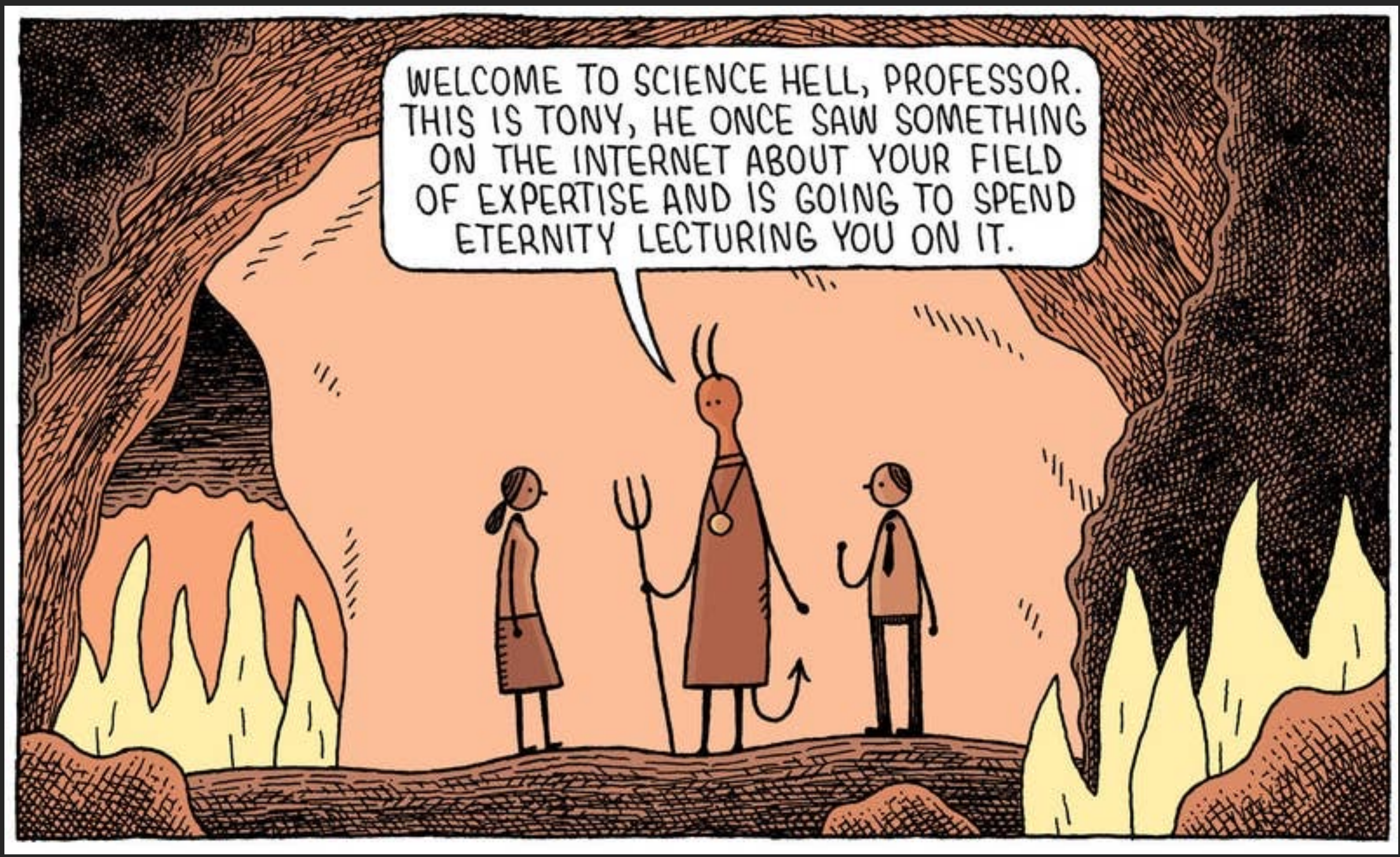
Biology of Computation

Stephanie Forrest
Biodesign Institute and School of Computing
Arizona State University
Nov. , 2024

Now ‘Tony’ will talk to you about intelligence



Naturalistic Approaches to Intelligence

The Brain

stories.uq.edu.au

The Brain

- Perceptron → Neural Networks → Deep Learning → "AI"
- Supervised training, reinforcement learning
- Mostly static network structure after learning
  - Neurons don't move

Naturalistic Approaches to Intelligence

Social insects and other collectives

argh.com

Social insects

trustalchemy.com

pxhere.com

- Individual agents have limited cognitive capabilities and local communication
- Behavioral rules pre-programmed by evolution
- Emergent collective behavior
- Move through space, Liquid brains

Naturalistic Approaches to Intelligence

Social Intelligence

inc.com

thoughtco.com

- Collective problem solving
- Cooperative learning
- Social structure
- Communication/language

