A Random Walk Down the Genomes: A case study of DNA evolution in Valis

Yi Zhou[a,b], Salvatore Paxia[a],
Archisman Rudra[a] and Bud Mishra[a,c]
a. Departments Computer Science & Mathematics,
Courant Institute of Mathematical Sciences, NYU
b. Department of Biology, NYU
c. Watson School of Biological Sciences,
Cold Spring Harbor Laboratory

1 Introduction

DNA molecules are chemically inert and physically inflexible. However, chromosomes and its discrete constituent base pair sequences have come to be seen as the repository of the functional blueprints of all of life. The information that is held in the genome and carefully transcribed and translated to govern metabolic and regulatory pathways has become the key focus of biological studies. Thus, while all of biology is built on the substrate of chemistry and physics, one now believes that a better understanding of biology will come through information theoretic studies of genomes. As a result, the key mathematical approaches that will play increasingly important roles in the "new biology" are ideas from systems sciences: dynamical systems, control theory, game theory, information and decision theory and mathematical logic.

Just understanding the evolutionary processes that a genome is subjected to could give interesting clues to how biology has come to look the way it does. There are many processes involved and some are not very well understood: point mutation, recombination, gene conversion, replication error (e.g., polymerase slippage), DNA repair, translocation,imprinting, horizontal transfer, etc.

In order to understand these processes, one needs to be able to analyze the vast amounts of genomic data that continue to become available. The challenges, intrigues and excitements that these "codes of life" have come to symbolize, in turn, have catapulted the embryonic field of bioinformatics to the forefront. Bioinformatics currently represents a hastily assembled set of tools to "contig" the sequences, organize the sequence databases, annotate and search these databases and occasionally generate a few computationally or statistically intriguing problems for the sister fields of mathematical and computational biology, etc. Nevertheless, a fully matured field of bioinformatics is likely to go well beyond these immediate questions to create tools to reason, ponder, question, infer, suggest experiments and pose examples and counter-examples dealing with the ensemble of available biological facts.

Faced with these issues, we have begun to create a programming language and a computational environment, dubbed Valis. By its design, Valis aims at solving the immediate problems of genomics and proteomics that the biological community currently faces, but still leaves enormous room to co-evolve as the field matures.

2 Valis: DNA Talk

Valis envisions a modern biology, driven by large scale processing of heterogeneous data coming from diverse sources. This could be anything from a Genbank sequence to the result of some microarray experiment. The current interfaces which let one access these different sources vary widely, so that a biologist needs to be an expert in very different areas of computer science: databases, networking, languages etc. Furthermore, at present, the algorithms used to extract biologically significant information tend to be developed in an ad hoc manner. These current approaches lead to very little code sharing between the data analysis algorithms with the concomitant increase in code complexity.

Instead of developing each tool *ab initio*, our bioinformatic system Valis defines low level building blocks and uniform APIs which let one use these from high level scripting languages. This enables biologists to write very simple scripts to perform fairly involved bioinformatic processing in a flexible fashion. As an example we use the Valis system to investigate the consequences of various cellular events on genomic DNA sequence evolution.

How genomes evolve is a very important problem in biology. It will lead to better understanding of the mechanisms of cancer development, and more accurate analyses of phylogeny data. We approach the study of sequence evolution by looking at statistical properties of the DNA sequences. As we will describe in a later section, we can measure the long-range correlation properties of DNA sequences and use this information to hypothesize the roles of various known cellular events and to seek unknown cellular events needed to explain the structure of the genomes. For instance, from the estimation of a few of the genomic statistical parameters, one may be able to distinguish between different models of DNA evolution, operating concomitantly and independently in coding and non-coding regions. Also, we will come back to the detailed model and how these models are represented in Valis.

2.1 Current Implementation

Valis is a language independent environment to prototype bioinformatics applications. It provides a set of libraries to read the input data stored in relational databases or in standard file formats, efficient implementations of algorithms useful to genomics and numerous visualization tools.

A Valis script can be written in any supported language: JScript, VBScript, Python, PERL and SETL. While the syntax of the scripts written in these languages varies wildly, they all see the same Valis class hierarchy. For example, once a user learns that a Valis `Sequence` Object has a method called `Input` that will read the sequence from a file, the user can subsequently use this same primitive from all the different languages.

At present, the data input objects can read into Valis sequences, maps, tables, annotations, microarray data etc. Unfortunately, in the genomics community there has been a proliferation of incompatible and proprietary file formats. By providing these objects, we can use input data from disparate sources, because once the data is loaded into Valis, it is presented in an uniform way to the computational layer.

The main strength of Valis comes from the fact that it provides extensive computational facilities to process genomics data. The current Valis environment provides numerical algorithms, string processing routines, alignment tools, sequence and map assembly facilities and statistical analysis algorithms. Of course one can always prototype the algorithm in any supported scripting language, but Valis derives its power by providing efficient implementations of the basic building blocks, enabling a biologist to process the data in quasi real-time. The system can be extended without recompiling it. New native libraries can be dynamically loaded into Valis.

Although the system is designed to be used in workstations, we can run the computation intensive processes on Beowulf computing servers. The current implementation of Valis runs on one such cluster, which is planned to be further enhanced.

Once the processing is completed, it is very important to be able to quickly visualize the results. For this reason Valis provides numerous visualization tools that allow a user to quickly display sequences, maps, microarray data, tables, graphs and annotations. These widget can be customized from the scripts.

An example of a Valis script used for analyzing a human chromosome is shown below. In this example, written in JScript (see Appendix II for the same example in PERL), we load chromosome 22 into Valis from a `fasta` file. Next, we annotate the sequence with data from a “GoldenPath” mirror. Finally, we run a word (“mer”) frequency analysis algorithm to find the probability distribution of all words of length k (“k-mers”) within that chromosome. (See Figure. 6.)

