Genome sequences contain a number of patterns that have biomedical significance. Repetitive sequences of various kinds are a primary component of most of the genomic sequence patterns. We extended the suffix-array based Biological Language Modeling Toolkit to compute n-gram frequencies as well as n-gram language-model based perplexity in windows over the whole genome sequence to find biologically relevant patterns. We present the suite of tools and their application for analysis on whole human genome sequence.

Keywords: Statistical language modeling; n-gram analysis; genome sequence analysis.
1. Introduction

N-grams, or sequences of n-words, are commonly used in statistical language modeling of natural language texts. We have previously demonstrated that n-gram language modeling of proteome sequences, treating amino acids as equivalents of words, can lead to biologically relevant insights. Rare n-grams in proteomes have been shown to be correlated to folding core in protein structure, and to proteome signatures and a text summarization technique called latent semantic analysis to protein structure prediction. These methods have subsequently been applied by others to biological sequence analysis in computational biology. For text, an n-gram refers to a sequence of n words; for protein or genome sequences, an n-gram refers to n amino acids or nucleotides. A compilation of the n-grams and their frequency of occurrence in text can be applied to classify documents by authors, topics or genres.

Human genome sequence contains nearly three billion nucleotides. Of this, merely 2% is made of coding regions; another 2% is made up of regulatory elements and the like. Nearly 66% of the genome, namely 2 billion nucleotides is made up of repetitive elements. Repetitions range from very short fragments of two to six nucleotides repeating many times in tandem, to a few thousand nucleotides repeated once or twice in the whole genome. Various well-defined repetitive patterns have been discovered previously and are noted to be important in the functionality of a gene, the genetic pathology of a disease, or in evolution.

In this paper, we describe the tools that we developed for the analysis of whole genome sequences and demonstrate the application of n-gram analysis and statistical language modeling to find repeat-rich regions in genome sequences, and demonstrate them by application to human chromosome X and chromosome 19.

2. Patterns in Genome Sequences

Satellite DNA are tandem repeats that occur in DNA. They are classified as satellites, minisatellite or microsatellites, depending on whether they are large repeating sequences that span megabases, or 6–100 nucleotide repeats that span hundreds of bases, or 1–5 nucleotide repeats that span few tens of nucleotides, respectively. Minisatellites are also called variable number tandem repeats (VNTRs), and the microsatellites are also called short tandem repeats (STRs). Trinucleotide diseases such as myotonic dystrophy and fragile X syndrome, are caused by an expanded repeated region in VNTRs, where the larger the expansion, the more severe the disease. Certain characteristic STRs are found to be located in or near disease-causing genes, including those associated with prostate cancer. Satellite DNA is widespread throughout the human genome and shows sufficient variability among individuals in a population that they have become important in genetic mapping, linkage analysis and human identity testing.

DNA palindromes are like palindromes defined in natural language but of the form \( \ldots s_4 s_3 s_2 s_1 s'_1 s'_2 s'_3 s'_4 \ldots \) where \( s'_1 \) is the complement of nucleotide \( s_1 \), such as in...
ACCTGCAGGT. DNA palindromes are associated with the initiation of gene amplification and tumor proliferation\textsuperscript{27} as well as initiating DNA rearrangement\textsuperscript{28}.

G-quartets are fragments of DNA with a high frequency of G residues (usually in sets of four), which result in the formation of G4 DNA, a highly stable alternative structure to the double strand. G-quartet structure in a gene can cause Huntington’s disease, where the structure causes inhibition of protein aggregation\textsuperscript{29,30}.

Short and long interspersed nuclear elements (SINEs and LINEs) are retrotransposons that are dispersed throughout the genome multiple times. SINEs are shorter than 500 nucleotides in length. Their occurrence within gene regions can result in diseases such as X-linked dystonia-parkinsonism\textsuperscript{31}. LINEs are usually longer than 1000 nucleotides. They are known to cause hemophilia A and Duchene muscular dystrophy\textsuperscript{32}.

Alu repeats (named after the bacteria \textit{Arthrobacter luteus}) are composed of more complicated patterns, containing a well-defined regular expression (e.g. A\textsubscript{5}TACA\textsubscript{6}), a high CG content and ending with a poly-A tail. They play an important role in the study of human genetics. Due to their mobility, the location of their re-insertions can have a negative effect and can lead to diseases, including cancer\textsuperscript{33}.