Wildlifefaq.com

Naturalistic Approaches to Intelligence

The Microbiome

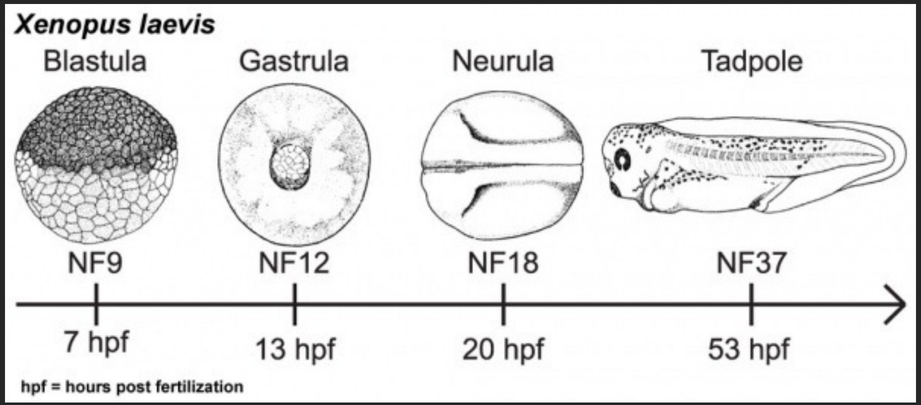
- *Digestion*
- *Gut/brain axis*
- *Behavior*
- *Arm of the immune system*
- *Endocrine regulation*

kidsandcompany.com

- Multiple interacting species (300 – 1000 different species in human colon)
- Complex regulatory logic, important for health and ecosystems
- Dynamic interaction structure

Naturalistic Approaches to Intelligence

Embryology



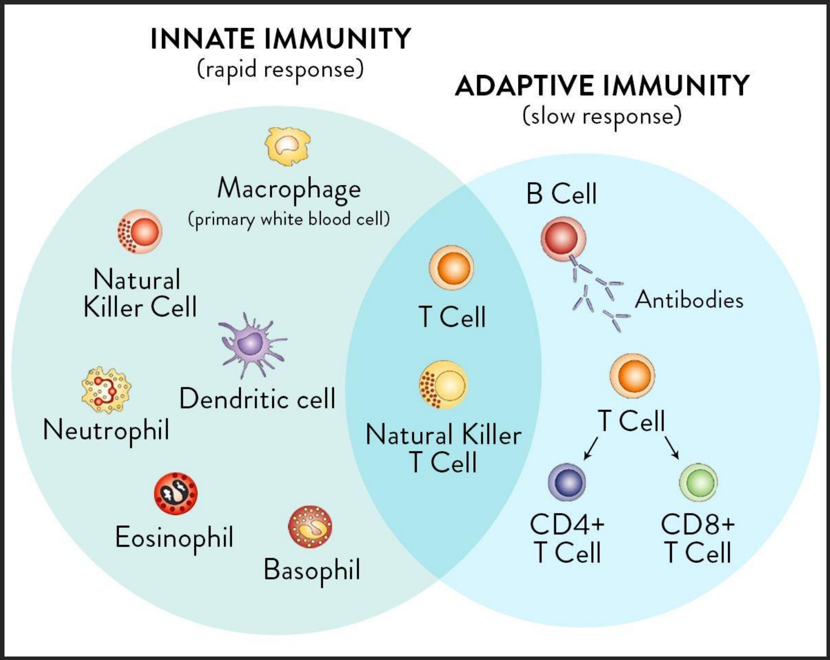
embryology.med.unsw.edu.au

wallup.net

- Nature's build process

Naturalistic Approaches to Intelligence

Immune Systems



vitalplan.com

- Endless complexity
  - What is historical accident and what is logically necessary?
  - Computational perspective can help

Information Processing in the Immune System

Immune systems learn to recognize relevant patterns

- Learned distinction between self and other
- Primary response to new foreign antigen
- Evolved biases towards common pathogens

They remember patterns see previously

- Secondary response
- Cross-reactive memory

They use combinatorics to construct pattern detectors

- $10^{11} – 10^{16}$ different foreign patterns from ~25,000 genes

They are massively parallel and distributed

Edward Jenner's first smallpox vaccine performed on James Phipps in 1796

http://www.history.com/news/vaccines-diseases-forgotten

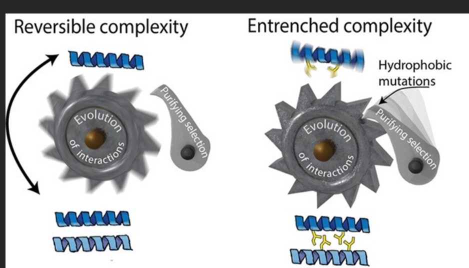
Cybersecurity Recapitulates Biology

Primary/secondary responses
- Anomaly intrusion detection, signature detection

Heterogeneous defense
- Address space randomization
- Natural diversity for N-variant systems

Second signals
- Two-factor authentication

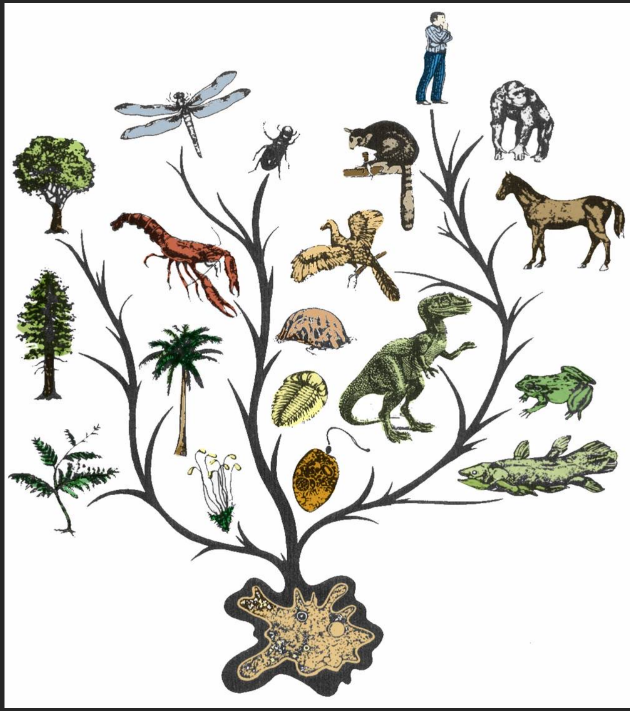
Increasing complexity
- Ratchets, constructive neutral evolution
- Defense-in-depth



