Generative and Developmental Systems

Kenneth O. Stanley
kstanley@eecs.ucf.edu
School of Electrical Engineering and Computer Science
University of Central Florida
July 13, 2008

Inspiration vs. Simulation

- Often confused in GDS
  - Simulation: Model biology to learn about biology
  - Inspiration: Abstract biology to create new algorithms

- This tutorial's perspective: Looking for inspiration
  - What from biology is essential to achieve what we want?
  - What can be ignored?
  - What should we add that is biologically implausible yet works better for our purposes?

Goal: Evolve Systems of Biological Complexity

- 100 trillion connections in the human brain
- 30,000 genes in the human genome
- How is this possible?

Development

(embryo image from nobelprize.org)
Solving this Problem Could Solve Many Others

Historical Precedent

- Turing (1952) was interested in morphogenesis
  - Experimented with reaction-diffusion equations in pattern generation
- Lindenmayer (1968) investigated plant growth
  - Developed L-systems, a grammatical rewrite system that abstracts how plants develop


A Field with Many Names

- Generative and Developmental Systems (GECCO track)
- Artificial Embryogeny
- Artificial Ontogeny
- Computational Embryogeny
- Computational Embryology
- Developmental Encoding
- Indirect Encoding
- Generative Encoding
- Generative Mapping
- ...

Development is Powerful Because of Reuse

- Genetic information is reused during embryo development
- Many structures share information
- Allows enormous complexity to be encoded compactly

(James Madison University http://orgs.jmu.edu/strength/KIN_425/kin_425_muscles_calves.htm)
The Unfolding of Structure Allows Reuse

Rediscovery Unnecessary with Reuse

• Repeated substructures should only need to be represented once
• Then repeated elaborations do not require rediscovery
• Rediscovery is expensive and improbable
• (Development is powerful for search even though it is a property of the mapping)

Therefore, GDS

• Indirect encoding: Genes do not map directly to units of structure in phenotype
• Phenotype develops from embryo into mature form
• Genetic material can be reused
• Many existing developmental encoding systems

Some Major Issues in AE

• Phenotypic duplication can be brittle

• Variation on an established convention is powerful

• Reuse with variation is common in nature
Developmental Encodings

- Grammatical (Generative)
  - Utilize properties of grammars and computer languages
  - Subroutines and hierarchy
- Cell chemistry (Development)
  - Simulate low-level chemical and biological properties
  - Diffusion, reaction, growth, signaling, etc.

Grammatical Example 1

- L-systems: Good for fractal-like structures, plants, highly regular structures

\[
\begin{align*}
A &\rightarrow B[-A][+A]B \\
B &\rightarrow B
\end{align*}
\]

L-System Evolution Successes

- Greg Hornby’s Ph.D. dissertation topic ([http://ic.arc.nasa.gov/people/hornby](http://ic.arc.nasa.gov/people/hornby))
- Clear advantage over direct encodings

Growth of a Table

Grammatical Example 2

• Cellular Encoding (CE; Gruau 1993, 1996)

Cell Chemistry Encodings

Cell Chemistry Example:
Bongard’s Artificial Ontogeny

Cell Chemistry Example 2

• Federici 2004: Neural networks inside cells


Differences in GDS Implementations

- Encoding: Grammatical vs. Cell-chemistry vs. Other (coming later)
- Cell Fate: Final role determined in several ways
- Targeting: Special or relative target specification
- Canalization: Robustness to small disturbances
- Complexification: From fixed-length genomes to expanding genomes

Cell Fate

- Many different ways to determine ultimate role of cell
- Cell positioning mechanism can also differ from nature

Targeting

- How do cells become connected such as in a neural network?
- Genes may specify a specific target identity
- Or target may be specified through relative position

Heterochrony

- The order of concurrent events can vary in nature
- When different processes intersect can determine how they coordinate

Canalization

• Crucial pathways become entrenched in development
  – Stochasticity
  – Resource Allocation
  – Overproduction


Complexification through Gene Duplication

• Gene Duplication can add new genes in any indirect encoding
• Major gene duplication event as vertebrates appeared
• New HOX genes elaborated overall developmental pattern
• Initially redundant regulatory roles are partitioned

General Alignment Problem

• Variable length genomes are difficult to align

Historical Markings (NEAT) Solve the Alignment Problem

NEAT: NeuroEvolution of Augmenting Topologies

**Exploring the Space of GDS (2003)**


---

**High-Level Abstraction:**

**Compositional Pattern Producing Networks (CPPNs)**

- An artificial indirect encoding designed to abstract how embryos are encoded through DNA (Stanley 2007)


---

**Gradients Define the Body Plan**

**Embryo**

Gaussian gradient

**Adult**

Body Segments

- 1
- 2
- 3
- 4
- 5
- 6
- 7

Anterior

Posterior

---

**Gradients Can Be Composed**

- Is there a general abstraction of composing gradients that we can evolve?
CPPNs

- A connected-graph abstraction of the order of and relationship between developmental events

CPPNs: Repetition with Variation

- Seen throughout nature
- A simple combination of periodic and absolute coordinate frames
- A novel view: *not a traditional subroutine*

Is Unfolding Over Time and Local Interaction Essential to Development?