\textit{CpG islands} are a cluster of CG dinucleotides dispersed frequently over a sequence of 200 to 3000 nucleotides with an overall C+G content of 50\% or higher. They are found as buffers between coding and non-coding regions\textsuperscript{18}. Abnormal methylation of the C of the CG dinucleotides can cause downregulation or silencing of the gene, resulting in disease-specific DNA methylation patterns\textsuperscript{34,35}.

Identifying repeats or other patterns of structural relevance in a genome, such as palindromes and G-quartets, provides insight into their functional role as well as their role in diseases. Several experimental methods have been devised for locating specific patterns in the DNA. However, as with other experimental methods, they are more time-consuming and less versatile than computational methods. With the availability of whole genome sequences for multiple organisms, as well as for many individuals of a given species, and with the advances made in the fields of computer science and large-scale information processing, it is due course of time that computational tools for large-scale pattern mining and discovery in genome sequences be developed. Let us call this area of pattern mining and discovery in large-scale biological sequences \textbf{patternomics}.

Advances in computer science that enable high-end patternomics are data structures and algorithms such as B-trees, suffix tries, suffix trees, suffix arrays, string B-trees and their variants and a number of optimized algorithms for construction of these data structures for a given input (genome) sequence\textsuperscript{36}. Of these, suffix arrays are one of the most suitable data structures, requiring linear space and time requirements. Ever since large-scale genome sequence data became available, the applicability of these data structures to pattern searches in genome sequences has been demonstrated. Since the genome sequence is invariant (for a given build), what would have been useful was to construct these data structures for all the available genome sequences, so that all future pattern mining algorithms may be computed on
these preprocessed data. However, this trend has not been observed. They have not yet been deployed as a ready-to-use preprocessed data but have only been applied independently in different tools.

The tools currently available for pattern searching in genome sequences are usually tailor made for specific types of patterns or are restricted by the amount of data they can handle. QUASAR, for local alignment matching, is restricted to aligning only strongly similar sequences; OASIS, for dynamic programming A*-search driven by a suffix-tree index, filters the input database allowing one to consider only a small subset of the database; CONSERV, to detect exact matches in two or more complete genome sequences, does not scale well with large database as it works basically on primary memory; STRMAT, to compute the tandem repeats supports only less than 0.5 MB of data; Tacg, to search for regular expressions, is not efficient for chromosome sizes and also does not handle regular expressions such as what is required to describe a palindrome. GESTALT, for mining CpG islands, and EMBOSS, a suite of about 150 tools for a variety of sequence analysis operations, do not scale for even single chromosome sizes; GESTALT can handle only 4 MB whereas EMBOSS failed repeatedly for even chromosome 21, one of the smallest human chromosomes measuring only 30 MB. Furthermore, there are tools like Vector NTI, which promise computational speed and provide visualization but are only commercially available. RepeatMasker uses a genomic search algorithm either WU-BLAST or CrossMatch (http://www.phrap.org/) to screen a DNA sequence for repeats and low-complexity DNA. DNA-SCANNER scans DNA sequence for a number of different properties such as biophysical, energy, potential for protein interactions and sequence-based features such as T density, AT density, etc. It scans the DNA through sliding window mechanism. MEME is an online tool for discovering protein and DNA sequence motifs. BLAT, the BLAST-like alignment tool to find nearly exact matches can quickly find sequences of 95% and greater similarity of length 40 bases or more. It is useful in matching closely related sequences such as mRNAs to a genomic sequence, usually within the same species. REPFFIND finds clustered exact repeats and SSAHA (sequence search and alignment by hashing algorithm) performs fast matching and alignment of DNA sequences. An approach to search short tandem repeats (microsatellites) in genome sequence was developed by specifying errors, gaps and copy numbers. The difference between this method and other tools such as Tandem Repeats Finder is that in this parameters are only gaps, mutations, copy numbers, which are easy to understand for biologists as opposed to specifying other parameters like alignment weights, matching probability, etc., which are required as inputs to the tool. The disadvantage of these existing methods is that they are disparate tools with different types of strengths and limitations; they consider only a small subset of the database or have limitations on the amount of data they can handle on account of memory requirements.