Hochberg et al. *Nature*, 2020

Naturalistic Approaches to Intelligence

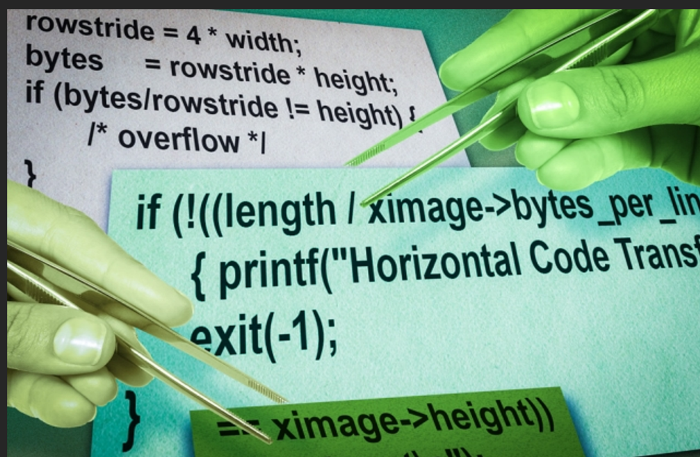
Evolution



fair-science.blogspot.com

- Nature's design process
- Robustness and diversity

Evolution in Software

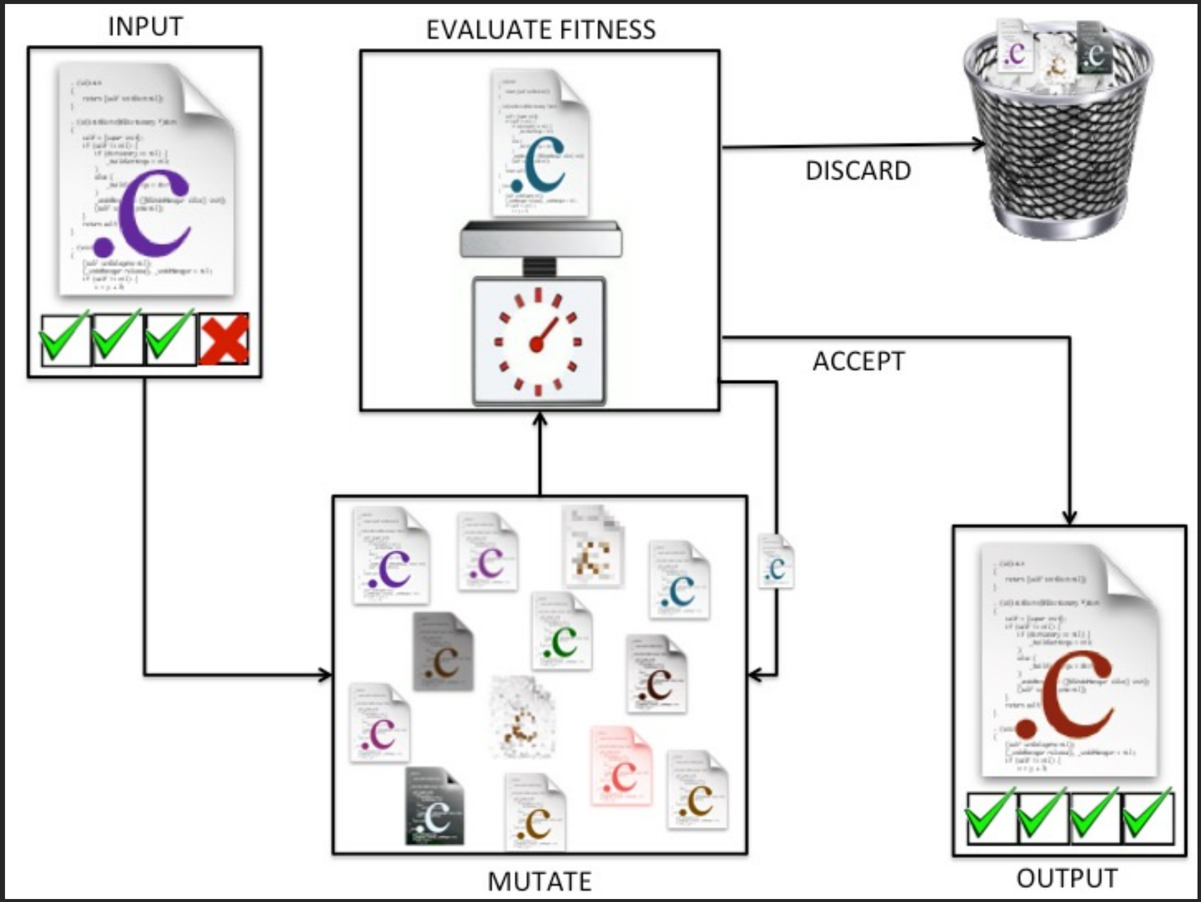


Jose-Luis Olivares

networkworld.com

- Micro-level: Evolutionary computation methods
- Macro-level: Inadvertent evolution