- What is lost if they are abstracted away?
- What is the role of local interaction?
  - “Where am I?”
  - If I know where I am, do I need it?
- What about adaptation/change over life?
  - CPPNs can be iterated over time
  - CPPNs can take environmental inputs
Hypercube-based NeuroEvolution of Augmenting Topologies (HyperNEAT)

- Evolving neural networks with CPPNs
- Insight: A connectivity pattern in 2-D is isomorphic to a spatial pattern in 4-D
- Result: Large-scale connectivity patterns

HyperNEAT

- Infinite-resolution connectivity patterns
- Massive working multimillion connection networks

Another Biological Abstraction: Implicit Encoding

- Idea: Let the interaction of genes in a GRN define connections in a network
  - Mattiussi and Floreano’s Analog Genetic Encoding (AGE)
  - Reisinger and Miikkulainen’s (2007) Implicit Encoding
- No explicit growth: The interaction is among genes
  - Similar products leads to stronger connections
- Advantages: Compact encoding, one-to-many interaction, allows gene duplication

Some GDS Theoretical Issues

- Expressive power of different encodings
- Chomsky hierarchy: Generative grammars of different expressive power
  - Is a CPPN comparable?
- Key consideration: Does the development process halt?
  - Yes (when phenotype complete): Then the issue is universal function approximation
  - No (continues indefinitely over lifetime): Then the issue is Turing completeness
- A CPPN can be a universal function approximator
  - An iterated CPPN may be more
- What is more important: Theoretical equivalence or bias in practice?
  - What can happen is not necessarily what will happen
How Can We Learn How Well GDS Works?

- **Benchmarks**
  - Evolution of pure symmetry
  - Evolving a specific shape
  - Evolving a specific connectivity pattern
  - Flags
  - Problems with repetition and/or variation
- **Interactive evolution**
  - Allow human to explore the space of a dev. encoding (like Dawkin’s Biomorphs, 1986)
  - Learn principles by seeing how things change, become canalized, etc..
  - Example: See [http://picbreeder.org](http://picbreeder.org) for CPPN exploration
- **Major application?**


Progress through Benchmarks

- **Visualization is most revealing**
  - Observe growth and final product

[Images of visualizations]

Julian Miller’s French Flags [http://www.elec.york.ac.uk/intsys/users/jfm7/french-flag/sld018.htm](http://www.elec.york.ac.uk/intsys/users/jfm7/french-flag/sld018.htm)

Where is GDS Useful?

- **Problems with regularities**
  - Board games
  - Visual processing/image recognition
  - Pictures
  - Music
  - Puzzles
  - Architectures/morphologies
  - Brains
  - Bodies
- **Problems requiring high complexity**
  - High-level cognition
  - Strategic thinking
  - Tactical thinking
- **Regeneration and self-repair**


Regeneration and Self-Repair

- **A major interest in much GDS research**
- **Is self-repair a side-effect of development?**

[Images of regeneration and self-repair]


- In some encodings self-repair is not needed
  - In CPPNs every cell knows its role instantaneously from its position
  - However, some applications may not provide positional information
Where is GDS not Useful?

- Problems without regularity
- Simple high-precision domains
  - Very small picture reproduction
- Simple control tasks
  - Go to the food
  - Balance the pole (5-connection solution)

GDS Testing Paradox

- GDS is a mismatch for simple problems
- Hard problems are too hard to just get started
- Where do we begin?
  - Somewhere in the middle
  - Example: Board games with ANNs

Long Term Issues

- What are the ultimate encodings?
- What are the ultimate applications?
- What application requires a structure of 100 million parts and actually utilizes the structure?

More information

- My Homepage:  
- NEAT Users Group:
  [http://groups.yahoo.com/group/neat](http://groups.yahoo.com/group/neat)
- Evolutionary Complexity Research Group:
- Email:  
  [kstanley@eecs.ucf.edu](mailto:kstanley@eecs.ucf.edu)
References from Slides


References from Slides


Additional References


Additional References


Additional References


