**GRN Models**

Trautteur supposes that universality in a biological sense is the ability to produce all producible chemicals and goes on to loosely sketch a model based on genes regulatory networks. His model consists of transcript/protein interactions (execution), slow genetic modifications (recoding), enzyme concentrations (conditionals… if, then), affinities and diffusion (addressing), gene pathway activation (procedure calls), and RNA/DNA (memory). I don’t understand the basis of the claim that this model is universal. Overall, this somewhat amateurish paper brings up a few good points, but should not be given heavy emphasis. (Trautteur 2007)

Bourchard and Osbourn offer a robust but scattered mapping between biological systems and the Turing complete RAM model of computation. They note that these systems are likely to be non-deterministic (error prone). They focus mainly on gene regulatory networks but claim and provide some examples that the same general model might be applied to cells, tissues, organs, organisms, populations, and ecosystems.

Their approach relies on registers built of unary numbers that might be represented by protein concentration, phosphorylation count, number of binding partners, polymer length, and others. Information processing on these registers might be performed through regulatory branches (like A phosphorulates both B and C), signal amplification (the equivilant of incrementing/decrementing), shape creation through recognition (two recognizing shapes outputting another shape (Conrad and Zauner, 1998; Conrad, 1999)), competition for binding partners (if protein A inhibits B: A < B then C elseif A > B then D elseif A=B then C and D (this is a fuzzy logic)). Order of operations might be maintained through the exposure of previously hidden binding sites or protein migration from one membrane to another. For instance, signal cascades proceed in a controlled sequence. Recoding might proceed from mutations. Two signal cascades could easily be spliced together by switching one protein binding domain with another.

The authors go on to provide a theoretical instantiation of their model and detail real world instantiations. They note that color changing fish move dye proteins through increment/decrement steps and congregate or separate algorithms (Sko ld et al., 2002). A more meaningful instance centers around microtubule construction which proceeds by adding a cap that is trailed by some sort of GFP operation. If the GFP operation proceeds faster than the cap, then the microtubule is eaten away and restarted from the centriole often in a different direction. This has the net effect of searching for a close anchor for the tubule.

This well-researched paper certainly deserves a place in my review, although I need to prove to myself that this is a complete and coherent mapping to the RAM model of computing. (Bouchard and Osbourn 2006)

Sharma builds off Weiss’s work to explore the human creation of logical gates in cellular systems. He points to a practical toggle switch built by Gardner using temperature sensitive florescence in e. coli (Gardner, Cantor et al. 2000). Stochastic ring oscillators in e coli built by Elowitz and Lieber (Elowitz and Leibler 2000). The author argues that the speed of genetic switches is 10^2 seconds whereas silicon is 10^-6. Notes that the Teramac computer operates well with defective chips (Heath, Kuekes et al. 1998). Overall this paper is useless biologically, but very useful in offering solutions to some of the intrinisic problems with biological computing. (Sharma 2004)

Ben-Hur and Sieglemann develop a fully fleged memory-bound turing complete computing model from a close mathematical abstraction of a genetic regulatory system based on a series of piecewise linear equations which model protein production rates (14). The slopes of these equations is determined by the patterns of genes which have crossed threshold values (artificially limited to one threshold). Memory is defined by the string of threshold values.

The authors bother to produce a simple turing machine which defines the initial state of the system w/o using more resources than subsequent computation. I will have to include this process in my analysis.

They claim their model would work with natural sigmodial gene responses because these approximate linear functions at the middle. They assume maximum interaction. They do not develop the interface used to read or write protein concentrations. They limit concentrations to one threshold. They claim their development would work in chaotic environments for reasons that I am not quite clear on. Papers on computing in chaotic environments are referenced (17-20) and probably worth looking at in addition to a second read of this paper. (Ben-Hur and Siegelmann 2004)

Busi and Zandron introduce the turing equivilatent Genetic Systems formalism. Genetic gates are regulated by the presence of activator and inhibitor proteins. Proteins are the product of gates and decay after some number of cycles or in the presence of repressors. Interactions are maximally parallel.