Micro-evolution of Software

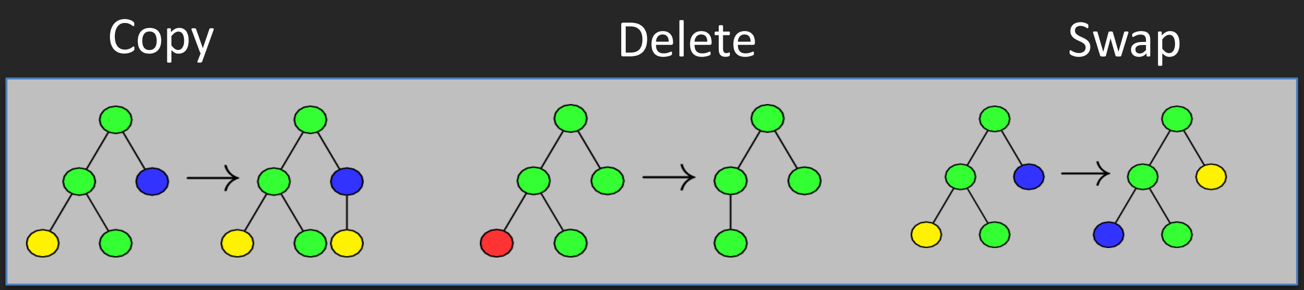


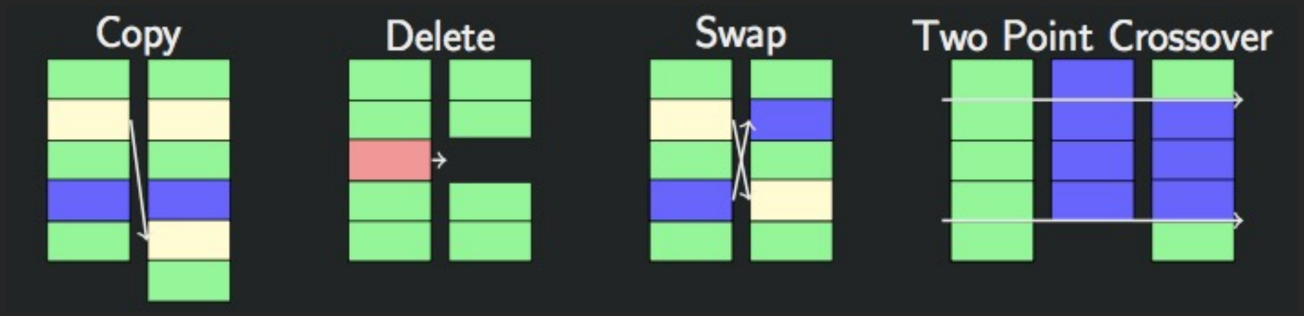
So: C. Le Goues

GenProg

ICSE '09: W. Weimer, T. Nguyen, C. Le Goues, and S. Forrest. Automatically finding patches using genetic programming. 2019: Award: Most influential paper published at the 2009 ICSE.

Mutations



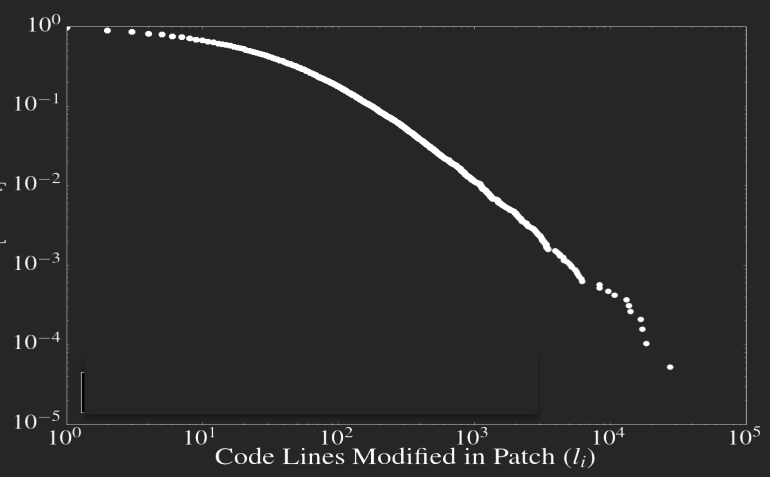


- Don't invent new code
- Statement-level operations

How can this possibly be a good idea?

- Why does GenProg succeed?
  - Algorithmic innovations
  - Exploits holes in test cases
  - Most bugs are small

frequency



size of repair

  - Software is amenable to evolution

Biological Properties of Software

- Mutational robustness
  - Mutation testing considered helpful
- Neutral landscapes
- Fitness distributions
  - Where should we look for repairs
- Epistasis

Hypothesis: Software today is the result of many generations of inadvertent evolution

Neutral Mutations

- Many biological mutations leave fitness unchanged
  - Buffering, genetic potential
- A neutral mutation passes the original test suite
  - It may or may not pass held-out failing test cases
  - Plentiful: ~30% of GenProg mutations are neutral!

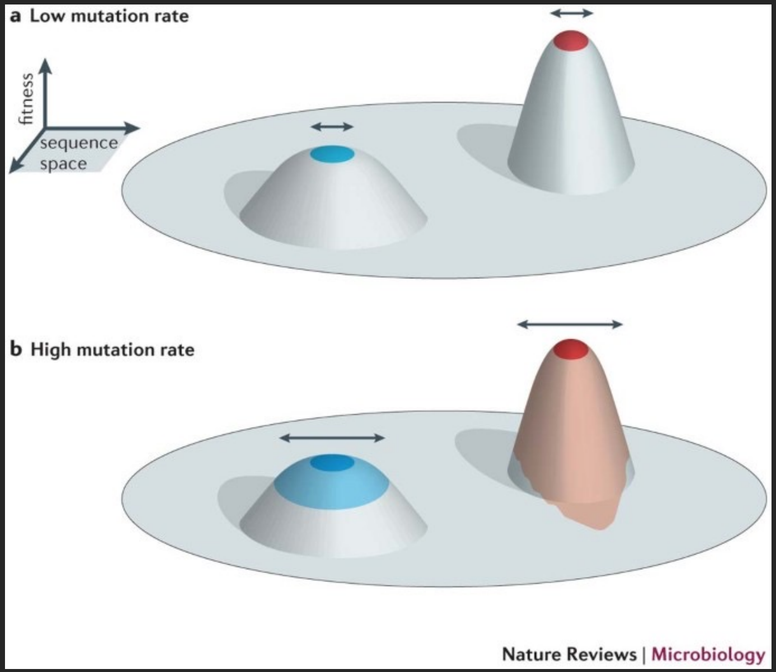
```
if (right > left) {
  // code elided
  quick(left, r)
  quick(l, right)
}
```

