Advanced Relation Model for Genome Sequence Visualization (ARM 4 GSV): Exploratory Visualization Examples

Brian J. d'Auriol
Department of Computer Science
The University of Texas at El Paso
El Paso, TX, USA 79968
Email: dauriol@acm.org

Kavitha Tupelly
Department of Computer Science
The University of Texas at El Paso
El Paso, TX, USA 79968
Email: ktupelly@utep.edu

Abstract—**The Advanced Relation Model for Genome Sequence Visualization (ARM 4 GSV) is proposed in this paper. This model is adapted from an earlier visualization model which has been applied to the visualization of computer programs. A review of the fundamental model components of the earlier visualization model is given. Enhancements so as to make it applicable in genome visualization are discussed. As part of these enhancements, a relational characterization of genome sequences in terms of bases, codons, and patterns such as close inversions is developed and described. An adapted form of the Conceptual Crown Visualization (CCV) model, a part of the earlier work, is discussed. Applications of the ARM 4 GSV to codon usage distribution and close inversion distribution are discussed. These applications are accompanied by many visualization figures of a 269 base RNA molecule, a part of the Hepatitis C virus NS5 gene, Locus AY769711. Our presentation illustrates the extent and flexibility of our approach. We conclude by observing that our objectives of showing the potential usefulness of our proposed visualization model are met.**

I. Introduction

Genome sequencing is an important science effort nowadays. Interesting aspects of sequences include codon usage, identification of palindromes and other similar kinds of subsequences, the distribution of palindromes, and secondary structures in the molecule. Determining a sequence and its properties provides information about the genetic structure of the organism and leads to better understanding of the genetic processes.

There are several databases which store information about genomes, including, nucleotide sequences. Some of these databases are accessible via the National Center for Biotechnology Information (NCBI) [1]. GenBank is very well known for its large storage of nucleotides. The Entrez Nucleotide database, accessed through the Entrez portal at NCBI, includes information from GenBank as well as from other sources. Other databases also exist. Algorithms to data mine these databases as well as to provide sequence analysis have been developed, see for example [2] and the references therein.

Visualization is generally accepted as a powerful approach in enabling understanding of scientific phenomenon. In this case, visualizations on sequence scale, gene scale and genomic scale help to see the structure of DNA or RNA, and its constituent components and properties thereof.

DNA or RNA strands are composed of sequences of bases. These sequences individually may be considered as having a single purpose, for example, as involved in regulating genes or as codons. Collectively, the molecule has a singular purpose in describing the genetic code of an organism. We consider the visualization of sequences pertinent to a single molecule and wish to present a more clear picture of the relationships between the various constituent sequences and the holistic purpose of the molecule. In particular, this paper considers visualizations of codon usage and palindrome distribution within RNA molecules.

Our approach is based on the Advanced Relation Model (ARM) reported in our earlier work [3]–[5]. The ARM has been applied to program visualization: ARM 4 PV. The ARM 4 PV is really a two-part decoupled model specification: in the first part, the ARM abstracts information as a relation hierarchy whereas in the second part, the 4 PV applies visualization techniques over the relation hierarchy where, the hierarchy and visualizations are suitable for program visualization. In a sense, genome sequencing is a biological equivalent of computer programs. We adapt our visualization approach where we couple the ARM with the specifics of genome sequencing, hence, we term our model the Advanced Relation Model for Genome Sequence Visualization: ARM 4 GSV. The focus of this paper is to present a relational hierarchy that is meaningful in genome sequencing and which reflects from the basis of the ARM. We follow this by presenting various visualizations based on those defined in the ARM 4 PV and and thereby, show the extent of the proposed ARM 4 GSV.

This paper is organized as follows. Section II provides a brief overview of genome sequencing and which is given so the casual reader may be able to follow the proposed model. The ARM 4 GSV is proposed in Section III. Section IV presents several visualization examples. Conclusions and discussion are given in Section V.

II. Background

Molecular Biology and Bioinformatics have seen dramatic rise of interests over the recent past. The subject area is quite extensive and a number of excellent books are available on the subject, for example [2], [6]. This section specifically concentrates on the RNA and DNA molecules and sequences of nucleotides contained therein. (Information is cited from [2], [6]).

A nucleotide is a three-component unit made up of a base, a sugar and a phosphate. Nucleotides are identified by a single letter that corresponds with the base: A – adenine, G – guanine, C – cytosine, U – uracil and T – thymine. A linear sequence of nucleotides is identified by a string of these letters, for example, AGC means the three nucleotides in the order of adenine, guanine and cytosine. An RNA molecule is composed of sequences of A, G, C and U while a DNA molecule is composed of sequences of A, G, C and T.

The information stored in RNA and DNA molecules may be described (simplified) as consisting of hierarchal nestings of units. A codon is a grouping of three nucleotides; in a messenger RNA (referred to as mRNA), codons are translated into amino acids during protein synthesis. For example, the sequence GAAAGG would refer to the two amino acids, in order, glutamic acid (Glu) and arginine (Arg). An interesting issue is that decoding a sequence could begin at any index, in the above example, staring at the second nucleotide, the codon would be AAA which refers to the amino acid lysine (Lys). There is a many-to-one relation between the three nucleotide triplet and the specified amino acid, that is, the same acid may be specified by more than one codon. A sequence of codons may specify a gene.

Base-pairing occurs due to hydrogen bonding between the two bases adenine and thymine for DNA, between the two bases adenine and uracil for RNA, or between the two bases cytosine and guanine. Hence, in terms of sequences, this occurs between A-T, A-U and C-G, respectively. DNA is double stranded with the two complementary linear nucleotide sequences base-paired. RNA is composed of a single strand of these sequences. However, relatively short nucleotides subsequences can participate in base-pairing.

One effect of base-parings in RNA is the construction of secondary structures, including, pseudoknots. Understanding secondary structures in RNA helps in understanding the RNA's functions [7]. A number of methods have been proposed to predict and locate such secondary structures; comments regarding these methods are made in [7].

