Visualizing Genetic Programming Ancestries

McPhee[1], Casale[2], Finzel, Helmuth, & Spector

[1] Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA

[2] Design Center Inc.
St. Paul, Minnesota, USA

21 July 2016
VizGEC

Why?

Evolutionary Computation allows us to:

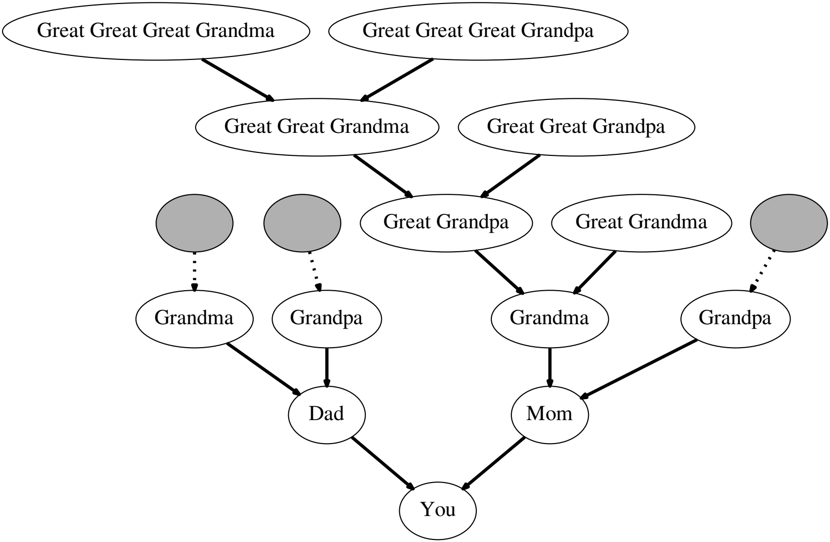
- Create algorithmic solutions by simulating biological evolution.
- Derive solutions that humans wouldn't necessarily find.
  - Like NASA's antenna! $\rightarrow$

Our Research has allowed us to:

- Turn our complex runs into organized ancestry trees.
- Modify aspects of both nodes and edges to visualize analytics.
- Find patterns in data that couldn't be seen in previous work.

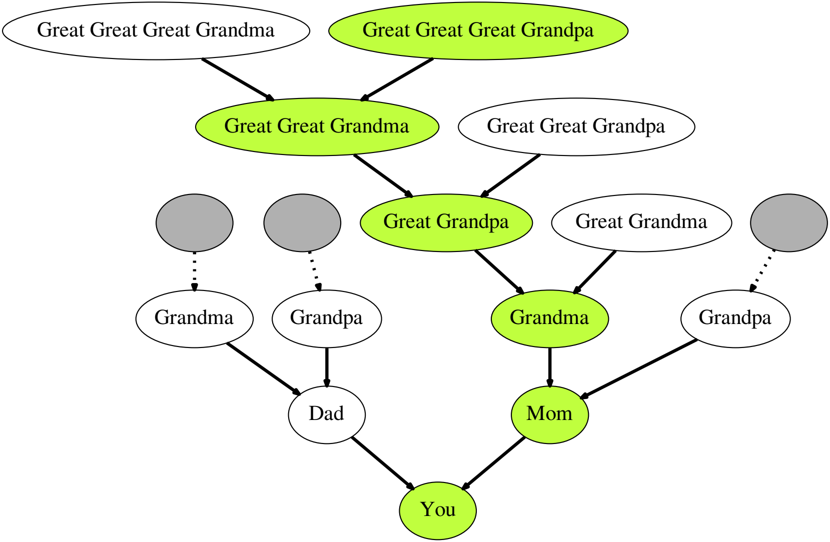
https://en.wikipedia.org/wiki/Evolved_antenna

Family Tree



Great Great Great Grandma
Great Great Great Grandpa
Great Great Grandma
Great Great Grandpa
Great Grandpa
Great Grandma
Grandma
Grandpa
Grandma
Grandpa
Dad
Mom
You