→

```
quick(l, right)
quick(left, r)
```

Schulte, et al. Software mutational robustness. *Genet. Program. Evolvable Mach.* **15**, 281–312 (2014).
Harrand, et al. A journey among Java neutral program variants. *Genet. Program. Evolvable Mach.* **20**, 531–580 (2019).

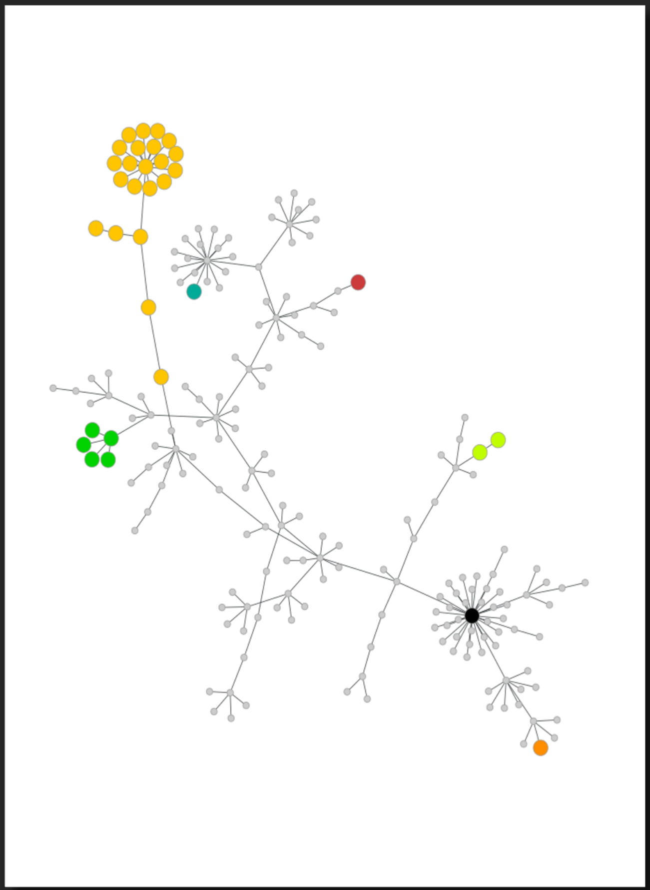
Neutral Mutations Enable Search



- Engineered diversity
- Reducing energy consumption
- For bug repairs
- For reducing GPU run-times

So: https://www.cambridge.org/core/journals/philosophy-of-science/article/abs/neutral-spaces-and-topological-explanations-in-evolutionary-biology-lessons-from-some-landscapes-and-mappings/B6F5D33A8ECB582D324CF7C8E1FC4127

Neutral Landscapes



Buffer overflow repair (look)

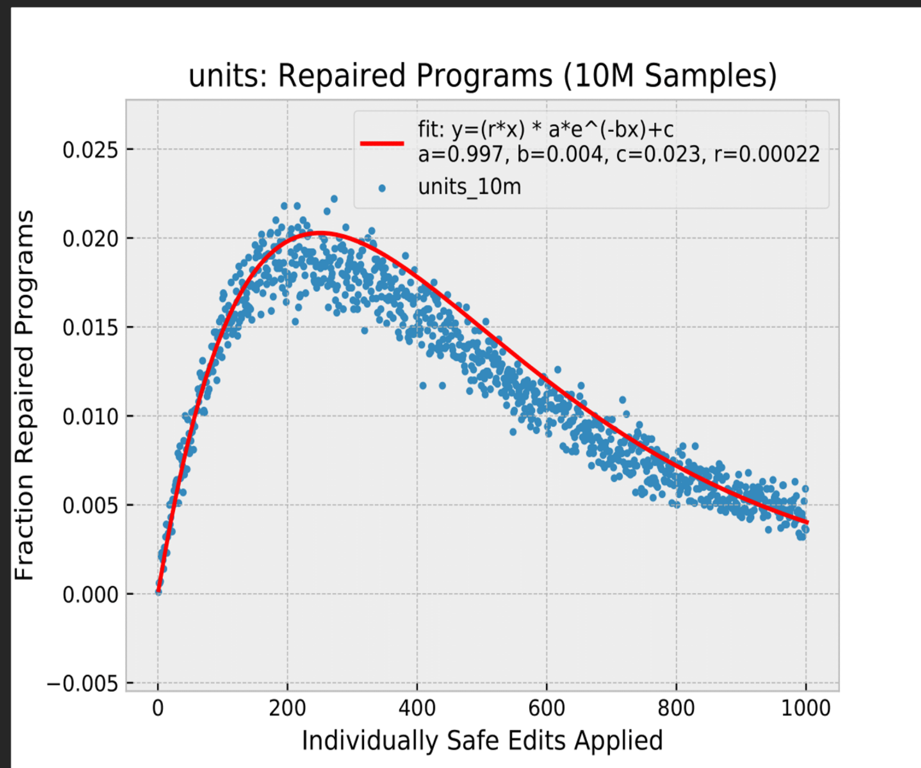
ICSE GI Workshop, 2018

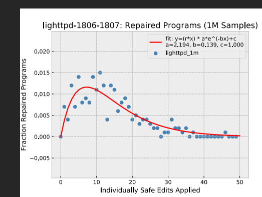
- Neutral mutations sometimes repair latent bugs
- Many semantically distinct repairs
  - Color indicates unique repairs
- Network connects diverse repairs by neutral intermediate mutations
- Insight: All repairs are neutral wrt original test suite