The script starts by selecting the language and clearing the output window:

```
#language JSCRIPT
Valis.Clear();
```

Now we can create a SQL data access object, and connect to one of our databases:

```
sql = Valis.CreateObject("Sql");
sql.Connect("DSN = mysql; UID = someuser; PWD = somepwd");
```

This section will create a DNA sequence (DNASeq, a string of A, T, C and G) object called seq and input its data from a fasta file. Most of the complex objects in Valis have a Display method. The sequence is here displayed with a sequence viewer, when the seq.Display() method is invoked.

```
seq = Valis.CreateObject("DNASeq");
seq.Input("C:\\GoldenPath\\chr22.fasta");
seq.SelectSequence(1);
seq.Display();
```

Next we run an SQL query on the SQL interface object. The method returns a Valis table. A Valis table is a flexible object, in which each column can have a different type (string, integer, double, etc.). The table columns and types are automatically created by the ExecSQL method, and the table is filled up with the query results.

```
table = sql.ExecSQL("select name,strand,
                    cdsStart,cdsEnd from genscan
                      where chrom = 'chr22'");
table.Display();
```

As before, we can display the table with the corresponding Display method. The last step creates a scrollable Bander widget, with which we can display a sequence along the X axis and a number of bands along the Y axis. We then load the sequence in the widget (at position zero) and create some bands. Here band number b1 (resp. b2) is a boolean band, which will be true when the sequence is either A or T (resp. G or C). bl1 is a block band, which will contain the results of the SQL query and finally freq will get the word frequency of this particular sequence.

```
a = Valis.CreateObject("Annotools");
a.LoadSequence(seq,0);

b1 = a.AddBand(1,"AT");
b2 = a.AddBand(1,"GC");
m=a.AddBand(1,"Masked");
bl1 = a.AddBand(5,"GenScan");
freq = a.AddBand(4,"Freq");
```

Now we can change color and sizes of the bands, and perform the necessary computations. Here CharBand will create a band for true values when one of the characters (e.g., AT for "A or T") is found:

```
a.CharBand(b1,"AT");
a.SetColor(b1,RGB(100,0,0)); //Red

a.CharBand(b2,"GC");
a.SetColor(b2,RGB(0,100,0)); //Green

a.CharBand(m,"N");           //Either of the match fails
a.SetColor(bl1,RGB(0,200,200)); //Cyan
```

The last step will load a block band with rows from a table. Here the parameters are the table containing the data to be accessed, the destination band, and the columns containing the starting position, ending position, the strand and the description. Finally, we run an efficient word frequency analysis algorithm, that will fill the frequency band with the occurrences of the substrings of length 15 starting at each position:

```
a.LoadBlocksFromTable(table,bl1,2,3,1,0);
a.SetColor(freq,RGB(100,0,100));
a.SetSize(freq,200);
a.FindRepeats(freq,14);

a.Display();
```

3 Valis: DNA Walk

DNA sequence can be thought of as a string composed of four letters $\{A,\ T,\ G,\ C\}$. A genome, represented as a long string of letters, can be divided into different substrings denoted as coding regions and noncoding regions. A genome is likely to exhibit many interesting local structures at various scales and and thus deviate significantly from a random string generated by drawing each letter independently and with equal probabilities. These deviations may have different statistical properties in the coding regions (that participate in eventual translation into proteins) compared to that in noncoding regions. In particular, we are interested in understanding how the letters on the string may associate with the other letters on the same string and in measuring such statistical correlations.

DNA sequences are subjected to the changes caused by various cellular events. Examining the long-range correlation within the sequences in different regions and different organisms is one of the simplest ways to estimate the effect of those cellular events on DNA evolution.

The most interesting and also complex phenomena caused by different degrees of correlation are exhibited via a fractal-like structure of the genome, with various patterns occurring in a scale-invariant and self-similar manner. The level of correlation in the fractal-like structure of the genome can be measured by its Hurst exponent, H, where $0 < H < 1$, with $H = 1/2$ representing complete absence of long range correlation.

There are two special behaviors that can be discerned from an annotated genomic sequence and are directly determined by the existence of long-range correlation. One behavior will be modeled as a *Brownian motion*, in which every letter in the string is independent of each other, indicating that there is absolutely no long-range correlation within the string and that a biological machinery examining the genome unidirectionally (say, from 5'-end towards 3'-end) cannot predict the unseen part of the genome ("the future") from the part of the genome ("the past") recently examined[1]. Such a string has its Hurst exponent, $H = 1/2$. The other behavior will be modeled as a *fractional Brownian motion*, in which every letter in the string is dependent on each other. A string with a fractional Brownian motion comes from a system with absolute long-term memory–thus the future counts on all the past. Such a string has $0 < H < 1$, but $H \neq 1/2$, excluding the case of pure Brownian motion. When $H < 1/2$, it is an *anti-persistent process* with negative feedback-like mechanism. When $H > 1/2$, it is a *persistent process* with positive feedback-like mechanism. It has been reported that the long-range correlation in DNA sequences differs in coding and non-coding regions in some prokaryote-s like *E. coli* (Peng et al., 1992 [15]). The DNA sequence behaves more like Brownian motion (with significantly low long-range correlation) in the coding regions, whereas it behaves like fractional Brownian motion (with some positive feedback-like long range correlation) in the non-coding regions. Similar phenomenon has been found also in 10% of the yeast chromosome III (Stanley et al., 1994 [18]) and in myosin heavy chain

[1] We will frequently use this notion of time throughout the paper and directly relate the genome's spatial dimension to time when discussing Brownian motions. The other notions of time (e.g., evolutionary time), when they occur, will be clearly distinguished in order to avoid confusion.