Although there have been numerous algorithms to locate specific patterns in genome sequences, there are no tools that offer scalability, versatility and efficiency. For instance, in the case of EMBOSS, though it provides a wide variety of tools, it does not use effective data structures like suffix arrays to pre-process the data and
hence does not scale for larger data. It has been shown that EMBOSS tool for finding DNA palindromes does not scale well for even sequences of size of 1 MB. EMBOSS tool (Palindrome Analysis by BLAST Program) has a greater efficiency in mining DNA Palindromes, but it requires time in the order of several hours to complete for sequences of size of few MBs and does not scale linearly with larger data. As several human chromosomes span more than 100 MBs, these tools cannot efficiently pattern mine them. STRMAT, which uses a suffix tree, has scalability issues arising from the higher memory requirements of suffix trees (in comparison to suffix arrays) and implementation issues. This restricts the tool from being able to analyze sequences more than 10 MB long.

3. Augmented Biological Language Modeling Toolkit (BLMT version 3.0)

An n-gram is a sequence of n nucleotides (or amino acids). The analyses presented here are based on the tools in Biological Language Modeling Toolkit (BLMT) previously developed, which constructs suffix array, longest common prefix array and rank array, for efficient computation of n-gram patterns. For the work presented here, additional tools have been developed to merge suffix arrays, and to carry out perplexity analysis, as described below. This augmented toolkit is BLMT v3.0, or more generally referred to as BLMT in the rest of the manuscript.

3.1. Functional organization of tools in BLMT

BLMT is built in three layers of functionality (Fig. 1).

**Foundation Layer:** It consists of tools that pre-process the genome sequence for efficient pattern matching. A suffix array, and its longest common prefix array and rank array are constructed for the whole sequence. The products computed in this layer are written to files in a well-defined file format, so that they may be used in

---

![Fig. 1. BLMT version 3: Logical organization of the tools.](image-url)
future application development. This layer is similar to the first version of BLMT originally developed by Ganapathiraju and Klein-Seetharaman, except that the suffix array construction tool has been reimplemented here such that it can scale to large data sizes. This layer forms the foundation for all further pattern analysis tools and applications. On account of data pre-processing by these tools, most patterns are mined in less than a few seconds in even the largest human chromosome.

**Features Layer:** It provides tools to compute various n-gram based features of the sequences. Examples are n-gram counts, search for occurrences of specified n-gram and a statistical language model that gives a measure of perplexity in the sequence. These characteristics can be computed over the entire genome or only over a specified segment of interest. It is also possible to compute the features in moving windows over the entire or partial sequence.

**Applications Layer:** Special purpose programs for specific applications reside in this layer. They directly make use of the output of the foundation layer or use the tools in the features layer to implement the specific application. For example, all the applications described in the results section would be placed in this logical layer.

**3.2. Open source release for future developments**

The source code for the toolkit is being made available in Open Source. All the algorithms are written in C and made available as a library of functions as well as stand-alone applications. Biomedical, genomics, genetics and proteomics community can use the existing tools to study a number of patterns in sequences of interest to them, or can use the existing functionality to build new applications or tools. The source code of the tools is being made available at http://severus.dbmi.pitt.edu/revelio/.

**4. Results and Discussion**

Data pre-processing has been carried out on all the chromosomes of the human genome. Suffix array, longest common prefix (LCP) array and rank array are computed for each of the chromosomes; (for a description of these arrays, see Refs. 1 and 55). Once computed, this pre-processed data, namely, the suffix array, LCP and rank arrays will be stored for all future pattern mining applications.

Analysis may be carried out on entire sequence or only on a selected region of the sequence; it may be carried out over the whole of the selected sequence or in moving-windows with or without overlap (Fig. 2). The analysis may also be performed iteratively on different window sizes for different resolutions of analysis. For instance,

![Fig. 2. Regions of the analysis: Analysis may be carried out over the entire sequence or over a sub-region, from start to end. Also, it may be carried out over the entire sequence together, or in moving windows, one window at a time.](image-url)
Chromosome X in the build 36 is 145 MB. Analyzing this sequence at 1 kB interval would produce too much output data, making further analysis cumbersome. The user may choose to analyze the sequence in larger window sizes, thereby producing lesser output, and then rerunning the computation on selected regions using a smaller window size. We demonstrate the application of BLMT for explorative patternomics by applying it to specific human chromosomes.