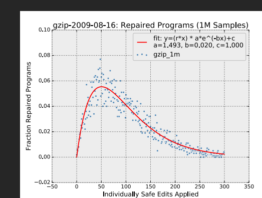
Fitness Distributions:

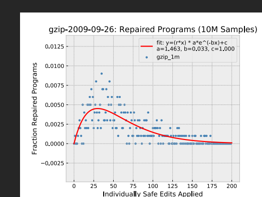
Where are the repairs in neutral space?

1. Generate large pool of neutral edits
2. Generate random subsets of pool
3. Apply each subset to original program
4. Measure repair frequency







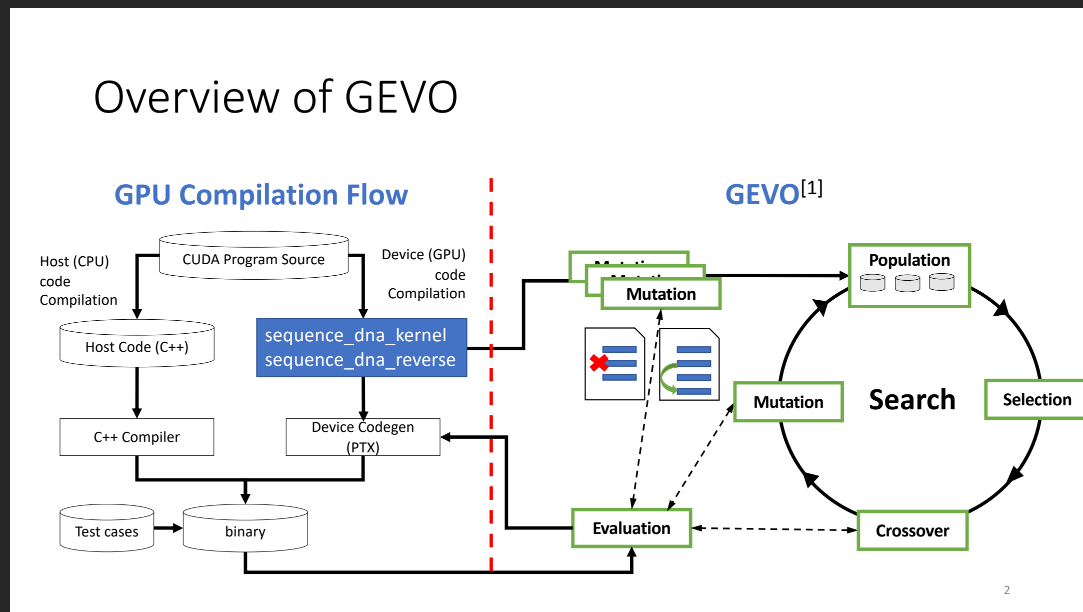


100 times more likely to find a patch at distance 200 than at distance 1

ACM TELO, 2023

Evolving Faster GPU Code

J. Liou, C. Wu and S. Forrest (TACO, 2020)

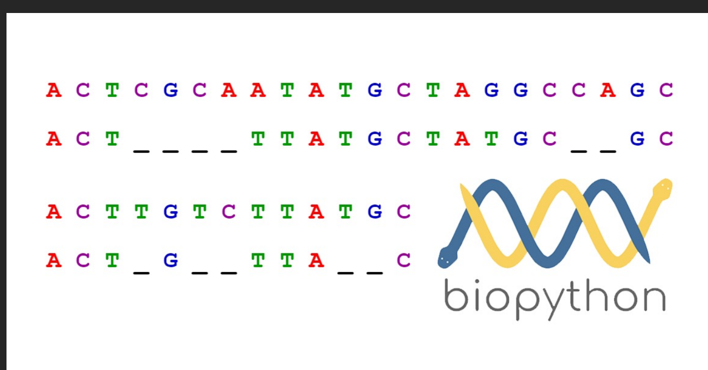


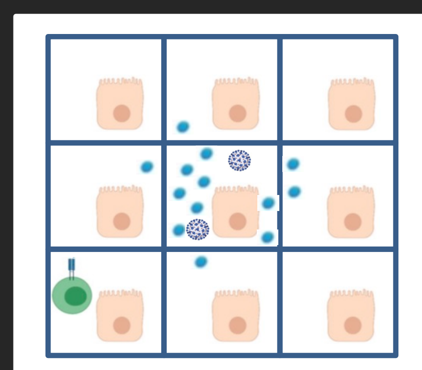
- GPUs important for ML and HPC, but challenging to optimize
- More complex mutation operators
- 49% average speedup on Rodinia benchmarks (NVIDIA Tesla P100)

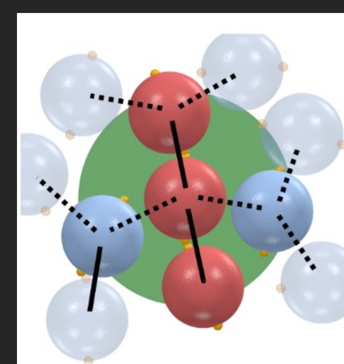
Optimizations: Application logic, architecture-specific, dataset specific