Highlighted Individuals



Outline

1. Background
2. Graph Structure
3. What did I learn?
4. Conclusions

Outline

1. Background
   - Evolutionary Computation
   - Runs
   - Ancestry Trees
2. Graph Structure
3. What did I learn?
4. Conclusions

Evolutionary Computation Versus Biological Evolution

Evolutionary Computation

- Programs
- Population growth occurs until solution

```
(defn create-snake []
  {:body (list [3 0] [2 0] [1 0] [0 0])
   :direction [1 0]
   :type :snake
   :color (Color. 15 160 70)})

(defn create-apple []
  {:location [(rand-int field-width) (rand-int field-height)]
   :color (Color. 210 50 90)
   :type :apple})

(defn point-to-screen-rect [[pt-x pt-y]]
  [(* pt-x point-size) (* pt-y point-size) point-size point-size])

(defn move [{:keys [body direction] :as snake} & grow]
  (assoc snake :body
    (cons
      (let [[head-x head-y] (first body)
            [dir-x dir-y] direction]
        [(+ head-x dir-x) (+ head-y dir-y)])
      (if grow body (butlast body)))))

(defn turn [snake direction]
  (assoc snake :direction direction))
```

https://i.ytimg.com/vi/-XzSGPJRBsw/maxresdefault.jpg

Biological Evolution

- Organisms
- Continuous growth

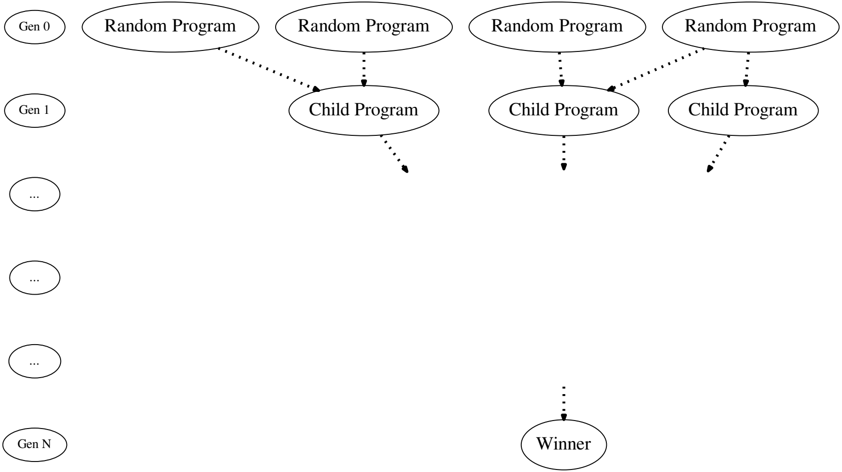
https://www.flickr.com/photos/keesey/9915333023

Run Structure

In a run, we have many generations of individuals or programs.

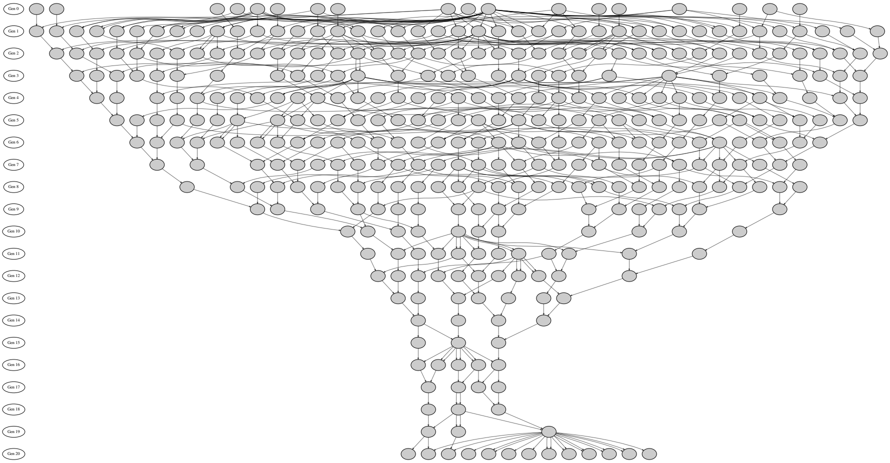
- Start with 1,000 individuals
- Test each individual on 200 test cases
- Assign error based on test results
- If a solution is found, stop
- Otherwise, make a new generation