Since our motivation is with respect to our proposed visualization model, we consider specific properties of sequences as follows. A *close inversion* is a sequence (p, s, p') where a subsequence p is separated by an arbitrary subsequence s from its complement subsequence p', $|p| = |p'|$. A *palindrome* is a special case of a close inversion where $|s| = 0$. A *close repeat* is a sequence psp.

III. Advanced Relation Model for Genome Sequence Visualization (ARM 4 GSV)

The Advanced Relation Model for Genome Sequence Visualization (ARM 4 GSV) is based on the

ARM for Program Visualization (ARM 4 PV) [3]–[5]. First, we review the ARM 4 PV placing emphasis on the decoupling of the visualization from the underlying data set. Next we introduce the ARM 4 GSV as extensions to the ARM 4 PV.

A. ARM 4 PV

The ARM 4 PV can be described in two ways. In terms of functionality, the model consists of three phases: (i) identification and extraction of data, (ii) representation and preparation of the data, and (iii) the visualization of the data. In terms of structure, the model consists of a suite of sub-models which collectively provide for this functionality, and, in addition, the model defines potential multiple visualization sub-models that may be incorporated into the ARM 4 PV.

The primary datum in the ARM 4 PV is a *relation* in a *relation hierarchy*. Individual relations are denoted by R_i^l where l refers to the relation's level in the hierarchy. At present, relations are binary. The hierarchy consists of a set of connected relations: $\{R_{i_1}^1, R_{i_2}^2, \ldots, R_{i_l}^l\}$ where $1 \leq i_j \leq n_j$ are integers that denote particular relations in the specified level; $1 \leq j \leq l$. Relations may be one of two types. *Semantic relations* bind both semantics and properties to an object of interest: $R = (o, d, p)$ where o is an object of interest, d represents semantic information about o and p is a property set associated with o. *Constructive relations* are higher-order relations that combine the semantics and properties of two existing lower level relations: $R^l(R^i, R^j, d_\alpha, p_\alpha)$ where $i, j < l$ such that either $i = l - 1$ or $j = l - 1$, and d_α, p_α are obtained by combining the respective information together. Details of this process are discussed below. At present, all Level 1 relations are of the semantic type while all other relations are of the constructive type.

The underlying data for the ARM 4 PV is a computer program. A program consists of individual statements, usually one-per-line in the file. A statement, in this work, is taken to mean a specification of a distinct operation, for example, the calculation of a formula or the output of a variable. In some cases, for example FORTRAN, each line must contain a single statement (statements may be continued over multiple lines). In other cases, for example C, the semicolon separates two statements and hence, multiple statements could be present on the same line. In whatever the case, the ARM 4 PV defines the program statement as the object of interest. Hence, $R^1 = (S, d, p)$ where S is a program statement and d is the statement's semantics (e.g. its operational semantics).

Essentially, the hierarchy establishes the means in which to *propagate* low-level abstractions about individual program statements (objects of interest) to higher-level abstractions about groups of program statements (groups of objects of interest). Let τ denote a semantic operation that combines two semantics and generates an abstraction of those combined: $\tau(d_1, d_2) = d_\alpha$. For example, if d_1 is 'prompt user for input' and d_2 is 'read input value', then $\tau =$ 'interactive user input'. With regards to the property sets, at present, a vector consisting of scalar rational numbers has been defined (determined from specific applications to particular given input programs). Propagation of such property set values is accomplished by averaging. This discussion follows the present definition of the ARM 4 PV. As we extend the model and apply it to genome sequences in this paper, we will propose a more cohesive and comprehensive model of relation propagation.

A simple example taken from [3] illustrates the first two phases of the ARM 4 PV as well as the relation hierarchy. Figure 1 shows a seven line program in pseudocode; each statement is identified by its line number. A corresponding relation hierarchy for this program is shown in Table I. A pictorial representation of the hierarchy is shown in Figure 2. For this example, there are five Level 1 relations, each of which bind the statement's semantics and property set to the associated statement. Here, the semantics represents a simple descriptive meaning of the purpose of the associated statement. The property set, in this example, is composed of a three element vector that corresponds to the degree to which the statement participates in specifying (as opposed to measured performance) a file input/output, a user output, or a computation, respectively. Relation R_1^1 therefore indicates that the purpose of Statement 2 (i.e., `printf("Some column headings")`) is to "print column headings"; furthermore, this statement does not specify any file input/output nor computation, but does

```
1.  fileopen("foo.bar",read-access)
2.  printf("some column headings")
3.  while not eof on file do
4.     readfile
5.     printf(detail report)
6.  end-while
7.  close file
```

Fig. 1. File processing pseudocode program.

TABLE I

TABLE FOR PROGRAM STATEMENT RELATIONS

Level1	**File I/O**	**User Output**	**Comp**
(R_1^1,S2,print column headings)	0.0	1.0	0.0
(R_2^1,S5,print data in columns)	0.0	1.0	0.0
(R_3^1,S3,eof query on file)	0.25	0.0	0.0
(R_4^1,S3,repeat loop-body)	0.0	0.0	0.1
(R_5^1,S4,read a record from a file)	1.0	0.0	0.0
Level2			
(R_1^2,R_1^1,R_2^1,print report)	0.0	1.0	0.0
(R_2^2,R_3^1,R_4^1,repeat eof query)	0.125	0.0	0.05
(R_3^2,R_4^1,R_5^1,repeat file read)	0.5	0.0	0.05
Level3			
(R_1^3,R_2^2,R_3^2,process every record in file)	0.31	0.0	0.05
Level4			
(R_1^4,R_1^2,R_1^3,general file report)	0.15	0.5	0.025

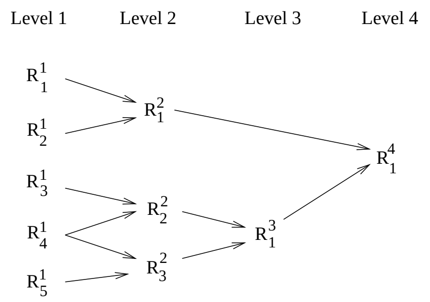

Fig. 2. Pictorial display of relation hierarchy for Figure 1.

specify a user output (to the degree of 100%). (The fact that we do not consider this particular statement to also be file input/output is a matter of technical definition not germane to this paper.) The second level relations are combined as indicated, so, for R_1^2, the semantics of the two lower level relations, “print column headings” and “print data in columns” are combined and abstracted as “print report”. Also, the averages of the corresponding property set are computed, and inserted in the table. This relation abstracts information about two particular program statements. The relation hierarchy is constructed so that the highest level relations represent abstractions (elsewhere in our earlier work referred to as concepts) about groups of program statements.