In proving that GeneticSystems can implement RAM register values are implemented as protein count, the successor function is implemented using the activity of a gate (which produces one protein per cycle when active), Decrement is mimicked by the production of a 1 cycle decayed repressor of the protein to be decremented, while jump is implemented by the cessation of negative regulation to a jump gate which produces the program counter protein, that has a lifespan of 1. They go through some detailed mathematical development of the system (sections 2 and 3) which deserver a more careful read and will be useful when I develop a formal system in RNA. For a review of GRN formalisms see (6).

This paper is great, but one small shortcoming is its inability to cope with the chaos conditions inherent in a cell. (Busi and Zandron 2008)

Following in the tradition of Staurt Kaufman, Istrail's paper describes transcript regulation as a network logic gates (cis-mediated regulatory modules or CRMs) that form a massive, parallel web of turing complete computation. Transcription factors act as CRM inputs while packaging models state and outputs are fuzzy logic expression rates. These factors combine to form AND/OR/NOT gates or more complicated combinations of these operators that affect the production of future transcription factors (next round of inputs). Because AND/OR/NOT gates can be viewed as universal, this system has an ulimited potential for calculation (verify this, please).

Because any given human cell has about 10^4 opperational CRMs at any given time and there are about 10^12 human cells in a body it might be said that there are 10^16 simultanous operations or compound operations with an average of 4-8 transcription factors occuring in a human body, though the authors do not clarify the rates of these calculations.

As noted in the summary of Lila Kari's paper, these biological operations are unique from silicon ones in their asynchrony, flexiability, slowness, parrelleness, memory storage, avoidance of the Neumann Bottleneck, error rates. Whereas wire communications *in silico* are one way static channels, *in vivo* diffusion based communication is slow (100 m/s^2), probabalistic, dependent on concentration, temperature, shape and mass. Binding is a probabilistic brute force phenomenon where register transformations happen by the rules of systematic and optimized algorithms.

Memory in these biological systems is stored in both active and passive states characterized by feedback loops and dna or protein modifications respectively (Davidson, 2006). Hardware and software seem are mixed in the world of biology, though certain structures like the genome and transcription machinery are far more fixed and predictiable than others.

This paper is a great example of computation through gene regulatory networks. They do an admirable job of estimating the bodies computational capacity and comparing this *in vivo* computation model to an *in silico* model. I am not entirely convinced of this paper’s Turing argument and it deserves another read as it was the first GRN paper that I had read. (Istrail, De-Leon et al. 2007)

* This paper has 51 citations (pretty good)
* It aims to implement digital circuits *in vivo* using GRN’s
* At least as early as 1974, Roessler and others [18, 19, 21, 20] noted the possibility of building universal automata by coupling bistable chemical reactions, and that chemical reaction kinetics share a formal relationship with electronic circuit action.
* Recently, McAdams and others [12, 10, 11] have constructed mathematical models of various genetic regulatory networks in vivo
* Monod and Jacob [13], Sugita [22], Kauman [9], and Thomas [23] have all made various and partially successful attempts at describing the global qualitative dynamics of genetic regulatory systems, by simplifying those systems to binary signal levels and pursuing a treatment in terms of boolean networks.
* Inverters are built by fusing a repressor protein to a promoter site that contains a different repressor sequence. The authors assume that the relationship between the input and output protein will be an simple linear inverse relationship, but this seems like a naïve assumption. I would propose that we could get around the complexities of RNA processing by trying to implement their model in bacteria, but even then the dynamics are going to be complex.
* From these inverters any other protein can be created
* This paper assumes that degredation, but not synthesis, is a linear function

(Weiss, Homsy et al. 2002)

**Immunocomputing**

This book popularizes the term immunocomputing (actually introduced in 2002) which distinguishes itself from artificial immune systems in that it pursues a hardware implementation of immune-principals, although it is still looking to use traditional circuits. It claims that Jerne’s work in the 1970’s is the origin point of a view of the immunesystem as an information processing network. (Tarakanov, Skormin et al. 2003)