Ancestry Trees: Basic Structure



Basic Run 0

Random Programs ↓



Winning Program(s) ↑

Outline

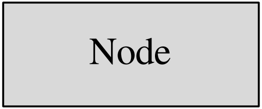
Node Basics

Nodes represent an individual or program.
They know:

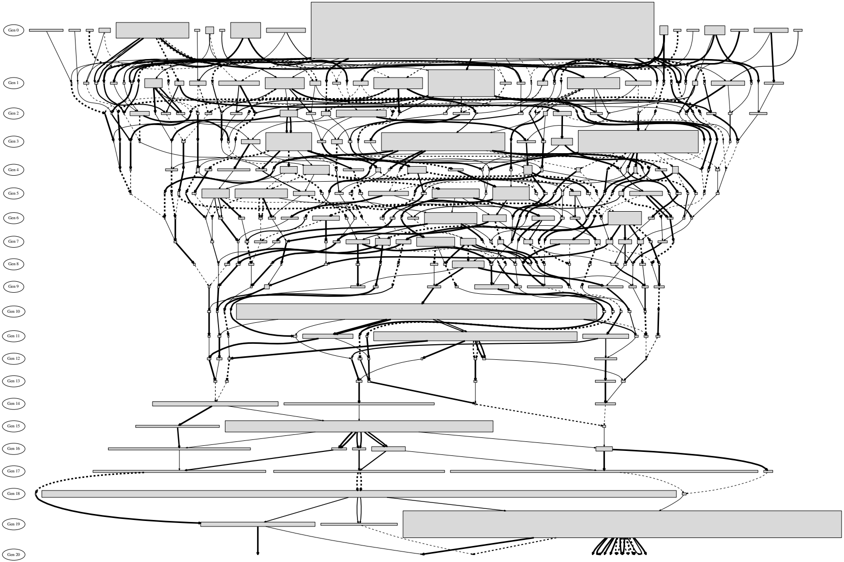
- Number of selections: Width
- Number of children: Height

← Number of Selections →

↑ Number of Children ↓

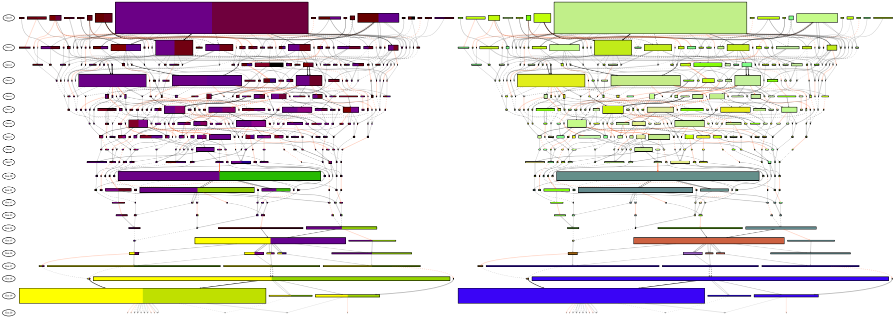


Run 0: No color

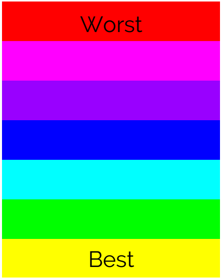


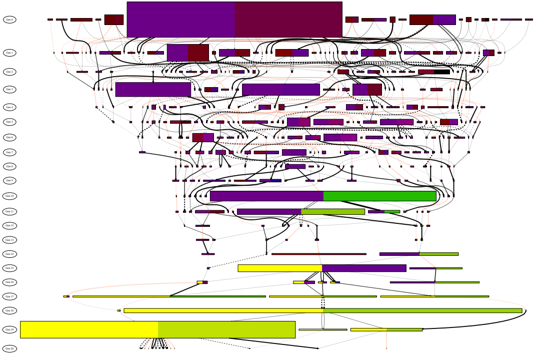
Node Coloring

There are two different techniques we use:
Error Based & RBM



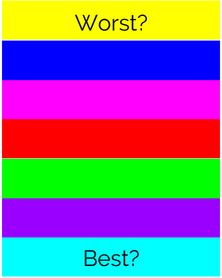
- Two part problem
- Based on an individual's total error.
- Additional shading for very high errors.