Optimizing Bioinformatics Applications
Liou et al. *TELO, in press*

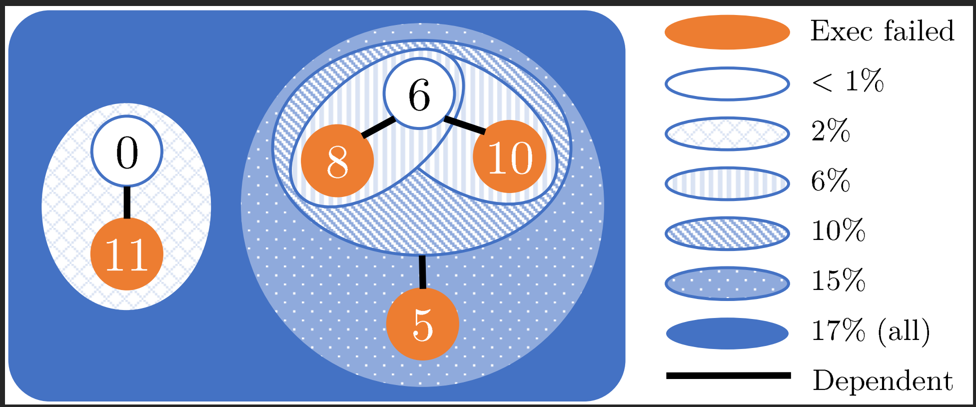
- Rodinia benchmarks are small test-oriented programs
  - What about ‘real’ programs?
- Optimized 3 GPU-enabled bioinformatics programs
  - Multiple Sequence Alignment (adept): 28.9%
  - Large-scale SARS-CoV-2 infection simulation (SIMCoV): 56%
  - Molecular dynamics (oxDNA): 17.8%

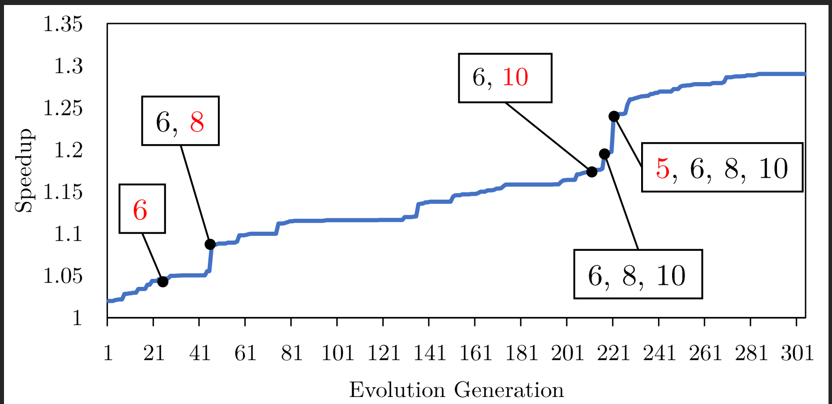






(Some) GEVO optimizations are epistatic



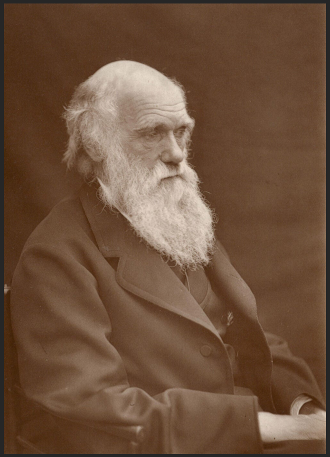


ADEPT-o on P100 GPU.

- Rearrange usage of sub-memory systems on GPU (15%)
  - Use shared memory instead of private registers
- Remove redundant synchronizations (~4%)
  - violates CUDA Programming guide
- Remove unnecessary memory initializations (30X on adept-b)

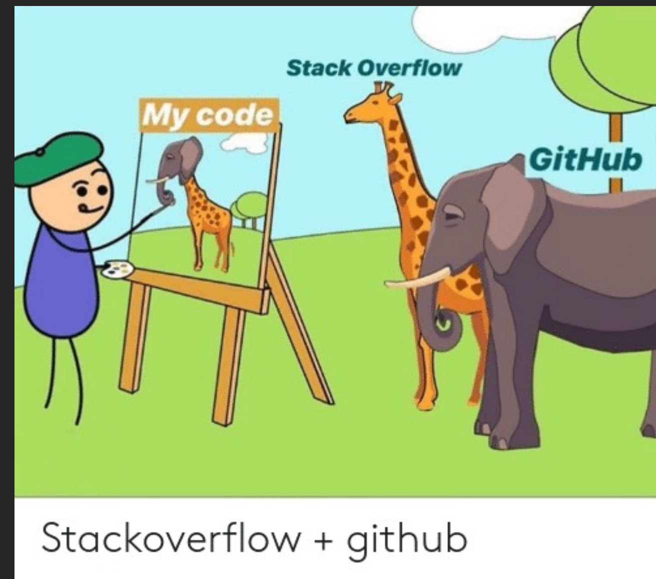
Epistatic optimizations can be hard for humans to find