### 4.1. Scalability

Data pre-processing is carried out once for every genome as it enables fast computation of subsequent pattern mining applications. Currently, suffix array computation is in linear time with the implementation of the algorithm developed by Ko and Aluru\(^6\) — the largest chromosome (245 MB) was computed in 1.5 h (this is a one-time computation and the genome is stored in this “pre-processed” format for all future pattern mining applications). The pre-processing is an investment that makes it feasible to find all the palindromes, regular expressions, etc for even such large data. As noted in literature review, without this pre-processing, pattern computation does not scale to even a fraction of these data sizes.

Subsequent to the one-time computation of suffix arrays, all pattern mining tools take only a few seconds for even for the largest chromosome (see Table 1). Most tools, such as palindromes and Alu repeats, may be computed individually in the different chromosomes. The pattern that is meaningful to be recomputed on the entire human genome is that of “exact repeats,” as this would provide new results on copies across chromosomes. The exact repeats tool when run on the chromosomes 1–22 put together (totaling 2.6 GB in length), took 29 min to list out every single sequence copy that is larger than 275 nucleotides. Copies as large as 200,000 bases were also found from one chromosome to another.

### 4.2. Perplexity analysis

The feature layer tool which facilitates explorative analysis within one genome and comparative analysis across multiple genomes is that which computes perplexity in the genome sequence. In English-text processing, perplexity is a measure of “how many distinct words can occur in the \(n\)th position given the \((n - 1)\) words preceding it,” computed as an average over the text. When applied to fragments of genome sequences, it can reveal if the fragment contains repetitive patterns, or whether the sequence has random distribution of the four nucleotides. For a uniform random sequence in which every nucleotide is independent of its previous nucleotide, the perplexity would be the same as the number of unique nucleotides in the sequence, namely four. In sequences that have many repeats of a specific pattern the perplexity would be relatively low. For example, in a repetitive sequence such as CAGCAG-CAGCAG, given one or more letters in the sequence, the next letter can be only one of the four possible nucleotides. A formal definition of perplexity as employed in the toolkit is described in supplementary material.
### Table 1. Chromosomes, patterns and time of computation.