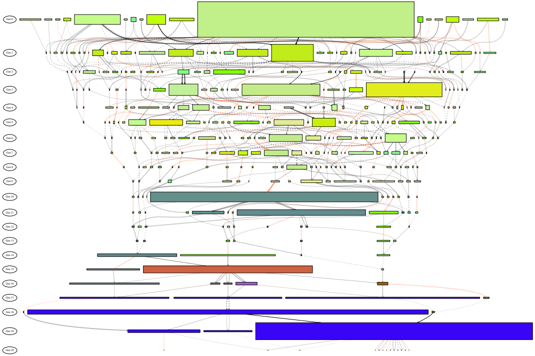




RBM Node Coloring

- RBM = Restricted Boltzmann Machines
- Dimensionality Reduction Problem
- 200 tests $\rightarrow$ 3 values





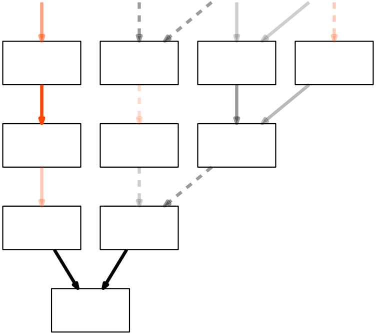
Edges

Edges represent how an individual was created.
There are two main techniques: crossover and mutation.

- Crossover
  Sections of both parents are combined together.
- Mutation
  One parent is copied and some instructions are mutated.

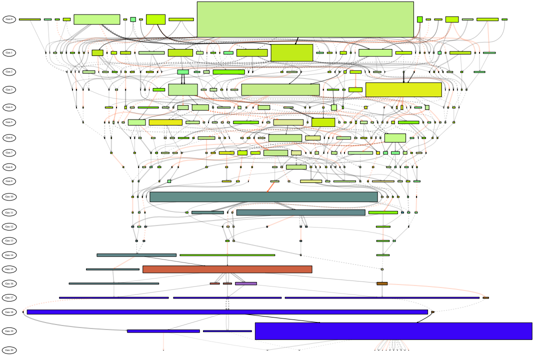
Edge Coloring & Style

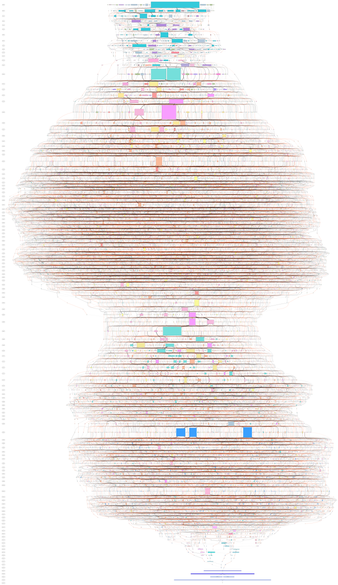
- Crossover is indicated by black edges.
- Mutation is indicated by orange edges.
- Solid versus dashed is for variations of crossover and mutation.
- Transparency is based on the number of children of the child.
    - Transparent: Few Children
    - Opaque: Many Children



Full Examples in RBM Coloring

- Run 0: Ends in 20 Generations
- Run 1: Ends in 129 Generations
  - Notice waist with large nodes