Organism **Region**	*E. coli* coding	*E. coli* non-coding	*S. cerevisiae* coding	*S. cerevisiae* non-coding	*Drosophila* coding	*Human* coding	**Exp. Value** **in Brownian** **motion**
H **value**	0.5556	0.5794	0.5749	0.6399	0.6016	0.6027	$0.5709^{\dagger}$
*Significance** P	> 0.05	> 0.05	> 0.05	< 0.001	< 0.001	< 0.001	

Table 1: Estimated Hurst exponents in different organisms and different regions of genomic sequences. $*$ The significance is tested using ANOVA with the null hypothesis being Brownian motion. † The Hurst exponent for Brownian motion is estimated by equation in Appendix I. Its deviation from 1/2 is due to data limit.

gene family for various organisms (Buldyrev et al., 1993 [1]). Valis provides a rich and comprehensive set of tools for conducting a systematic analysis of such statistical behaviors for a broader data set available now.

3.1 Analyzing Known Genomes

In order to study the scale-invariant long-range correlation of the DNA sequences, we view the DNA sequences as being generated from a random walk model. We first map the whole genomic DNA sequences following purine-pyrimidine binary rule:

$$\begin{aligned} f \quad &: \quad \{A, G, C, T\}^* \to \{-1, +1\}^* \\ &: \quad A \mapsto +1, \;\; G \mapsto +1 \\ &: \quad C \mapsto -1, \;\; T \mapsto -1 \end{aligned}$$

That means, we change purines (A/G) to $+1$ and pyrimidines (C/T) to -1. This creates a "DNA walk" (see Peng et al., 1992 [15]) along the genome. The "DNA walker," while scanning the genome from 5'end to 3'end, in each step, moves either up ($+1$) or down (-1) at every base pair according to the binary rule described earlier. If there is no long-range correlation, the walk is expected to realize a Brownian motion. Otherwise, we observe a "walker" with long-term memory and thus a fractional Brownian motion. Those two processes can be characterized by different values of the Hurst exponent H. Since a high H value ($H > 1/2$) suggests the presence of stronger persistent long-range correlation, we could look for subtle effects of evolutionary process and hypothesize how they may have come about. We use many different algorithms, implemented in Valis, to estimate H—examples of such algorithms include *R/S analysis* and *Detrended Fluctuation Analysis* (DFA). An outline of the R/S analysis algorithm is given in Appendix I.

We have analyzed various genomes using Valis: bacteria, invertebrate and vertebrate. All the DNA genomic sequences and annotations were downloaded from the publicly available NCBI Genbank database(`http://www.ncbi.nlm.nih.gov/Genbank/`). Genomic sequences of a certain organisms were separated into two subsequence files: one containing a concatenation of all the coding sequences (all the regions making up the "genes," i.e., all the exons) and the other containing all the non-coding sequences (so-called "junk DNA," i.e., all the introns and the intergenic regions). The same Hurst exponent analysis with the R/S algorithm was used for all different organisms and their different sub-regions within the genomic sequences. The Hurst exponent values were tested for significant differences against the expected value in Brownian motion. Results R/S analysis are shown for the following organisms: the whole genomic sequence of bacteria (*Escherichia coli*: K12), unicellular eukaryote (*Saccharomyces cerevisiae*, yeast) and the coding region sequences of invertebrate (*Drosophila melanogaster*, fruitfly) and vertebrate (*Homo sapiens*, humans). The long-range correlations calculation was based on a sample size of at least 400,000bp. The results are summarized in Figure. 1. and in the following table:

We observe a consistent difference in H in the coding regions compared to the non-coding regions as in previous work. The H values tend to be higher in the non-coding regions than in the coding regions. Thus, the DNA walk down the coding region sequences behaves more like a Brownian motion, while it acts as a fractional Brownian motion in the non-coding regions. For example, yeast has $H \approx 0.57$

in the coding regions, versus $H \approx 0.64$ in the non-coding regions. The higher H values in non-coding regions indicate that the sequences in the non-coding regions possess stronger long-range correlation than the coding regions. In addition, the H values in different regions increase with the evolutionary position of the corresponding organism. What can cause such a genomic structure? What can be the evolutionary advantages of having correlations? Or, is it possible that these are historical "debris" that the evolution has not cared to clean up ("garbage-collect")? How can one go about generating and testing these hypotheses?

3.2 Making a Hypothesis

Based on our observations, we hypothesize that:

> "*The differences in the strengths of long-range correlation in DNA sequences are caused by the counteractions of two sets of biological events. One set includes insertion and deletion events caused by replication slippage and DNA mobile elements, which tend to increase DNA long-range correlation. The other set includes natural selection and DNA repair mechanisms, which try to eliminate the long-range correlation caused by the former events.*
>
> "*However, the coding regions are under a higher natural selection pressure and possess the transcription-coupled DNA repair mechanism that is unique to them. Thus, the stronger correlation-elimination forces in the coding regions explains the weaker long-range correlation observed there.*
>
> "*The greater flexibility offered by larger genome sizes in the higher organisms allows for the increase of long-range correlation in DNA sequences along the paths of the evolution tree.*"