The decoupled nature of the ARM 4 PV arises from the relation hierarchy definition of the data. The first phase of the model identifies and extracts relations about the underlying program. The second phase constructs a hierarchy of relations (preparation). The third phase allows visualizations of the relational hierarchy. In essence, the ARM part of the model describes a relational abstraction over the underlying data set while the 4 PV part of the model describes the visualization of the relational abstraction.

B. ARM 4 GSV

The approach described here is to concentrate on enhancing the ARM part of the model so that it becomes suitable for applications to genome sequence visualizations. The essential components of the ARM 4 PV are kept intact, in particular, its three-phase functionality, it sub-model structure and the idea of the relational hierarchy. However, in order to adapt the ARM 4 PV as intended, the following components need to be re-defined or enhanced: the selection of objects of interest, the determination of relational abstractions including semantics and property sets over these objects of interest, and propagation functions used to generate constructive relations. The decoupled nature of the ARM 4 GSV (and the ARM 4 PV) allows the transfer of the various existing visualization models. These are described separately later.

The selection of objects of interest is inherently user-driven, and derives from the user’s decision as to which data elements are important or useful for further research exploration. In this paper, we select a nucleotide as the object of interest and justify this by showing the ‘could-be’ usefulness as we construct visualizations of codon usage and palindrome distribution based on this selection.

We now describe the form of the relational abstraction. Figure 3 illustrates this form. Although not used in the subsequent visualizations, we define a Level 0 relation that abstracts the base, sugar and phosphate molecules that make up a nucleotide. Properties of these relations include the base (exactly five base types are allowed), the molecular structure, molecular bonding properties (e.g. bonding energy), and a short text description. In the figure, Level 0 relations are denoted as B, P and S. Level 1 relations are trinary relations that abstract the nucleotides. These model the objects of interest as described earlier. Relation properties include those of the previous level, and, in addition, the index of the nucleotide in the original sequence. In the figure, these relations are denoted as N. Level 2 relations abstract codons. These are also trinary relations that combine three nucleotides. Properties include the three nucleotide bases that describe the codon, the molecular structure, bonding information and a short text description. Due to the issue that a reading frame of three nucleotides defines respective three different codons, a set of Level 2 relations is correspondingly defined and denoted as C$_1$, C$_2$ and C$_3$ respectively. The figure shows these sets. Level 3 relations abstract the amino acids coded by the codon with properties including the amino acid's name and molecular structure. In the figure, A denotes these relations. Level 4 relations are multirelations that abstract subsequences of nucleotides, denoted by s, p or p$'$ in the figure. Properties of these relations include a sequence designation (i.e., some identifying text), its starting index and length, together with the appropriate combinations of the molecular structure from the participating lower-level relations. Since close inversions, palindromes and close repeats are defined over sequences, an additional property to identify relations that participate in these special formations is needed. Level 5 relations abstract the secondary structures of the sequence, for example, pseudo-knots where such are defined purely by close inversions or palindromes. In the figure, n denotes these relations. Lastly, Level 6 relations abstract a gene, denoted by G in the figure. We note the utility of connecting Level 3 to Level 4 relations, and in so doing, the necessary requirement of connecting Level 2 relations to Level 4 as well. This provides information about the sequence of amino acides coded by the respective subsequences and is useful for the subsequent visualization of codon usage. In the figure, these connections are shown by the dashed lines (for visual clarity, we connect these dashed lines to a different subsequence). The figure also shows the dual linear and three dimensional structuring of this relational hierarchy. The linearity comes from the fact that the $i+1$ level may have participating relations at the i level (although neither illustrated nor explicitly discussed here, Level 5 relations abstract secondary structures that make up genes). The three dimensionality comes from the fact that codons and amino acids are non-necessary components of sequences, hence, can be considered to 'fill-in' the third dimensional space equivalent to the importance of the nucleotides (similar for the pseudo-structures defined by Level 5 relations).

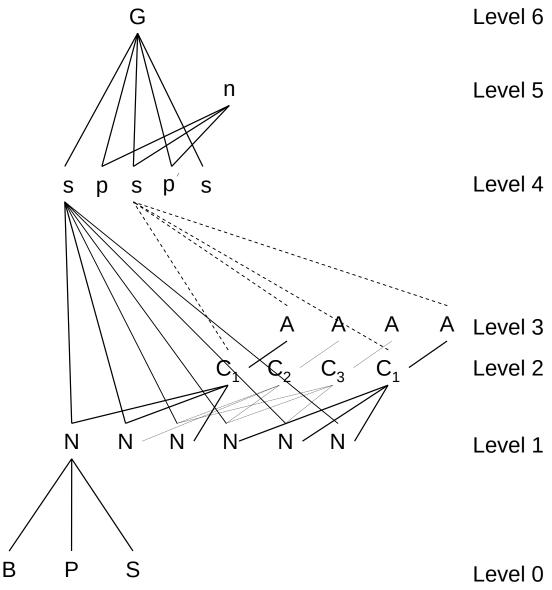

Fig. 3. Generic structure of the ARM 4 GSV relation hierarchy.

The example in Figure 4 illustrates the construction of a relational hierarchy for a portion of the Hepatitis C virus NS5 gene. The sequence was taken from the NCBI Nucleotide database, Locus AY769711, and consists of 269 bases. In the example, we show a close inversion at indices 1–6 and 253–258.