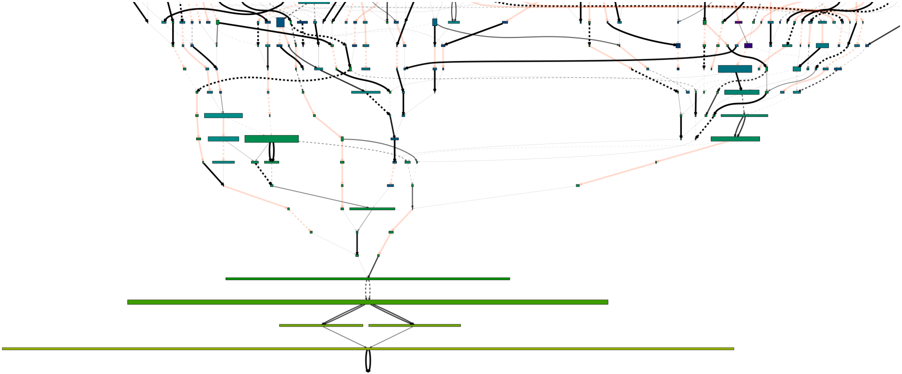




Outline

Hyperselection

One focus of this research was to examine *hyperselection*. This occurs when an individual receives a noticeably high number selections in a population.



In our graphs, we can see this as a very wide node.

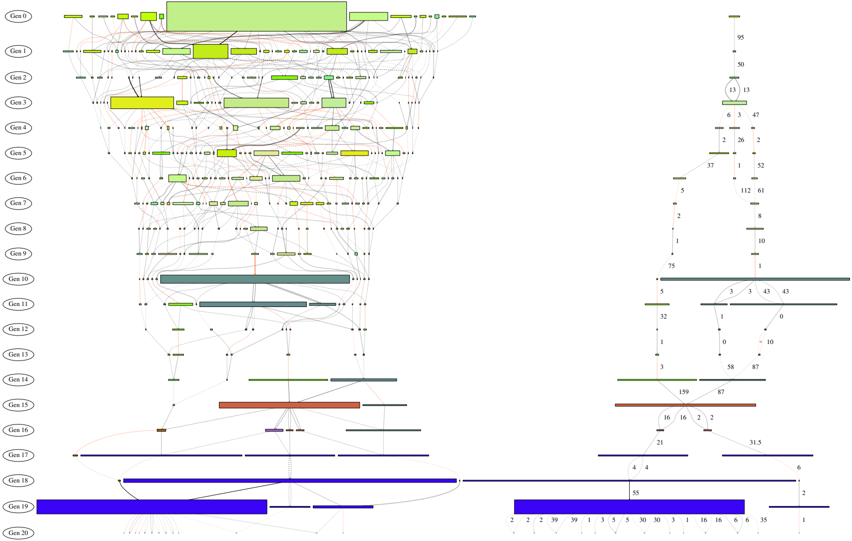
Filtering Ancestry Trees

- This is still a lot of data!
  - 22,435 Individuals
  - 1,597 Individuals
- We filter out parents who did not contribute much genetic information.
- Numbers on edges indicate similarity.
- The smaller the number is, the more similar they are.

Another Filtered Example

394 Individuals

54 Individuals



Outline

Conclusions

In Our Research We've:

- Turned our complex runs into organized ancestry trees.
- Modified aspects of both nodes and edges to visualize analytics.
- Learned about hyperselection and filtering ancestry trees.

The Next Step:

- Presenting related work in May at GPTP in Ann Arbor, Michigan
- In July we'll be attending GECCO in Denver, Colorado

Thanks!

Thank you for your time & attention!

:-)

Special Thanks to Nic McPhee!

References

B. Burlacu, M. Affenzeller, M. Kommenda, S. Winkler, and G. Kronberger.
Visualization of genetic lineages and inheritance information in genetic programming.
In Christian Blum, *et al*, editors, *GECCO '13*, pages 1351–1358, New York, NY, USA, 2013.

T. Helmuth, N. F. McPhee, and L. Spector.
The Impact of Hyperselection on Lexicase Selection.
GECCO '16, Denver, CO, USA 2016.

N. F. McPhee, M. Casale, M. Finzel, T. Helmuth, and L. Spector.
Visualizing genetic programming ancestries using graph databases.
GECCO '16, Denver, CO, USA 2016.

N. F. McPhee, M. Finzel, M. Casale, T. Helmuth, and L. Spector.
A detailed analysis of a PushGP run.
(Forthcoming) *GPTP '16*, Ann Arbor, MI, USA 2016.