Testing these hypotheses with the available genomic sequences pose several challenges. Analyzing the variation of the Hurst exponent from region to region does provide some clues to the structure of the genomic processes, nonetheless, this is a rather crude tool for examining interactions of many complex cellular processes. A better tool is provided by the power spectrum analysis algorithms for the "DNA walker" models. Consider the random walk generated by a genomic sequence after it has been binarized. Let $X_1 X_2 \cdots X_m \cdots X_N$ be sequences of ± 1 obtained by binarizing a genomic sequence. Let $S_m = \sum_{i=1}^{m} X_i$ $(1 \leq m \leq N)$ and let the discrete Fourier transform of the series $S_1 S_2 \cdots S_m \cdots S_N$ be given as

$$S_m = \sum_{k=1}^{N} A_k cos(m\omega_k + \psi_k), \quad m = 1, \ldots, N$$

where A_k's are magnitudes at various frequencies. If the sequence $S_1 S_2 \cdots S_m \cdots S_N$ behaves like a fractal process (i.e., Brownian or fractional Brownian motion), then its magnitudes at various frequencies obey an inverse power-law spectrum (see Figure. 5),

$$|A_k|^2 \propto (\omega_k / 2\pi)^{-\alpha}, \quad 1 \leq \alpha \leq 3,$$

thus implying that

$$\alpha \approx -\frac{2 \ln |A_k|}{\ln \omega_k - \ln 2\pi}, \quad \text{where } H = (\alpha - 1)/2.$$

This relationship suggests that a fractal process is generated by a set of infinite or nearly infinite events that operate at multitude of different scales. In particular, both Brownian and fractional Brownian motion can be generated by such a biased distribution of events obeying an inverse power law with respect to frequencies/scales. However, Brownian and fractional Brownian motion differ from each other in their degrees of bias at different scales.[23]

What are such cellular events operating at multiple scales with a biased distribution? Do all such events described in the biological literature suffice to explain the structure of the genomes? Are there anomalous results that indicate that the current biological knowledge in this area lacks a sense of finality?

In cells, different cellular events acting on DNA sequences have different effective radii (number of consecutive bases that are affected by such an event) and different probability of occurrence. Those events act on DNA sequences concomitantly. Their various effective radii and diversified probabilities readily suggest a power spectrum that could explain the Brownian or fractional Brownian motion like behavior observed in genomic DNA sequences (see Figure. 5). Such a hypothesis can be tested via in silico experiments and analysis of the available genomes. For both processes, we cataloged all the known cellular events and examined how the design of Valis could be enriched to facilitate in silico experiments to test our hypothesis (or other future hypotheses of similar nature.)

- **Spontaneous Point Mutation:** Spontaneous point mutation, where a single base of A, T, C or G is substituted by a different base, is the random initiation force for the generation of small repeats, deletions or even palindromic structures. Although it has only a very small effective radius (usually 1 bp) it provides the potential sites for events with much larger effective radius.

- **Replication Slippage:** (See Figure. 2) Some DNA sequences are prone to misalignment during DNA replication, resulting in insertion or deletion of short subsequences in the replicated genome. For example, if there are tandem repeats or secondary structure in the template DNA strand during DNA replication, it may cause DNA polymerase to pause, dissociate and continue strand extension after misalignment (see Viguera et al., 2001 [21]; also see Fujii et al., 1999 [7] and Bzymek et al., 2001 [2]). A run of the same nucleotide increases the rate of small frameshift significantly. Similarly, sequences with inverted patterns flanked with directed repeats (a pattern capable of forming hairpin structures during replication) are about hundred times more likely to be deleted. (For other examples, see Tran et al., 1995 [19]). Therefore the effective radius of replication slippage causing deletions and insertions on DNA sequences ranges from 1bp to around 100bp, with the probability decreasing dramatically with the increase of the radius.

- **Mobile DNA Elements:** (See Figure. 3) Mobile DNA elements mostly include insertion elements, transposons, and retrotransposons. In their nonreplicative transpositions, the mobile elements is excised out of its original position (donor DNA), leaving in its donor DNA a double-strand break and sometimes, two tandem copies of its flanking directed repeats. In their replicative transpositions, the mobile elements are replicated through transcription and reverse transcription. The original copy on the donor DNA is not removed. The newly replicated copies can insert themselves elsewhere in the genome where target sequences can be found. Thus, the transposition of mobile DNA elements can also cause deletion and duplication in genomic DNA sequences. The frequency of transposition and size of deletion or insertion varies corresponding to specific elements. But since the target sequences are widely distributed in the whole genome, the mobile elements can essentially affect the sequence changes on the whole genome range.

- **Mismatch Repair:** (See Figure. 4) Mismatch repair (MMR) mechanism, as other DNA repair mechanisms, is highly conserved from *E. coli* to human (Eckstein & Lilley, 1998 [5]). It specifically targets and corrects DNA mismatches (from single base pair mismatch, to small indels [insertions-deletions], and to larger loops formed by deletion or duplication events) with strand specificity. While bacteria have a single pathway that is well understood, the eukaryotes have evolved different subpathways that specifically target mismatches of different sizes. The α complex-dependent subpathway mainly corrects single base mismatch or frameshifts of size $+1$ or -1 bp. The β complex-dependent subpathway can correct small loops efficiently up to around 13 bp (Sia, 1997 [17]). Furthermore, there is emerging evidence for a less well-known mismatch repair system that targets for large loops up to several Kb (Clikeman et al., 2001 [4]). Additionally, MMR plays an important role during DNA recombination. It has anti-recombination effect on homologous (more divergent) recombination. MMR recognition and correction can destroy or reverse formation of recombination intermediates.

  However, MMR efficiency depends on the target context. DNA loops with palindromic sequences or other sequences that can form hairpin structures can escape MMR (Moore, 1999 [11]). These

MMR dependent events have an effective radius ranging over a wide region—from 1 bp to several Kbs. But its efficiency is largely affected by the surrounding sequences in the genome and thus can effectively modulate long-range correlation difference in coding and non-coding DNA sequences.