Hug and Schuler in 2001 Immunocomputing foundational work

* The authors are trying to recreate the work of adelmann in DNA computing in the immune system.
* They note unique advantages of antibodies, their specificity, use of 20 aa instead of four nts, multiple antibodies will bind to the same epitope
* In addition to base pairing, computing with nucleic acids can exploit splicing mechanisms, cutting and religation of DNA (Winfree, 1996; Rothemund, 1996; Laun and Reddy, 1999)
* DNA hairpin formation (Sakamoto et al., 2000) or RNA editing (Landweber,1999).
* Notes that multiple binding partners to a single epitope makes logical gates easy to implement
* They go through a procedure for detecting the number of a particular element in a multiset implemented using proteins and biological procedures
* They go on to solving the NP-complete satisfiability problem
* All of this work relies on the idea of competitive and overlapping binding between antibodies
* I don’t think the authors have thouroughly considered the effects of randomness or the scalability of this idea
* Basically this is just a natural extension of the Adelman work into proteins, worth noting, but nothing profound

Cohen presents a plea to biologists to collaborate with computer scientists in developing their model of the immune systems. He does not bring up the mechanisms or capacity of calculation directly, but he addresses some meaningful points. He points out that the outputs of the immunesystem are many: defense, repair (cuts, bruises, and broken bones), cell proliferation activation, differentiation and regeneration, angiogenesis, killing of abnormal cells, disposal of debris. He argues homunculus theory - that the immune system is responsible for detecting the unknown but also different categories of self. For example the immune system is responsive to stress proteins which he supposes informs the immune system of the state of stressed tissues. Moreover theT-cells are known to respond to other T-cells meaning the immune system has a recursive feedback loop. Last tumors develop by gaming the computational methods of the immune system and so cancer studies may benefit from this approach. In all this paper provides a nice broad overview of the potential for computation in the immune system without speaking to the mechanistic concerns that would inform a Universiality argument. It should be used, but supplemented. (Cohen 2007).

Cohen goes into more details in his 2006 paper. He notes that the interactions between several dozen varieties of immune cells are dictated by the circulatory system and a series of weigh stations, lymph nodes, immune organs, and ad hoc assemblages (like tissue repair sites). Moreover all immune cells receive output of other immune cells and some cells specialize in this function. (Cohen 2006)

Balan and Jurgenson aim to develop a turing complete model of immunocomputing that eliminates most of the external agents that they employed in Balan’s 2002 paper. They abstract the interactions between antibodies as strings and therefore the displacement of one antibody another is an ordered (and stochastic) string replacement system. This paper was really math intense so while useful, I gave up on it and should come back to it later. (Sakthi Balan and Jürgensen 2008)

**Whole Cell Models**

Ji Explores the cell as a DNA computer using the ‘bhopalator’ model of a cell which he claims is Turing complete in his 1992 paper. This work ranges from insane to ingenious. Ji is essentially arguing for a computational cell which stores information as biopolymerers and chemical gradients receives inputs through traditional cell-sensing equipment and communicates with other cells through some (undefined?) cellular language which is isomorphic to human language. Along the way he defines new linguistic conventions including “Cellese” for the language of cells and “Bhopalator” for the cell as a computer (due to its inception in the Indian city of Bhopal). This paper might be mentioned but should not be explored further. (Ji 1999)

**Implemented Algorithms**

Adelmann initiated the field of DNA computing with a brilliant little solution to the Hamaltonian path problem which seeks to go from one point to another in a graph w/o retracing a line. While Adelmann relies on a human investigator to implement the algorithm, the process clearly exemplifies the process of calculating using DNA and would serve as an excellent example.