Although we do not explicitly express the transcription process in terms of our model, we do point out that the relational hierarchy supports this process as follows. Let Level 4 relations be supported by Level 1 through 3 relations (as discussed earlier), then, construct an operation such that models ribosome bonding to sequences (perhaps by iterating over the sequences as defined by the Level 4 relations). The effects of a 'slip' of the ribosome may be modeled by shifting from the C$_i$ to the C$_{i-1}$ modulo

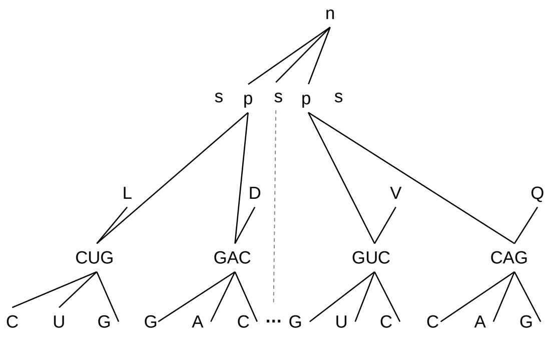

Fig. 4. Relational hierarchy illustration for a portion of the Hepatitis C virus NS5 gene.

three Level 3 Relation sets (which are available as the participating relations in the Level 4 sequence relations). Clearly, clarification of this process is beyond the scope of this paper, nevertheless, this discussion motivates future work.

The form of the relation hierarchy required for genome sequence visualization via the ARM 4 GSV is somewhat changed than that used in the earlier work with program visualization. In the earlier work, all the relations are binary with a clear upwards propagation of semantics and properties extending from the only objects of interest identified at the lower level. In particular, the next higher level is based on at least one relation at the immediate lower level. As with the earlier work, we have defined a singular object of interest. However, this said, there are semantic bindings at the higher levels that do not follow from the mere propagation of lower level bindings. For example, Level 3 relations abstract amino acids, the existence of which provides added semantic information not obtainable from the relation propagation. Consequently, the notion of semantic versus constructive relation is modified to include a third category, namely, semantic-constructive, that reflects the nature of these relations. For these relations, semantics based on the relation's level provides a simplistic definition for modeling the semantics of these additional objects of interest. In addition, the limited alphabet constrains the possible types of relations, for example, there are exactly five types of Level 0 relations, five types of Level 1 relations, 20 types of Level 2 relations and 20 types of Level 3.

Propagation of some of the properties of the relations is natural, for example, the molecular structure of the three Level 0 relations that support a Level 1 relation are propagated by combining these into a single molecular structure for the designated nucleotide. Chemical rules may be applied to ensure the correct propagation. However, given the semantic-constructive nature of this relational set, it is likely easier to merely inject the molecular structure based on the nucleotide type.

IV. Visualization

The visualization phase of the ARM 4 GSV inherits the visualization models that have been defined in (or may yet be defined) the ARM 4 PV. Four such models are described in [3]–[5]. Each model emphasizes certain aspects of the data in the relational hierarchy. The idea is that suitable combinations of the visualizations taken from the multiple models combine to provide heightened understanding of the data to the users. The presentations in some of the earlier work illustrate this idea. Here, to maintain focus within the scope of the paper, we describe and apply one of these visualization models.

The Conceptual Crown Visualization (CCV) model provides concept visualizations to facilitate the viewer's better understanding of the concepts inherent in the relational hierarchy. Here, concepts are defined by the relative abstractness of the relation, in particular, the semantics of relations at a higher level define higher level concepts. There are two basic types of visualizations that are defined: a line structure and a space structure visualization. The former renders selected relations in the relation hierarchy as either single vertical lines (Level 1) or multiple piece-wise single point-connected lines (higher levels) whereas, the latter renders concepts as a convex hull of the participating lower-level relations.

Relations are graphed in the $x - y$ plane. The relation's level is mapped to the y-axis. The linear x-axis is ordinal, that is, represents an ordering of instances of the objects of interest. With respect to computer programs, program statements in lexicographic order are mapped to increasing ordinal numbers on the x-axis. With respect to genome sequence visualization as proposed in this paper, nucleotides are mapped to the x-axis such that the index of the nucleotide is identified with its value on the axis.

For perceptive reasons, the x-axis itself is mapped to a circle in the viewer's coordinates thereby creating a cylinder shaped visual object. This enables: (a) immersive visualization by allowing the viewpoint to be placed in the center of the circle, (b) a zooming operation by providing greater forefront focus while compressing the information in the peripheral area, and (c) greater information density by providing up to twice the amount of information displayed on the screen. Further details are described in [8].

Although the CCV model presents the relational hierarchy for visualization, the current systems implementation of the model is limited to binary relations and, moreover, is targeted for the ARM 4 PV. We consider two types of work-arounds, first, we adapt the relational hierarchy proposed in this paper to include binary relations and introduce middle layers used to assemble the required relationships, second, we construct multiple unary or binary relations that describe a single common abstraction. (We intend in the future to re-implement the CCV model to natively support the multi-relational needs of this application.)

Visualizations of sequences have been reported in the literature. The VISTA toolset [9]–[11] includes the VISTA browser to view visualizations of sequences and comparisons between sequences. Additional visualization models and programs are also available. Some plotting and visualization tools are provided as part of the European Molecular Biology Open Software Suite (EMBOSS) [12]. A visualization system called GenomePlot is reported in [13] where genomic scale visualizations are illustrated.

The subsequent visualizations are based on the 269 base sequence for a portion of the Hepatitis C virus NS5 gene, Locus AY769711 (i.e., the same dataset used in Figure 4). Although the primary emphasis with respect to these visualizations is to show the application of our proposed visualization model and further, we lack the biological background to assess the realism and usefulness of the visualization, we feel that these example visualizations illustrate the potential of our approach.