- **Transcription-coupled DNA repair:** (TCR) It is one of the two sub-pathways in nucleotide excision repair (NER) mechanism. It specifically corrects the DNA lesions in actively transcribed strands. It is also highly conserved both functionally and structurally from bacteria to human. (see Eckstein & Lilley, 1998 [5]). The presence of such extra surveillance force on gene-containing sequences may decrease the chance of spontaneous point mutations in those regions. Thus, TCR can largely decrease the random initiation force in gene-containing regions for the generation of small repeats, deletions or even palindrome structures, compared with non-gene regions in the same genome.
- **Natural selection:** It is well known that the sequence changes in coding regions are much less tolerable than those in the non-coding regions. Although sometimes coding region changes may reflect the selection of adaptation, in most cases it leads to higher lethality or infertility of the individual. However, since changes are more tolerable in non-coding regions, they may be replicated and propagated in the progenies. Therefore, most changes in the coding regions, although not directly prevented, are 'screened out' by natural selection, maintaining the coding region unperturbed. In contrast, similar changes in the non-coding regions are subject to much less selection pressure and are left uncorrected. Finally, the greater flexibility offered by larger genome sizes in the higher organisms allows for the increase of long-range correlation in DNA sequences along the evolution tree.

In order to test the hypothesis that the processes described earlier explain how the genome structure has evolved, we suggest that we could carry out *in silico* evolution embodying dynamics of these processes and statistically test whether the *in silico* genomes have same statistical structures as the *in vivo* genomes at every scale. Thus this suggests that Valis as a bioinformatic tool must possess not just analysis tools but also synthesis tools of comparable power.

4 Genome Grammar and in silico evolution

The genome synthesis tools are structured around a "*Genome Grammar*". This is a stochastic grammar with primitives for many kinds of mathematical probability distributions. Valis can even generate a sequence with the same probability distribution as measured from biological data. Furthermore, there are tools that let one apply some hypothesized processes on sequences obtained from the grammar. This enables biologists to test any model and conduct evolutionary experiments *in silico*.

DNA evolution simulation of the kind, needed to test our earlier hypothesis, should have essentially two qualities: the ancestral genome that we start the evolution from must have the structure similar to the natural ones and such genomes can be modified efficiently in the computer with a set of primitive operators that can be endowed with a probability distribution and effective radii. The former can be achieved with a "stochastic context free grammar" and the later an be accomplished with a series of array operators that can operate on the character array (DNA sequence) simultaneously. These operators can be context-dependent and can afford sufficient expressibility to simulate the actual cellular events. Furthermore, many parameters associated with these operators can be easily changed to simulate effect of different ambient conditions or a mutant. Use of stochastic grammars to generate genomic sequences is not new (see Myers, 1999 [12] and Searle, 1993 [16]).

4.1 Grammars to Describe Genomes

Grammars are basic tools from linguistics that also find widespread usage in computer science to formalize computer languages (usually context free languages). They are also used to describe strings generated by a discrete system with constant amount of memory (regular languages or linear grammars) or other structured objects (e.g., graph grammars). These grammars formalize how concrete objects can be constructed out of structured abstract objects (non-terminals) each one capable of being further recursively expanded to create partially-formed structures with both abstract (nonterminal) and concrete (terminal) objects. This process of successive expansion of "sentential forms" terminates with a concrete structure whose interpretation depends on how it was constructed by expansion rules (production rules) of the grammar.

These concepts can be used in the domain of biology. For instance, in the genome, certain regions of the genomes have different functional purposes and they must associate with each other in a specific manner to provide those biological functionalities. The natural selection determines how well these functionalities and their associations need to be conserved. Thus, the genome may be associated with rules such as "genes are made of exons and introns;" "the regulatory regions for the genes must have certain physical relations with the genes that they regulate;" "the telomeric regions may have certain repeat structures," etc. These rules can be translated into a "genome grammar," where the abstract notions are embodied in the notions of "genes," "promoters," "enhancers," "transposons," "satellites," etc.

Thus, a (context free) grammar consists of a (finite) set of symbols T (the terminals, here the bases), another (finite) set of symbols N (the non-terminals, the hidden variables like $\langle$ *CodingRegion* $\rangle$), a specific non-terminal symbol S (the start symbol, e.g. $\langle$ *Genome* $\rangle$) and a sequence of rules of the form

$$\text{NONTERMINAL} \quad \longrightarrow (\text{NONTERMINAL} + \text{TERMINAL})^*.$$

Such a rule is interpreted as denoting an allowed rewriting, replacing the non-terminal on the left hand side (head) of the rule with the string on the right (the body). A generalization of this process allows a rule to be applied probabilistically and leads to a more natural approach to describing biological sequences. Thus, we are led to rules of the form

$$\text{NONTERMINAL} \quad \xrightarrow{p} (\text{NONTERMINAL} + \text{TERMINAL})^*.$$

These probabilistic grammars, (also called, *stochastic grammars*), naturally lead to not just a set of sequences, but certain probability distributions over the set of all possible sequences. The so-called *hidden Markov models*, with wide-spread use in the bioinformatics community, are special cases of stochastic grammars.

A "toy" genome example using "stochastic grammar" can capture the following biological concepts: "A region immediately surrounding a gene can be thought of as consisting of a regulatory sequence, which determines when the gene will get turned on, followed by a sequence of *exons* and *introns*, followed by a sequence called the terminating sequence. Exons are the parts of the gene that actually carry the code for a protein. Introns are subsequences in this region that occur between successive exons." A grammar describing these ideas is as follows:

$$\begin{aligned} \langle Gene \rangle \quad &\longrightarrow \quad \langle RegulatorySequence \rangle \, \langle CodingRegion \rangle \, \langle TerminatingSequence \rangle \\ \langle CodingRegion \rangle \quad &\longrightarrow \quad \langle Exon \rangle \, \langle Intron \rangle \, \langle CodingRegion \rangle \mid \langle Exon \rangle \end{aligned}$$

While, at this level, the grammar is non-stochastic, probabilities enter in a significant manner as we describe the base pair structure of the exons and introns or how the lengths of the exons and introns are distributed.