The algorithm he proposes simple. First he generates random DNA 20mers that represent each node and a set of overlapping 20mers that represent the directed graph that connects them. Mixing these elements together he applies a ligation reaction which fuses the combinations resulting in every possible path through the graph. PCR is used to amplify the sequences beginning and ending at the proper nodes, which ensures and accurate start and stop point. Then gel electrophroesis was used to dissect the band which contains sequences of an appropriate length to have made. Last rounds of gel bead affinity assays are used to isolate those sequences which contain each of the nodes. The resulting isolated sequences represent the appropriate Hamoltonian paths.

Adelmann goes on to discuss the scalability of this method of calculation and as history proves is too optimistic in his assessment. Difficulty in generating 20mers that won’t interact with themselves or eachother beyond the assumptions of canonical binding makes this method inefficient compared to *in silico* techniques. Still, he shows that the energy needed to run such a reaction is very low and moreover that computing with DNA molecules is possible indeed (Adleman 1994).

Various algorithms have been proposed to solve NP complete problems and other fundamental operations using DNA strands in the tradition of Adelman. Oyang (princeton) finds the Maximal Clique subgraph which has the most interconnected verticies. Guarnari and Bacroft do addition in 1996. Sakakibara (Japan) finds DNF boolean formulas in 2004 using weighted algorithms to overcome variability. Wu and Seamen (NYU) show multiplicaiton in 2006. Braich and Adelman (USC) solves a 20 variable 3-SAT or satisfiability problem which asks if T/F values can be given to variable in a boolean expression so that the expression evaluates TRUE. They do this calculation in 2002 test tubes, the largest to date. (see http://localhost/Archives/Natural\_Computing\_Review.txt)

**Ciliates**

The micronucli of exmated ciliates is converted to a polytene macronucli by cutting, splicing, macro-nuclear-destined segments (MDS) rearangement and internal eliminated section (IES) excision from 25,000 genes. This paper offers an alternative hypothesis to the templating hypothesis involving three genetic operations, loop excision, hairpin loop excision and double loop exchange. Together these operations are thought to be turing complete. (Prescott and Rozenberg 2002)

Prescott, Ehrenfeucht, and Rozenberg present an alternative hypothesis to the pointer based transition between micronuclei and macronucli. Because MDS overlapping pointers are only 3-20 bp long, they contain insufficient information to define the new macronucleus and evidence suggests that the vestigal macronucleus affects the production of the next macronucleus (Douharcourt et al 1998), the authors suggest that the micronuclus uses the vestigal macronucleus as a template.

I wonder if the transcriptome or virome is in some ways template-guided? Is there physical evidence for this template guiding? is a mechanism necessary to prove the information processing capacity of such cells. (Prescott, Ehrenfeucht et al. 2003)

Fienberg’s molecular computing paper from 1959

* Claims biology and chemistry would be well served if physics could produce an electron microscope 100 fold more powerful
* Preciently says that computers could get down to wires 10-100 atoms in diameter.
* Says that biology is already doing exactly this
* Notes that in making small machines we are limited by the precision possible with atoms (we can’t make a straight line with balls)
* Stresses like weight would be of little import at a small scale
* Metal’s grain structure might be annoying, Feynmannn suggests using plastic and glass as their amorphous qualities would make them more homogenous
* Notes that electrical and magnetic induction will operate differently and the nature of ‘wires’ and magenetic disks will have to be redefined
* Lubrication is not such a big deal as heat dissipates quickly
* Proposes making the small system in stages starting from large ones and moving to ¼ size at each iteration
* Notes that each stage must improve precision while reducing size
* I will note that biology has solved some of these problems by living in a more chaotic environment and working with more chaotic tools (a protein is not so solid as a car, but far more flexiable… lifelike)
* Van der walls forces and decreased effects of gravity will make all molecules sticky like molasses
* Preciently notes that by having atomic level controls we will be able to create a far greater variety of uniquely propertied substances
* Resistance becomes more prominent in a very small world
* All of this makes me want to have an atomic simulator program, so that I might see what it is like to have molecular servos and the like
* In the end Fynman proposes prizes to justify work on these projects

**Relavent Papers**

RNA logic gates source - http://arstechnica.com/news.ars/post/20081017-rna-based-logic-gates-compute-inside-cells.html

!CRM topology the topology of gene regulatory networks (Davidson, 2006)

Takashi and Kari prove that splicing systems are Turing complete.