The visualization system experimental prototype used to generate these visualizations has three parts, first, the relational abstraction is prepared, second, the CCV model is applied to map the relations to a 2-D plane and subsequently to manipulate the presentation in terms of line, space or landscape plots, and third, AVS/Express is used to render the images and provide user interactive features (e.g. rotatation, zooming, perspective viewing).

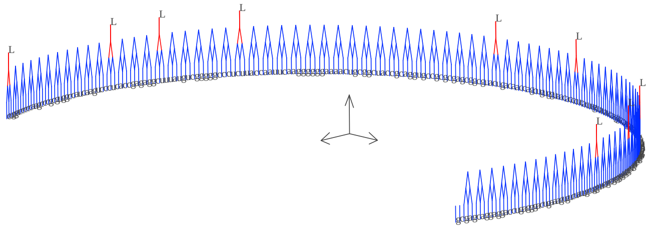

Fig. 5. Codon usage visualization: binary relationships, angle factor of 12, Leucine coding highlighted in red, top front left-rotated view.

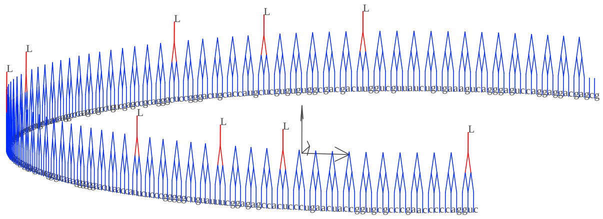

Fig. 6. Codon usage visualization: binary relationships, angle factor of 12, Leucine coding highlighted in red, top side right-rotated view.

A. *Codon Usage Distribution*

In this section we describe various presentations of the CCV model applied in the area of codon usage. Common aspects of Figures 5 through 18 include: a circular x-axis at varying degrees of closure with the nucleotide sequence ordered left-to-right and (in most of the figures) displayed below a corresponding vertical line, various rotated views, a three dimensional axis located near the center of the image, and highlighting via colors. Specific aspects are described below.

Figures 5, 6 and 7 present a codon as a two-level hierarchy of binary relations; this is best seen in the zoomed image, Figure 7. The select amino acid of interest, Leucine (L) in this case, is highlighted in red. The distribution of L throughout the sequence is readily identifiable from the first two figures. In addition, these figures also allow the user to visually inspect all regions of the sequence, for example, both the beginning and ending parts of the sequence is easily viewable in Figures 5 and 6.

Figures 8, 9 and 10 present a codon using a

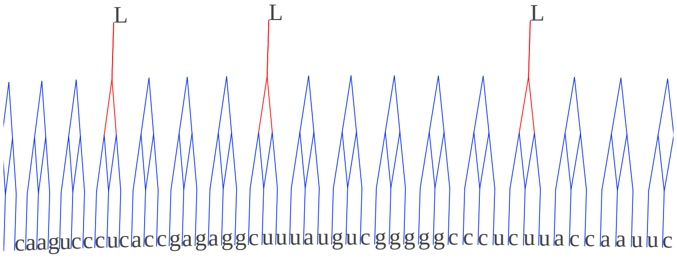

Fig. 7. Codon usage visualization: binary relationships, angle factor of 12, Leucine coding highlighted in red, front zoomed view.

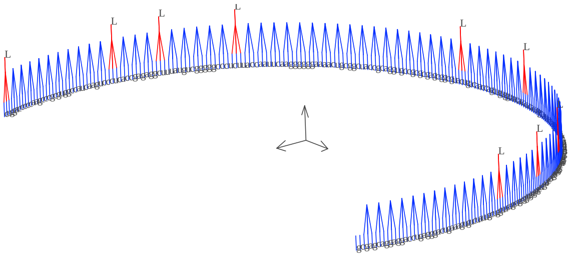

Fig. 8. Codon usage visualization: trinary relationships, angle factor of 12, Leucine coding highlighted in red, top front left-rotated view.

trinary relation (due to the systems limitations noted earlier, there is actually one binary relation connecting the first and the third nucleotide and one unary relation connecting the second nucleotide such that the relations describe the same abstractions; due to the CCV model, these dual relations are rendered in appearance as a single trinary relation). Figure 10 best shows the structural difference of the presentation with the previous method. The highlighting of the selected amino acid (also L in these figures) is more visible due to the single abstraction of the trinary relationship. Again, these figures allow all regions of the sequence to be visually inspected, including, both the begging and ending portions.

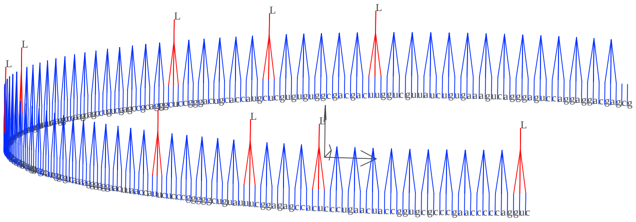

Fig. 9. Codon usage visualization: trinary relationships, angle factor of 12, Leucine coding highlighted in red, top side right-rotated view.

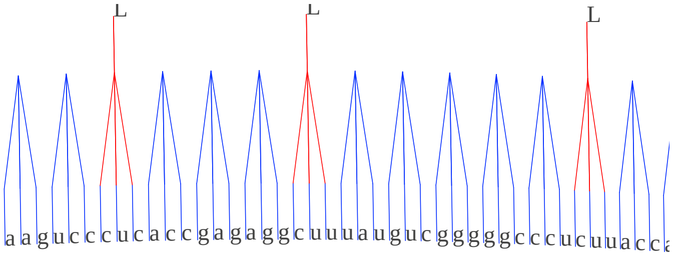

Fig. 10. Codon usage visualization: trinary relationships, angle factor of 12, Leucine coding highlighted in red, front zoomed view.

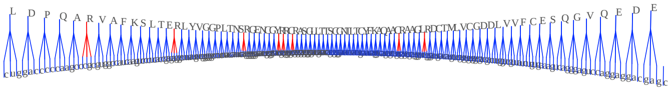
L D P Q A R V A F K S L T E R L Y V G P L T
V C G D D L V V F C E S Q G V Q E D E

Fig. 11. Codon usage visualization: trinary relationships, angle factor of 12, Arginine coding highlighted in red, all acids displayed, front view in perspective.