<table>
<thead>
<tr>
<th>Chromosome</th>
<th>Size actual (MB)</th>
<th>Size available (MB)</th>
<th>% in sequence</th>
<th>CPG time (s)</th>
<th>Palindrome location</th>
<th>Palindrome length</th>
<th>Palindrome content</th>
<th>Large exact repeats</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>245.2</td>
<td>218.7</td>
<td>4.56</td>
<td>147</td>
<td>128</td>
<td>214.5</td>
<td>6.21</td>
<td>55</td>
</tr>
<tr>
<td>2</td>
<td>243.3</td>
<td>237.0</td>
<td>0.97</td>
<td>52</td>
<td>60</td>
<td>90.8</td>
<td>0.02</td>
<td>56</td>
</tr>
<tr>
<td>3</td>
<td>199.4</td>
<td>193.6</td>
<td>3.59</td>
<td>104</td>
<td>618</td>
<td>79.7</td>
<td>6.98</td>
<td>48</td>
</tr>
<tr>
<td>4</td>
<td>191.6</td>
<td>186.6</td>
<td>3.18</td>
<td>95</td>
<td>164</td>
<td>27.1</td>
<td>7.47</td>
<td>44</td>
</tr>
<tr>
<td>5</td>
<td>181.0</td>
<td>177.5</td>
<td>3.46</td>
<td>147</td>
<td>156</td>
<td>153.1</td>
<td>6.95</td>
<td>42</td>
</tr>
<tr>
<td>6</td>
<td>170.7</td>
<td>166.9</td>
<td>3.69</td>
<td>138</td>
<td>128</td>
<td>165.3</td>
<td>6.78</td>
<td>40</td>
</tr>
<tr>
<td>7</td>
<td>158.4</td>
<td>154.5</td>
<td>1.08</td>
<td>134</td>
<td>144</td>
<td>40.5</td>
<td>6.72</td>
<td>36</td>
</tr>
<tr>
<td>8</td>
<td>145.9</td>
<td>141.7</td>
<td>2.51</td>
<td>64</td>
<td>130</td>
<td>9.6</td>
<td>6.81</td>
<td>33</td>
</tr>
<tr>
<td>9</td>
<td>134.5</td>
<td>115.2</td>
<td>4.22</td>
<td>93</td>
<td>124</td>
<td>89.8</td>
<td>6.35</td>
<td>28</td>
</tr>
<tr>
<td>10</td>
<td>135.5</td>
<td>130.7</td>
<td>4.21</td>
<td>59</td>
<td>176</td>
<td>94.2</td>
<td>6.33</td>
<td>27</td>
</tr>
<tr>
<td>11</td>
<td>135.0</td>
<td>130.7</td>
<td>3.83</td>
<td>61</td>
<td>160</td>
<td>127</td>
<td>6.4</td>
<td>22</td>
</tr>
<tr>
<td>12</td>
<td>133.5</td>
<td>129.3</td>
<td>4.5</td>
<td>55</td>
<td>124</td>
<td>33.7</td>
<td>6.58</td>
<td>18</td>
</tr>
<tr>
<td>13</td>
<td>114.2</td>
<td>95.5</td>
<td>1.67</td>
<td>44</td>
<td>140</td>
<td>72.2</td>
<td>7.44</td>
<td>12</td>
</tr>
<tr>
<td>14</td>
<td>105.3</td>
<td>87.2</td>
<td>2.94</td>
<td>37</td>
<td>126</td>
<td>61</td>
<td>6.59</td>
<td>16</td>
</tr>
<tr>
<td>15</td>
<td>109.1</td>
<td>81.1</td>
<td>2.73</td>
<td>34</td>
<td>144</td>
<td>61.8</td>
<td>6.12</td>
<td>16</td>
</tr>
<tr>
<td>16</td>
<td>90.0</td>
<td>79.9</td>
<td>5.81</td>
<td>43</td>
<td>172</td>
<td>8.5</td>
<td>5.88</td>
<td>16</td>
</tr>
<tr>
<td>17</td>
<td>81.7</td>
<td>77.5</td>
<td>7.21</td>
<td>39</td>
<td>128</td>
<td>43.3</td>
<td>5.73</td>
<td>19</td>
</tr>
<tr>
<td>18</td>
<td>77.8</td>
<td>74.5</td>
<td>3.4</td>
<td>26</td>
<td>120</td>
<td>48.9</td>
<td>6.83</td>
<td>17</td>
</tr>
<tr>
<td>19</td>
<td>63.8</td>
<td>55.8</td>
<td>9.1</td>
<td>23</td>
<td>504</td>
<td>15.4</td>
<td>5.91</td>
<td>13</td>
</tr>
<tr>
<td>20</td>
<td>63.6</td>
<td>59.4</td>
<td>4.49</td>
<td>21</td>
<td>476</td>
<td>48.4</td>
<td>5.72</td>
<td>13</td>
</tr>
<tr>
<td>21</td>
<td>47.0</td>
<td>33.9</td>
<td>3.63</td>
<td>12</td>
<td>46.1</td>
<td>9.5</td>
<td>6.74</td>
<td>8</td>
</tr>
<tr>
<td>22</td>
<td>49.5</td>
<td>34.4</td>
<td>6.61</td>
<td>23</td>
<td>44.1</td>
<td>168</td>
<td>5.19</td>
<td>8</td>
</tr>
<tr>
<td>X</td>
<td>152.6</td>
<td>147.7</td>
<td>7.36</td>
<td>122</td>
<td>174</td>
<td>140.1</td>
<td>0.02</td>
<td>47</td>
</tr>
<tr>
<td>Y</td>
<td>51.0</td>
<td>22.8</td>
<td>3.19</td>
<td>16</td>
<td>106</td>
<td>7.5</td>
<td>6.82</td>
<td>7</td>
</tr>
<tr>
<td>MT</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
<td>0</td>
</tr>
</tbody>
</table>

**Note:** For Chromosomes 1-22, X, Y and MT, the following are shown: Size: Lengths of chromosomes in megabytes. The actual length of the chromosome and the total length of the contigs that have been sequenced in the reference build are shown. Alu elements: % sequence covered by Alu elements and total time to compute all the Alu elements in the chromosome. CpG islands: % sequence covered by CpG islands, total time to compute all the CpG islands in the chromosome. Palindromes: Length of longest palindrome, location of longest palindrome, % content of the sequence covered by all the palindromes, total time taken to compute all the palindromes in that chromosome. Exact Repeats: length of longest exact duplication, locations of the occurrence of the two copies (Loc 1 and Loc 2), % sequence covered by all the exact repeats larger than 21 nucleotides, total time taken to compute all these repeats in the chromosome.
4.2.1. Finding regions that contain repetitive patterns