4.2 Linear Automaton

We need a few other components to construct a usable system that can model the cellular evolutionary events. These events can be described via the following process, that models a biological machinery (e.g., a protein complex or an enzyme) walking along the DNA and modifying the DNA under local scrutiny:

1. The machinery looks for a certain initiation recognition site on the DNA sequence (they could be determined by the constituent bases, the physical or chemical properties determined by them, etc.) The initiation site also depends on the properties associated with the machinery itself. This process can be used, for instance, to model how a transcription factor may associate itself to a regulatory sequence.

2. After such a machinery recognizes the site, it starts to "walk" along the DNA sequence changing its own state if necessary, and changing the DNA sequence it passes over.

3. At some point (probably when it encounters the terminating site), this process halts.

Depending on the distribution of the initiation sites, and the relative distributions of the corresponding termination sites, we can associate an effective radius as well as a probability distribution to each of the events that can be mediated by these machinery.

A naive implementation of the above scheme is straightforward, but will not efficiently scale to allow the presence of many large eukaryotic genomes. Thus, simulation of recombination events in DNA evolution in large populations might involve code of the form:

```
$StrandSet \longleftarrow \{Strand1, Strand2\}$
loop
  $S1 \longleftarrow random(StrandSet)$
  $S2 \longleftarrow random(StrandSet)$
  $(Succ1, Succ2) \longleftarrow RandomRecombination(S1, S2)$
  $StrandSet \longleftarrow StrandSet \cup \{Succ1, Succ2\}$
end loop
```

Since a recombination event changes only a limited region of the chromosome, we are led to an ever-expanding pool of large strands which differ at very limited regions from one another. A naive implementation which does copy-on-write will not perform well on such complex simulation problems requiring, for instance, the ability to track recombination events over multiple chromosomes. Furthermore, in practice, we will generate our sequences from some stochastic grammar artificially, but may combine them with some natural genomes obtained from a database of sequences.

Our implementation in Valis, makes intelligent use of the "biological properties" of the sequence data and avoids inefficient memory usage, as follows: we keep the sequences as $B+$ trees of logical nodes which point to physical nodes containing actual data. We also keep track of the transformations which were applied to that particular sequence. The data-structure supports many "house-keeping operations" with the property that even after insertion or deletion events, one can still access the appropriate element from the sequence. In Valis, direct concatenation of sequences still remains expensive and is implemented via a logical concatenation operator, which "remembers" that two sequences were concatenated. All these operators are available to a higher level garbage-collector like subsystem which hides these details of the implementation.

5 Biology and in vivo evolution

Results of the in silico evolution will take us back to biology. We need to be able to recreate (in a statistical sense) the in silico evolution using a real organism. There are various mutant organisms that differ from their related "wild-types" in the way our modeled cellular events has been naturally modified in these mutant organisms. These mutants may have suffered significant changes to some of these cellular events in one (or both) of the following two ways: either the effective radius has changed or in the occurrence probability at a particular scale. Those mutants are the natural candidates for "debugging" our in silico experiments, completing a cycle of reasoning. As an example, one of our favorite candidates is *pol3-t*, a temperature-sensitive mutant in yeast DNA polymerase δ. Under non-permissive temperature, the *pol3-t* mutation increases the rate of small deletions by up to 100 fold (Tran et al., 1995 [19]; Tran et al., 1996 [20]). For another example of how these systems can be perturbed *in vivo*, consider the two mismatch recognition complexes in eukaryotes. Both of them contain MSH2 protein. α complex has MSH6, and it preferentially targets single base pair mismatch or frameshifts of size one. β complex contains MSH3, and it recognizes larger DNA loops more readily. Mutation in MSH2 leads to an increase of DNA sequence changes in a wide range (from 1 bp to 16 bp). However, the deficiency in MSH6 only specifically elevates the rate of 1–2 bp changes and causes instability of small microsatellites and minisatellites. The defect in MSH3, on the contrary, can increase the rate of larger sequence changes (> 5 bp) by tens of fold (Sia et al., 1997 [17]). Therefore, even the mutations involved in the same cellular event can lead to completely different changes in the effective radius or occurrence probability of such event.

By tracking the changes in the genomic sequences of those mutants during evolution and by comparing them with their wild-type "controls," we expect to gather all kinds of information about the genomic evolutionary processes. Other researchers have traced over 10,000 generations of bacteria population in lab (Lenski and Travisano, 1994 [10]). The *E. coli* strains undergoing uninterrupted laboratory evolution include some MMR mutants (Vulic et al., 1999[22]). Similar experiment can be set up for budding yeast in order to explore eukaryotic sequence evolution. A biologist will find Valis an invaluable tool for conducting similar large-scale experiments, making hypotheses and validating/falsifying these hypotheses through quantitative analysis.

6 Acknowledgment

We would like to thank Prof. Mike Wigler and Ms. Elizabeth Thomas of Cold Spring Harbor Lab for many exciting ideas, useful contributions and their own examples on how to think about genomes. We would also like to thank Profs. Misha Gromov and Ale Carbone for introducing us to the idea of looking at genomes in a large scale and examining them for structures related to physical properties. In particular, they introduced us to the work of Yeramian and Gorban. In addition, we have benefited from numerous ideas from our close collaborators who have indirectly influenced our thinking in computational biology, genomics and bioinformatics (CBGB): Thomas Anantharaman, Dave Schwartz, Izzy Edelman, Jack Schwartz, Eric Siggia, Chris Wiggins, Gloria Coruzzi, Dan Tranchina, Suse Broyde, Phil Benfey, Harel Weinstein, Mike Seul and Bob Desnick and many others.