The Bigger Picture

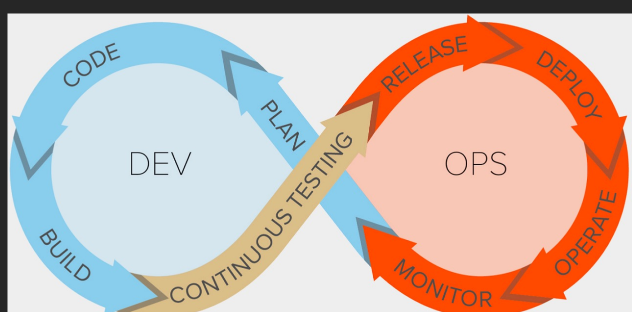


- Key ingredients of Darwinian evolution
  - Variation: Mutation and recombination
  - Natural selection
  - Inheritance
- Software
  - Selection and inheritance: Successful genes are copied: libraries, packages, code snippets, etc.
  - Variation: Programmers make small changes and recombine successful genes

Thesis: Software today is the result of many generations of inadvertent evolution stackoverflow



Macro-evolution in Software



Continuous Integration

Uber Two-factor authentication attack

Arms races

The Tinkerer and the Craftsman

Evolution

- Unplanned and openended
- Survival, relative fitness
- Ongoing process
- Incremental
- Driven by random mutation

Engineering

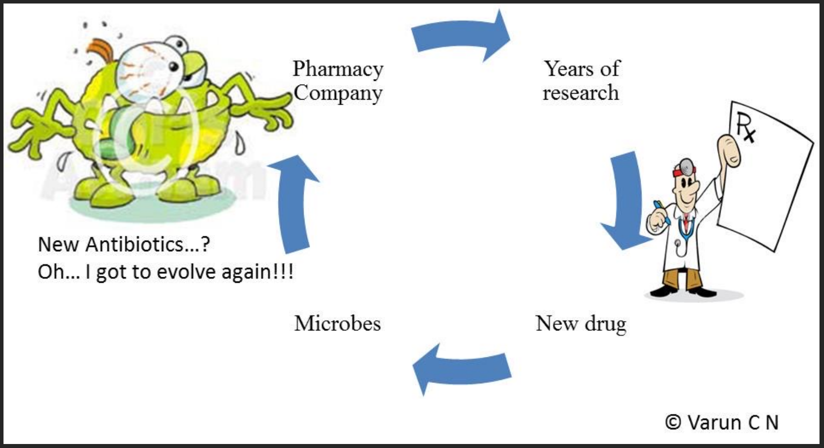
- Planned, with specifications
- Purposeful, goal-driven
- Clean slate design
- Large jumps
- Conducted by agents with foresight and intent

'Nature is a tinkerer, not an inventor'

F. Jacob

*Evolution **and** Engineering*

- **Antibiotic resistance**
- Chemotherapy
- Directed evolution
- Synthetic biology
- Attack fuzzing in cybersecurity
- Randomized algorithms
- Software

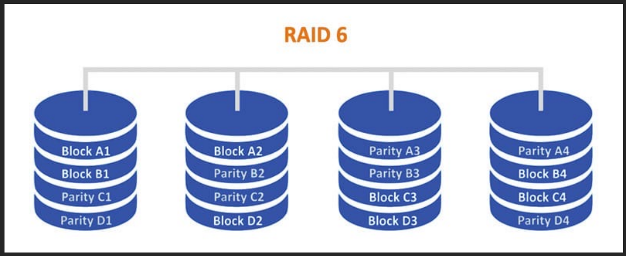


The Paradox of Robustness

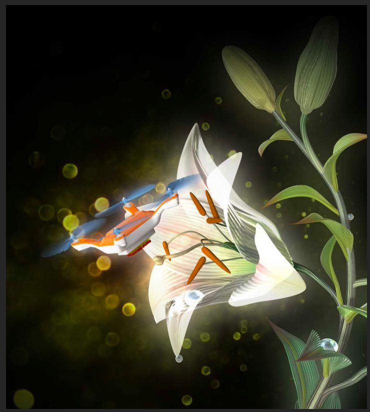
Frank, S. A. (2023). Robustness and complexity. Cell Systems, 14(12), 1015-1020.

- Evolution discovers robustness mechanisms that improve fitness
  - Regulatory controls
  - Cellular repair mechanisms, homeostasis, apoptotic mechanisms, two-factor
- But, robustness mechanisms add overhead and cost
- Reduced selective pressure on underlying components leads to degradation of components
  - Increased evolutionary drift (neutral mutations)
  - Also, potential for increased evolvability and novel discoveries
- Irreversible
- Can lead to diminishing returns, where cost of next mechanism outweighs the benefit

Does this logic apply to cybersecurity?

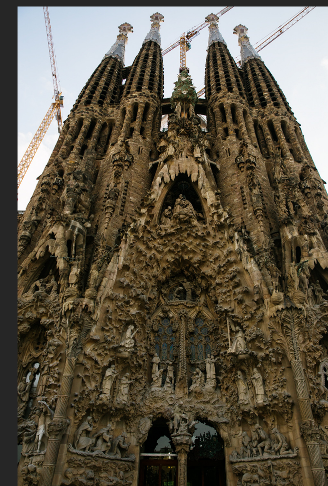


What are the best practices for engineering systems in the context of evolution?



- Claim: Software is an excellent starting point
- Co-evolution
  - Software interactions with humans
  - Software interactions among software components
  - Software interactions with biology
- Highly optimized tolerance
  - Understanding tradeoffs between performance and robustness (e.g., Carlson and Doyle)
- Rethinking defense-in-depth and technological ratchets
- Adaptive therapy for cancer
  - Manage cancer as a chronic disease and only treat enough to keep it in check

THANK YOU



steph@asu.edu

https://profsforrest.github.io

ASU Arizona State University NSF DARPA ARPA H