The average perplexity in fixed-length windows of sequences is computed. Windows with low perplexity indicate presence of repeats in the sequence. Using this perplexity tool we demonstrate how a chromosome may be analyzed iteratively at increasing levels of resolution. Figure 3(a) shows the perplexity variations (for \( n = 1 \) to \( n = 8 \)) in Chromosome 19, in windows of 5 MB. The window which showed low perplexity value (marked by a purple rectangle in A) is chosen and analyzed in greater detail at regions of 250 KB. Again the particular window of interest in B is chosen to be further analyzed in detail at a resolution of \( (c) \) 10 KB windows and then \( (d) \) at a still lower resolution of 1 KB. In each panel \((A,B,C,D)\) the eight rows correspond to analyses with \( n = 1, 2, \ldots, 8 \).

![Fig. 3. Exploring the genome using perplexity measure. Features may be computed over the entire genome sequence or over a sub-region “start to end.” Analysis may be carried out over the entire region of interest together, or it may be carried out in moving windows, one window at a time. (a) Chromosome 19 perplexity is analyzed in windows of 5 MB. (b) The 5 MB window of interest (marked in a purple rectangle in A) is chosen and analyzed in greater detail at regions of 250 KB. (c) Again the particular window of interest in B is chosen to be further analyzed in detail at a resolution of (c) 10 KB windows and then (d) at a still lower resolution of 1 KB. In each panel (A,B,C,D) the eight rows correspond to analyses with \( n = 1, 2, \ldots, 8 \).](image)

4.2.2. Finding how similar or dissimilar two sequences are in terms of the sequence composition

The average perplexity of generating a sequence (test sequence) based on the n-gram model of another sequence (reference sequence) would tell whether the two are
similar to each other in terms of the nucleotide composition. Typically, the average perplexity of test sequence is larger if the test sequence is dissimilar to the training sequence. On the other hand, if the perplexities match, especially for \( n > 1 \), it means that there is a better overlap in their composition.

To demonstrate the application of this tool, we employed it to compare different bacterial proteomes that affect animals and plants. Perplexity models, also called biological language models\(^{57}\), have been computed for 27 animal-affecting bacterial proteomes, and are tested against 3 animal-affecting bacterial proteome models and 3 plant-affecting bacterial proteome models. In Fig. 4, the 27 test proteomes are arranged along the X-axis, and along y-axis are shown the cross-perplexities with the reference proteomes. As expected, animal pathogenic proteomes are more similar to each other (therefore, their cross perplexities (red lines) are usually less than 4, whereas when the 27 animal pathogens are tested against plant-pathogen models (blue lines), the perplexity is higher. Note that a perplexity of value larger than 4 can occur only when computing cross-perplexity, and not self-perplexity. When an n-gram that is very rare in the reference sequence occurs frequently in the test sequence, a cross-perplexity larger than 4 can occur. For a few pathogens, unexpected behaviour is noticed (see the region highlighted in green on x-axis). These genomes happen to be of strains of *Brucella suis*, which as the interesting result points out, is found to have played a significant role in the divergence of animal and plant affecting microbes.\(^{58}\) Creating the n-gram and perplexity based language models of proteomes sequences allows such comparisons, and more importantly provides a channel for new discoveries. In another study, we compared 970 microbial proteomes and found that the cross perplexity of these organisms is proportional to the evolutionary distance between the reference and test organisms.\(^{6}\)
4.3. **N-gram counts**

An n-gram is a sequence of n nucleotides (or amino acids). The *n-grams counts* tool in the feature layer computes which n-grams occur in a sequence, and how many times they occur.

It provides flexibility of choosing the *n* of the n-gram, and the flexibility of displaying all or only the most frequent n-grams. While the previous tool computes perplexity to identify regions of interest in chromosomes, *n-gram* counts may be run on those regions to identify the characteristics of those selected sequences.