The following research grants supported the work reported in this article: "High-Density Gene Copy Number Microarrays," National Institutes of Health; "Genomics via MicroArrays," NYU University Research Challenge Fund; "Bioinformatics Prototyping Language for Mapping, Sequence Assembly and Data Analysis," US Department of Energy; "Faculty Development Program for Bioinformatics and Genomics," New York State Office of Science, Technology, & Academic Research (NYSTAR); "Algorithmic Tools and Computational Frameworks for Cell Informatics," DARPA; and "Algorithmic and Mathematical Approaches in Cell Informatics," HHMI Biomedical Support Research Grant.

References

[1] S.V. Buldyrev, A.L. Goldberger, S. Havlin, C.-K. Peng, H.E. Stanley, M.H.R. Stanley and M. Simons. "Fractal Landscapes and Molecular Evolution: Modeling the Myosin Heavy Chain Gene Family," *Biophy. J.*, **65**:2673, 1993.

[2] M. Bzymek and S.T. Lovett. "Instability of Repetitive DNA Sequences: The Role of Replication in Multiple Mechanisms," *Proc. Natl. Acad. Sci. USA.*, **98**:8319, 2001.

[3] C. Cantor, and C. Smith. *Genomics: The Science and Technology Behind the Human Genome Project*, John Wiley and Sons, New York, 1999.

[4] J.A. Clikeman, S.L. Wheeler and J.A. Nickoloff. "Efficient Incorporation of Large [> 2 Kb] Heterologies into Heteroduplex DNA: Pms1/Msh2-Dependent and -Independent Large Loop Mismatch Repair in *Saccharomyces cerevisiae*," *Genetics*, **157**:1481, 2001.

[5] F. Eckstein and D.M.J. Lilley. *DNA Repair*, Springer, NY, 1998.

[6] W. Edelmann, K. Yang, A. Umar, J. Heyer and K. Lau. “Mutation in the Mismatch Repair Gene Msh6 Causes Cancer Susceptibility,” *Cell*, **91**:467, 1997.

[7] S. Fujii, M. Akiyama, K. Aoki, Y. Sugaya, K. Higuchi, M. Hiraoka, Y. Miki, N. Saitoh, K. Yoshiyama, K. Ihara, M. Seki, E. Ohtsubo and H. Maki. “DNA Replication Errors Produced by the Replicative Apparatus of *Escherichia coli*,” *J. Mol. Biol.*, **289**:835, 1999.

[8] John W. Eaton et al.. “GNU Octave,” `http://www.octave.org`, 2001.

[9] R. Ihaka and R. Gentleman. “R: A Language for Data Analysis and Graphics,” *Journal of Computational and Graphical Statistics*, **5**(3):299–314, 1996.

[10] R.E. Lenski and M. Travisano. “Dynamics of Adaptation and Diversification: a 10,000-Generation Experiment with Bacterial Populations,” *Proc. Natl. Acad. Sci. USA*, **91**:6808, 1994.

[11] H. Moore, P.W. Greenwell, C.-P. , N. Arnheim and T.C. Petes. “Triplet Repeats form Secondary Structures that Escape DNA Repair in Yeast”, *Proc. Natl. Acad. Sci. USA.*, **96**:1504, 1999.

[12] G. Myers. “A Dataset Generator for Whole Genome Shotgun Sequencing”, *Proc. Seventh International Conf. on Intell. Syst. for Mol. Biol.*:202, 1999.

[13] Microsoft. “Microsoft Active X Scripting,” `http://msdn.microsoft.com/scripting`, 2001.

[14] NYU Bioinformatics Group. “Valis,” `http://bioinformatics.cat.nyu.edu/valis/`, 2001.

[15] C.-K. Peng, S.V. Buldyrev, A.L. Goldberger, S. Havlin, F. Sciortino, M. Simons, H.E. Stanley. “Long-Range Correlation in Nucleotide Sequences,” *Nature*, **356**:168, 1992.

[16] D.B. Searle. “The Computational Linguistics of Biological Sequences,” *Artificial Intelligence and Molecular Biology*, (Ed. L. Hunter), MIT Press, 1993.

[17] E.A. Sia, R.J. Kokoska, M. Dominska, P. Greenwell and T.D. Petes. “Microsatellite Instability in Yeast: Dependence on Repeat Unit Size and DNA Mismatch Repair Genes,” *Mol. . Biol.*, **17**:2851, 1997.

[18] H.E. Stanley, S.V. Buldyrev, A.L. Goldberger, Z.D. Goldberger, S. Havlin, R.N. Mantegna, S.M. Ossadnik, C.-K. Peng, M. Simons. “Statistical Mechanics in Biology,” *Physica. A.*, **205**:214, 1994.

[19] H.T. Tran, N.P. Degtyareva, N.N. Koloteva, A. Sugino, H. Masumoto, D.A. Gordenin and M.A. Resnick. “Replication Slippage between Distant Short Repeats in *Saccharomyces cerevisiae* Depends on the Direction of Replication and the RAD50 and RAD52 Genes,” *Mol. Cel. Biol.*, **15**:5607, 1995.

[20] H.T. Tran, D.A. Gordenin and M.A. Resnick. “The Prevention of Repeat-Associated Deletions in *Saccharomyces cerevisiae* by Mismatch Repair Depends on Size and Origin of Deletions,” *Genetics*, **143**:1579, 1996.

[21] E. Viguera, D. Canceill and S.D. Ehrlich. “Replication Slippage Involves DNA Polymerase Pausing and Dissociation,” *EMBO J.*, **20**:2587, 2001.

[22] M. Vulic, R.E. Lenski and M. Radman. “Mutation, Recombination, and Incipient Speciation of Bacteria in the Laboratory,” *Proc. Natl. Acad. Sci. USA*, **96**:7348, 1999.