Takahara and Yokomori (Japan) show that Insertion/deletion systems equal the set of all enumarable recursive languages.

Balan and Jurgenson use a protein antibody model for computation in 2006.

"RNA Guided DNA Assembly"

Angeleska, Jonoskaa, Saitoa, Landweber - Theoretical Biology 2007 - \*7 citations\*

http://localhost/Archives/Angeleska\_RNA\_Templating.pdf

Naturally-occuring switches

Mandal and Ronald R. Breaker. "Riboswitches Control Fundamental Biochemical Pathways in Bacillus subtilis and Other Bacteria." Cell 113 (2003): 577-586.

Tang, J., and R. R. Breaker. "Rational Design of allosteric ribozymes." Chemistry and Biology 4 (June 1997): 453-9.

Engineered switches

Gardner, and J. J. Collins. "Construction of a genetic toggle switch in Escherichia coli." Nature 403, no. 6767 (January 20, 2000): 339-42.

Gardner, T. S., and J. J. Collins. "Inferring genetic networks ... expression profiling." Science 301, no. 5629 (July 4, 2003): 102-5.

Adleman, L. M. (1994). "Molecular computation of solutions to combinatorial problems." Science **266**(5187): 1021-1024.

Ben-Hur, A. and H. T. Siegelmann (2004). "Computation in gene networks." Chaos: An Interdisciplinary Journal of Nonlinear Science **14**: 145.

Bouchard, A. M. and G. C. Osbourn (2006). "Dynamic self-assembly in living systems as computation." Natural Computing **5**(4): 321-362.

Busi, N. and C. Zandron (2008). "Computational expressiveness of Genetic Systems." Theoretical Computer Science.

Cohen, I. R. (2006). "Immune System Computation and the Immunological Homunculus."

Cohen, I. R. (2007). "Real and artificial immune systems: computing the state of the body." Nature Reviews Immunology **7**(7): 569.

Elowitz, M. B. and S. Leibler (2000). "A synthetic oscillatory network of transcriptional regulators." Nature a-z index **403**(6767): 335-338.

Gardner, T. S., C. R. Cantor, et al. (2000). "Construction of a genetic toggle switch in Escherichia coli." Nature **403**(6767): 339-342.

Heath, J. R., P. J. Kuekes, et al. (1998). "A defect-tolerant computer architecture: Opportunities for nanotechnology." Science **280**(5370): 1716.

Istrail, S., S. B. T. De-Leon, et al. (2007). "The regulatory genome and the computer." Developmental Biology **310**(2): 187-195.

Ji, S. (1999). "The cell as the smallest DNA-based molecular computer." BioSystems **52**(1-3): 123-133.

Prescott, D. M., A. Ehrenfeucht, et al. (2003). "Template-guided recombination for IES elimination and unscrambling of genes in stichotrichous ciliates." Journal of Theoretical Biology **222**(3): 323-330.

Prescott, D. M. and G. Rozenberg (2002). "How ciliates manipulate their own DNA–A splendid example of natural computing." Natural Computing **1**(2): 165-183.

Sakthi Balan, M. and H. Jürgensen (2008). "On the universality of peptide computing." Natural Computing **7**(1): 71-94.

Sharma, V. (2004). "Is it Possible to Build Computers from Living Cells?" BioTeach Journal **2**: 53-60.

Tarakanov, A. O., V. A. Skormin, et al. (2003). Immunocomputing: Principles and Applications, Springer.

Trautteur, G. (2007). "Does the Cell Compute?" LECTURE NOTES IN COMPUTER SCIENCE **4497**: 742.

Weiss, R., G. E. Homsy, et al. (2002). Toward in vivo digital circuits, Springer.