**Composition of A, C, G, T and AT and CG:** The n-gram counts tool also outputs the following by-product information: Percentage of each of the four nucleotides A, C, G, T, and AT and CG in the entire sequence or a window of the sequence. It also computes the percent composition of the same within the n-grams, which is useful for analyses involving large numeric value of *n*.

This tool can address questions of these types:

- List all n-grams and their counts in a data set (of gene sequences) or in a chromosome or in a region of a chromosome. This in turn may be used to compute distinguishing n-grams between two different data sets.
- What are the top most frequent n-grams in a sequence (analyzed overall or over windows).
- For large *n* (such as *n* = 21), this tool may be used to find if a sequence of 21 residues repeats many times in a small window, which would indicate the location of SINEs and LINEs.

4.4. **Search N-grams**

This tool is used to search for specific N-grams in a given sequence. Given an N-gram say ‘ACCGT’ it can find all its occurrences in the sequence. It can also work with windowed option.

4.5. **Large repeats**

This tool finds all exact repeats in the chromosome larger than a user-given length. As this computation is very efficient (as it uses the pre-processed suffix arrays and auxiliary data structures in the foundation layer) and requires only a couple of seconds to compute, all large repeats in the chromosomes may be easily found. The tool outputs not only the exact sequence and its length but also the two or more positions where it has been duplicated. Figure 5 shows the large repeats in Chromosome 19. We could observe very large repeats of nearly 3500 nucleotide length at around 22 MB location and 40 MB location.

4.6. **Regular expressions**

This tool may be used to locate short sequences that conform to regular expressions. The tool allows enumeration of n-grams that conform to a specific pattern, such as
“XYabc??cbaXY”, where X and Y are specific amino acids, a to j are wildcards that allow describing a pattern, and a general wild card ‘?’ that matches with any of the 4 nucleotides. For example, the regular expression “Xab?baX” retrieves the counts of all the patterns shown in column 1 of Table 2. This tool may be applied, for example, to detect splice site regions.
As with the other tools, this tool also may be run on the entire sequence or on a selected sequence, with or without the windowing option.

4.7. Applications

Using the output of foundation and/or features layer various pattern mining applications may be built. Some are demonstrated here.

4.7.1. DNA Palindromes

This application uses the foundation layer data structures to find all the occurrences of DNA palindromes. A DNA palindrome is like English palindromes mom or ere, but the letters equidistant from the center are complements of each other (instead of being identical to each other). Functionally, this implies that this DNA fragment is capable of forming an intra-strand hairpin structure.

This application may be run with the window option, but it is more meaningful to compute it on entire chromosome at a time. A minimum length of the palindrome and maximum mismatches on the two sides may be defined. Figure 6 shows all the palindromes occurring in chromosome 19. The tool was able to compute all palindromes longer than 10 bases in chromosome 19 in 18 seconds. The figure also shows an interesting observation that a palindrome as long as 500 bases occurs at 45 MB location in chromosome 19.

4.7.2. Splice sites

Splice site regions are critical for splicing — the process of removing the introns and joining the exons together to form mRNA. Regular expression searches are often used

<table>
<thead>
<tr>
<th>Table 2. Patterns generated with regular expression Xab?baX.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Broad regular expressions</strong></td>
</tr>
<tr>
<td><strong>given by user</strong></td>
</tr>
<tr>
<td>Aac?cAa</td>
</tr>
<tr>
<td>Aag?gAa</td>
</tr>
<tr>
<td>Aat?Taa</td>
</tr>
<tr>
<td>Cac?cAc</td>
</tr>
<tr>
<td>Cag?gAc</td>
</tr>
<tr>
<td>Cat?TAC</td>
</tr>
<tr>
<td>Gac?cAg</td>
</tr>
<tr>
<td>Gag?gAg</td>
</tr>
<tr>
<td>Gat?Tag</td>
</tr>
<tr>
<td>Tac?Cat</td>
</tr>
<tr>
<td>Tag?Gat</td>
</tr>
<tr>
<td>Tat?Tat</td>
</tr>
<tr>
<td>Tat?Tat</td>
</tr>
</tbody>
</table>
in finding splice sites; an example regular expression is AGGT?AGT to identify 5’ splice site. All sequences matching this pattern are returned by the Regular Expression tool. Figure 7 shows the distribution of the regular expression patterns in chromosome 19 which was computed by the tool.