Figures 11 and 12 show several variations; the latter is a zoomed-in image. First, perspective viewing is enabled. This provides heightened immersive experiences to the user including support of placing the user within the image, hence, the image extends from in-front-of the user out along both sides and beyond the user. Peripheral vision is therefore enabled as well. Figure 12 provides some illustration of this experience. Second, all of the amino acids that are encoded are displayed. Third, Arginine (R) is highlighted.

Figures 13, 14 and 15 show a space-structure variation of the previous figures. The angle is more closed than the earlier figures as well. Here, the trinary relations are replaced by the corresponding triangle. The figures are displayed using various rotational views. Although difficult to see in the printed form, the first two figures use a unidirectional light source from the interior of the figure, thereby, the interior surface is rendered more

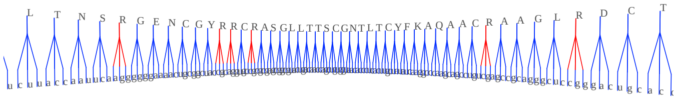
L T N S R G E N C G Y R R C R A S G L L T T S C G N T L T C Y F K A Q A A C R A A G L R D C T

Fig. 12. Codon usage visualization: trinary relationships, angle factor of 12, Arginine coding highlighted in red, all acids displayed, front zoomed view in perspective.

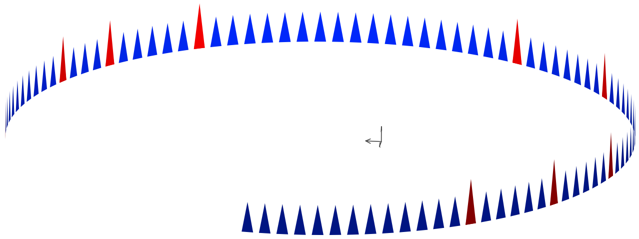

Fig. 13. Codon usage visualization: triangle represented relationships, angle factor of 16, Leucine coding highlighted in red, top front left-rotated view, unidirectional (interior) lighting.

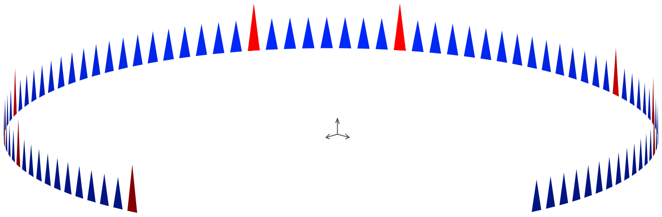

Fig. 14. Codon usage visualization: triangle represented relationships, angle factor of 16, Leucine coding highlighted in red, top front view, unidirectional (interior) lighting.

brightly than the exterior surface. One advantage is that heightened depth-cues are provided to the user thereby making it easier for the user to maintain visual identification with the image. Figure 15 however, uses bi-directional lighting, thereby, both the interior and exterior surfaces are shown brightly.

Figures 16, 17 and 18 highlight the various codons that code for L. There are six such codons, of which, the following appear: two coded by ‘cug’ shown in cyan, four by ‘cuc’ shown in green, two by ‘cuu’ shown in red, and one by ‘uug’ shown in purple. In addition, these figures display the sequence in a shallow curve thereby allowing better printed visualizations.

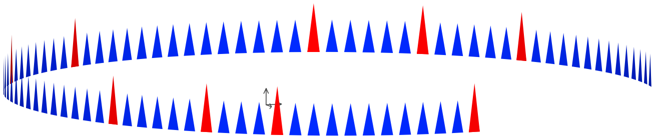

Fig. 15. Codon usage visualization: triangle represented relationships, angle factor of 16, Leucine coding highlighted in red, top side right-rotated view, bi-direction (interior and exterior) lighting.

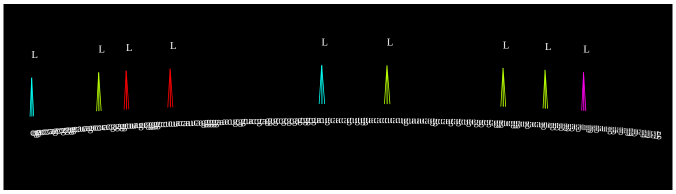

Fig. 16. Codon usage visualization: triangle represented relationships, angle factor of 6, Leucine codings highlighted, front view, black background.

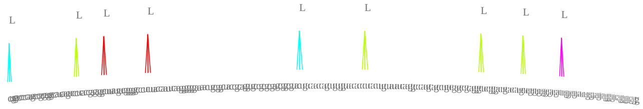

Fig. 17. Codon usage visualization: triangle represented relationships, angle factor of 6, Leucine codings highlighted in red, front view, white background.

Informal comparisons of these figures lead us to the following specific observations. First, trinary relations are more appealing than the binary counterparts. Second, space structure visualizations provide the same visual information with less visual clutter.

B. Close Inversion, Palindrome and Close Repeat Distribution

The modeling of close inversions, palindromes and close repeats also extends naturally from the relational hierarchy. A single Level 5 binary relation is used to connect p with p' or, for repeats, p with p. The figures shown in this section present these relations in various ways.

The ‘palindrome’ program of the European Molecular Biology Open Software Suite (EMBOSS) is used to generate the close inversions displayed

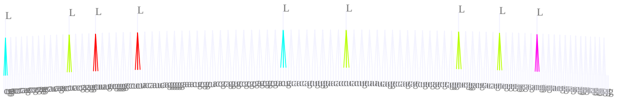

Fig. 18. Codon usage visualization: triangle represented relationships, angle factor of 6, Leucine codings highlighted, front view, white background with white-out applied.