[23] B.J. West and B. Deering. *The Lure of Modern Science — Fractal Thinking*, World Scientific, Singapore, 1995.

A Appenix I

R/S Analysis.

Let $\mathbf{X} = X_1 X_2 \cdots X_m \cdots$ be a sequence over the alphabet ±1.

Consider n disjoint substrings of $\mathbf{X}$ each of length l. Without loss of generality, assume that one such substring is denoted as $X_1 X_2 \ldots X_l$. Let

$$\overline{X} = \frac{\sum_{i=1}^{l} X_i}{l}.$$

Similarly, let its *Range* be defined as

$$R_l = \max_{1\leq t\leq l}\left\{\sum_{i=1}^{t}(X_i - \overline{X})\right\} - \min_{1\leq t\leq l}\left\{\sum_{i=1}^{t}(X_i - \overline{X})\right\}.$$

Its *Standard Deviation* is defined as

$$S_l = \sqrt{\left[\frac{\sum_{i=1}^{l}(X_i - \overline{X})^2}{l}\right]}.$$

Now the R/S is computed for a fixed value of l as

$$(R/S)_l = \frac{\sum_{k=1}^{n}\frac{R_l^{(k)}}{S_l^{(k)}}}{n} = \beta l^{\alpha},$$

where the index k ranges over the n blocks selected. Thus

$$\log(R/S)_l = \alpha\log l + \log\beta,$$

and

$$H = \alpha = \frac{\log(R/S)_l - \log\beta}{\log l}.$$

B Appendix II

The JScript example in Section. 2 can be rewritten in PERL and run in Valis. Many bioinformatics researchers are more familiar with PERL and may prefer PERL over JScript. The flexibility provided by Valis allows the user to maintain backward compatibility with existing bioinformatics code in PERL.

```
#language PERL

$Valis->Clear();

$sql = $Valis->CreateObject("Sql");
$sql->Connect("DSN = mysql; UID = someuser; PWD = somepwd");

$seq = $Valis->CreateObject("DNASeq");
$seq->Input("C:\\GoldenPath\\chr22.fasta");
$seq->SelectSequence(1);
$seq->Display();

$table=$sql->ExecSQL("select name,strand,cdsStart,
                      cdsEnd from genscan
                       where chrom = 'chr22'");
$table->Display();

$a = $Valis->CreateObject("Annotools");
$a->LoadSequence($seq,0);

$b1 = $a->AddBand(1,"AT");
$b2 = $a->AddBand(1,"GC");
$m = $a->AddBand(1,"Masked");
```

```
$bl1 = $a->AddBand(5,"GenScan");

$a->CharBand($b1,"AT");
$a->SetColor($b1,RGB(100,0,0)); #Red

$a->CharBand($b2,"GC");
$a->SetColor($b2,RGB(0,100,0)); #Green

$a->CharBand($m,"N");

$a->SetColor($bl1,RGB(0,200,200)); #Cyan
$a->LoadBlocksFromTable($table,$bl1,2,3,1,0);

$freq = $a->AddBand(4,"Freq");
$a->SetColor($freq,RGB(100,0,100));
$a->SetSize($freq,200);
$a->FindRepeats($freq,14);

$a->Display();
```

C Biographical Notes

C.1 Yi Zhou

Yi Zhou has an MS in Biology from New York University, and is currently continuing with the PhD degree program in Biology at NYU. She has experiences in both molecular biology and computational biology. Her research is currently focused on modeling DNA evolution events, including sequence analysis, biological hypothesis forming, mathematic modeling, computer simulation, and model testing. She is also a member of the NYU Bioinformatics Group.

C.2 Salvatore Paxia

Salvatore Paxia is a Research Scientist with the NYU Bioinformatics Group. He has an undergraduate degree in Electrical Engineering from University of Catania, Italy, an MS in Computer Science from NYU, and is completing his PhD in Computer Science at New York University. He is currently working on the Valis bioinformatics environment, very high level programming languages, a novel free-format database to store biological data, an autostereo 3D display device and special purpose hardware to remotely manage Beowulf computing clusters. He is a co-founder of Ondotek, an information technology company in New York.

C.3 Archisman Rudra

Archisman Rudra completed his PhD in Computer Science at the Courant Institute of Mathematical Sciences in 2002 in Computer Vision, and is now working on the Bioinformatic system, Valis, as a Research Scientist with the NYU Bioinformatics Group. His recent research in biology benefits considerably from the ideas and techniques he had developed in statistics and learning theory in the course of his graduate work. He has an undergraduate degree in Computer Science (BTech, IIT Kharagpur, 1994).

C.4 Bud Mishra

Bud Mishra is a Professor of Computer Science and Mathematics at the Courant Institute of Mathematical Sciences of NYU and a Professor at the Watson School of Biological Sciences of Cold Spring Harbor Laboratory. His earlier education has been in Physics (ISc, Utkal University, 1975), Electronics and Electrical Communication Engineering (BTech, IIT, 1980) and Computer Science (MS and PhD, Carnegie-Mellon University, 1983 and 1985). He is a co-founder of a biotechnology company, OpGen, Inc, specializing in the construction of high-resolution high-throughput genome-wide restriction maps using single molecules. He serves as a scientific advisor or a consultant to several financial, information technology and biotechology companies. His current research focuses on: modeling of cellular and genomic evolutionary processes, Valis: a bioinformatic language and environment, single molecule based technology for mapping, sequencing, karyotyping and haplotyping, microarray-based technology for correspondence mapping and gene-expression analysis with applications to cancer study. Bud Mishra is also a co-director of NYU's recent Center for Comparative and Functional Genomics, a member of the NYU Bioinformatics Group, an Adjunct Professor of Human Genetics at Mount Sinai School of Medicine and an Associated Member at Taub Urban Research Center of Wagner School of Public Policy.