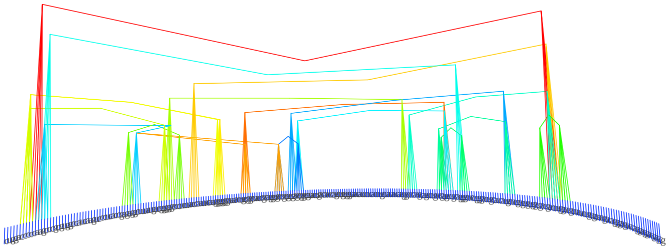

Fig. 19. Palindrome distribution visualization: palindrome length of 5 with no maximum gap between elements, no mismatches allowed, perspective front view, height ordered by gap between elements, each palindrome is colored individually

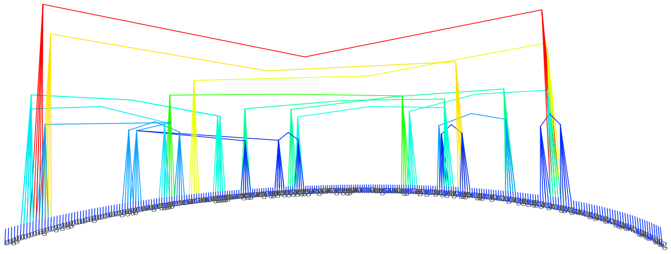

Fig. 20. Palindrome distribution visualization: palindrome length of 5 with no maximum gap between elements, no mismatches allowed, perspective front view, height and color coding ordered by gap between elements.

in this section. For convenience, the output of the EMBOSS software is filtered to provide the input into our visualization system.

Figures 19, 20 and 21 show the distribution of close inversions of length five which occur anywhere in the sequence. Both color and height are used to highlight properties. In all three figures, height is based on $|s|$, i.e., the length of the gap between p and p'. In Figure 19 each close inversion is represented by a distinct color, thereby, aiding navigation during the visualization. However, in the remaining two figures, color is mapped as the height, thereby, aiding identification of small versus large gaps. Both Figures 19 and 20 use perspective viewing. Figure 21 is a top view, that is, a 90 degree rotation (tilt) about the x-axis.

Figures 22, 23 and 24 show close inversions of lengths four and five for $|s| \leq 20$. Both color and height represent $|p|$, i.e., the length of the close inversion. The first two figures use perspective views, the first of which use a blue-purple color coding whereas the second uses a blue-red coding. Figure 24 shows a front tilted view.

A landscape visualization is an adaptation of the space structure variation of the CCV where the z-axis is used to plot additional information. Figures 25 and 26 show front-tilted and front-tilted-rotated views, respectively, of the previous three figures, namely, of close inversions of lengths four and five for $|s| \leq 20$. Both of these landscape visualizations also use color and height to represent

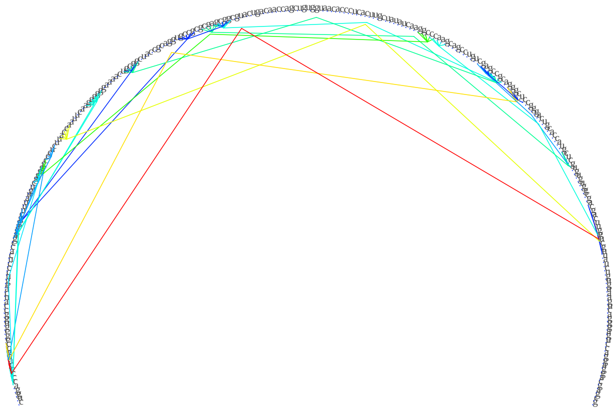

Fig. 21. Palindrome distribution visualization: palindrome length of 5 with no maximum gap between elements, no mismatches allowed, top view, height and color coding ordered by gap between elements.

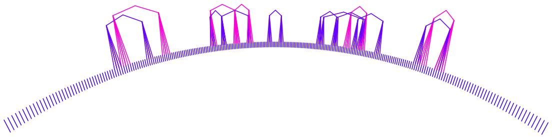

Fig. 22. Palindrome distribution visualization: palindrome lengths of 4 and 5 with maximum gap between elements of 20, no mismatches allowed, front perspective view, height and color coding ordered by palindrome length.

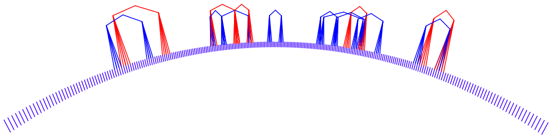

Fig. 23. Palindrome distribution visualization: palindrome lengths of 4 and 5 with maximum gap between elements of 20, no mismatches allowed, front perspective view, height and color coding ordered by palindrome length.

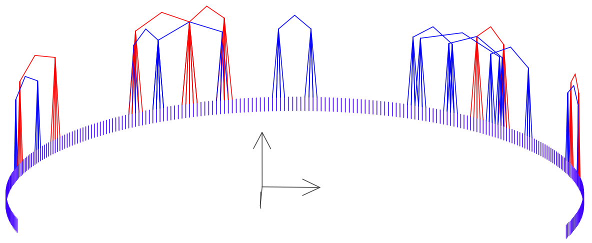

Fig. 24. Palindrome distribution visualization: palindrome lengths of 4 and 5 with maximum gap between elements of 20, no mismatches allowed, front rotated view, height and color coding ordered by palindrome length.

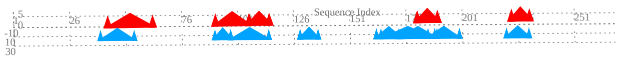

Fig. 25. Palindrome distribution visualization: palindrome lengths of 4 and 5 with maximum gap between elements of 20, no mismatches allowed, landscape front view.

palindrome length, blue is used for $|p| = 4$ and red is used for $|p| = 5$. For this set of visualizations, we have also included the output of the EMBOSS 'palindrome' program in Figure 27 specific to the generation of relations for $|p| = 4$.

The homogenius data abstraction of relations over the underlying domain data set of genome sequences allows for the visualization of combinations of codon usage distribution and close inversion distribution in the same visualization. Figure 28 shows Leucine (L) distribution in blue together with close inversions of length five where $|s| \leq 20$ in red. The uninteresting relations have been whited out for clearer visual inspection. This figure employs a perspective view.

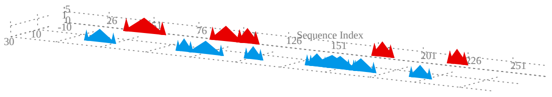

Fig. 26. Palindrome distribution visualization: palindrome lengths of 4 and 5 with maximum gap between elements of 20, no mismatches allowed, landscape rotated view.

```
Palindromes of:
Sequence length is: 269
Start at position: 1
End at position: 269
Minimum length of Palindromes is: 4
Maximum length of Palindromes is: 10
Maximum gap between elements is: 20
Number of mismatches allowed in Palindrome: 0
```

Fig. 27. EMBOSS palindrome program output corresponding to part of the input data set used for the landscape visualizations.

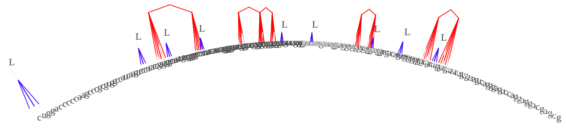

Fig. 28. Combined coden usage and palindrome distribution visualization: Leucine coding highlighted in blue, palindrome lengths of 5 with maximum gap between elements of 20, no mismatches allowed, perspective front view.

V. Conclusions

This paper is focused on adapting and extending as necessary the Advanced Relation Model for Program Visualization (ARM 4 PV) to apply to visualizations of genome sequences. The new model is termed the Advanced Relation Model for Genome Sequence Visualization (ARM 4 GSV). We have presented a decoupled version of the model which allows for the adaptation and subsequent extensions. We have proposed a relational hierarchy based model of various aspects of genome sequences; this hierarchy is based on the ARM. We have also adapted the Conceptual Crown Visualization (CCV) model and have applied it to a simple genome sequence. Lastly, we have shown some visualizations based on the ARM 4 GSV. We believe that our method provides useful information and insight into genome sequences. This satisfies the goal of this paper.

Clearly, we have demonstrated possibility and potential; but lack conclusive results based on realistic needs of the research community. This is an area that requires much additional work, namely, to couple our visualization model with the analytic needs of the molecular biology and bioinformatics researcher so as to provide more conclusive results.

Some insight into additional future work is now described. First, we have illustrated sequence scale visualizations. We feel that adaptation via multiple concentric circles (i.e., by mapping data to the z-axis) as well as stacking images on top of each other (i.e., by mapping data to multiple levels on the y-axis), we can provide for visualizations of six to nine kilo-bases, thereby, enabling small gene scale visualizations. Second, much greater systems development is needed to provide for a useful system that can be exported to users. Third, although we have demonstrated only one visualization model in this paper, the ARM 4 GSV via inheriting visualization models from the ARM 4 PV, contains additional visualization models. Investigation into adapting and enhancing these additional models is needed. Fourth, in this paper we had selected the nucleotide as the object of interest. Other objects of interest could be eligible for considerations.

VI. Acknowledgments

We thank Prof. Ming-Ying Leung for her many helpful comments on this work and Dr. Dag von Lubitz for his interest and support.

References

[1] "National center for biotechnology information," Nov. 22 2004. [Online]. Available: http://www.ncbi.nlm.nih.gov

[2] M. S. Waterman, *Introduction to Computational Biology, Maps, sequences and genomes*. 2000 N.W. Corporate Blvd., Boca Raton, FL 33431: Chapman and Hall/CRC, 2000.

[3] B. J. d'Auriol, "Advanced relation model for program visualization (arm 4 pv)," in *Post-conference proceedings of The 2004 International Conference on Modeling, Simulation and Visualization Methods (MSV'04)*, I. A. A. Hamid R. Arabnia and G. A. Gravvanis, Eds. Monte Carlo Resort, Las Vegas, NV, USA: CSREA Press, June 2004, in press.

[4] ——, "A concept visualization study of a parallel computing program," in *Proceedings of the 2004 Internatioal Conference on Parallel Processing Workshops (ICPP Workshops),*, Y. Yang, Ed. Montreal, Canada: IEEE Computer Society, August 2004, pp. 239–246.

[5] ——, "Concept visualizations of computer programs," in *roceedings of the International Conference on Advances in Internet Technologies and Applications, with special emphasis on E-Education, E-Enterprise, E-Manufacturing, E-Mobility, and related issues (CAITA 2004),*, Purdue University, West Lafayette, Indiana, USA, July 2004, published on CD, ISBN: 86-7466-117-3.

[6] B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, and J. D. Watson, *Molecular Biology of The Cell, Third Edition*. 29 W. 35th St., New York, NY, USA 10001-2299: Garland Publishing, Taylor & Francis Group, 2000.

[7] R. D. Dowell and S. R. Eddy, "Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction," *BMC Bioinformatics*, vol. 5, no. 1, p. 71, 2004. [Online]. Available: http://www.biomedcentral.com/1471-2105/5/71

[8] A. Gajjala, "A model for visualization of program conceptual information," Master's thesis, Department of Computer Science, The University of Texas at El Paso, May 2004.

[9] "Vista," Nov. 22 2004. [Online]. Available: http://gsd.lbl.gov/vista/index.shtml

[10] C. Mayor, M. Brudno, J. R. Schwatz, A. Poliakov, E. Rubin, K. A. Frazer, L. S. Pachter, and I. Dubchak, "Vista: Visualizing global dna sequence alignments of arbitrary length." *Bioinformatics*, vol. 16, no. 1046.

[11] I. Dubchak, M. Brudno, G. G. Loots, C. Mayor, L. Pachter, E. Rubin, and K. A. Frazer, "Active conservation of noncoding sequences revealed by 3-way species comparisons," *Genome Research*, vol. 10, no. 1304.

[12] EMBOSS. [Online]. Available: http://emboss.sourceforge.net/

[13] R. Gibson and D. R. Smith, "Genome visualization made fast and simple," *Bioinformatics*, vol. 19, no. 11, pp. 1449–1450, 2003.