It may be seen from Fig. 8 that splice site is generally found to be located within gene regions, however, there are a few instances where this pattern is also seen occurring in non-gene regions. The first location of AGGT?AGT occurs before the
starting position of the first gene. This is also seen at 48.9 MB, where 3 occurrences of AGGT?AGT are in a non-gene region.

4.7.3. CpG Islands

CpG islands are regions of high CG nucleotide content. CpG islands are determined using multiple runs of the windowed version of the Search n-grams tool. The n-gram to be searched is CG. The Search n-grams tool gives CG concentration in the various windows. It has been run on windows of 500 base pair length. The regions where CG covers more than 50% of the content are labeled as CpG islands. CpG islands are usually found flanking genes and are responsible for protecting cytosines in CG dinucleotides from being methylated. Cytosines that are methylated cause down-regulation of transcription and gene silencing. In Fig. 8, the majority of the CpG islands are shown to be closely flanking the genes.

4.7.4. ALUs

This tool finds all occurrences of Alu elements in the given sequence. Alu elements are sequences of length around 300 base pairs and are characterized by a poly-A tail and patterns AGCT and A^5TACA^6. They have a C+G content greater than 50%. ALUs can be identified using the features we provide. The application calls the Search N-grams feature which can locate occurrences of the patterns A^5TACA^6, AGCT and poly A-tail in that order and in the vicinity of each other. C+G content is determined by calling the Search N-grams to computing the nucleotide content as by-product. The sequences matching the patterns and having high C+G content are identified by this application. This has been implemented as a standalone application. Figure 9 shows the distribution of Alu elements in chromosome 19, computed using this tool.
4.7.5. Explorative analysis

Here we demonstrate how the various feature computation tools may be used to carry out explorative analysis on genome sequences: First, perplexity in windows of 1 MB each is computed on the human Chromosome X (Fig. 10(a)).

It is observed that the perplexity is low around the 50th MB of the sequence. This region corresponds to the centromere of the X chromosome. These windows where the perplexity was low were selected and analyzed with *N-gram Counts* tool in feature layer to identify the 20 most-frequent 8-grams in this region. Ten of these most-frequent 8-grams have been found to be most frequent (Fig. 10(c)) compared to the remaining 8-grams. The *N-gram Counts* tool was run with a windowed option to

![Fig. 10](image_url)

**Fig. 10.** Analysis of X-chromosome: The X-axis shows the position in the chromosome. (a) The perplexity across the X-chromosome in windows of 1 MB. (b) The distribution of these N-grams across the X-chromosome. (c) The top 10 N-grams in the region around 50 MB.
find the counts of these specific 8-grams over the other regions (windows) in the X chromosome. This led to a very interesting observation. It was found that the most abundant ten 8-grams in the centromere region are very sparsely represented in other parts of the genome (Fig. 10(b)). This observation leads to other interesting questions.

- Do the centromere regions of all chromosomes contain large number of repetitive elements?
- Are characteristic n-grams found in centromeres of all chromosomes or is this a feature peculiar to the X-chromosome?
- Are these abundant n-grams comprising the centromere region same across all the chromosomes?

5. Conclusion

N-gram analysis and pattern mining of genetic data are applied frequently in various research methods pertaining to genetics and genomics. Examples are n-gram based promoter prediction,\textsuperscript{59} genomic island determination using n-gram analysis\textsuperscript{12} and genome signature identification.\textsuperscript{60} It requires considerable investment of time and effort to apply state-of-the-art string matching techniques and develop robust pattern mining applications. This is a big bottleneck for researchers to apply n-gram analysis and pattern mining on large-scale genomic data. BLMT provides a fast, scalable and versatile framework for pattern mining. The foundation layer preprocesses the sequence into efficient data structures, which enables extraction of features (n-grams, perplexity, palindromes, etc.) robust and scalable. The enhanced suffix array of the sequence as well as the features can be used by researchers to build custom pattern mining applications on large-scale data easily. Hence, the infrastructure we provide should help researchers from reinventing the wheel and enable them develop fast and scalable applications to suit their needs at minimal time and effort.

\textit{BLMT versin 3.0} is a valuable resource to analyze genomes in contexts such as that of the Next Gen Sequence and Personal Genomes Project. The suite is not only for pattern mining in individual genomes, but also a method to compare multiple genomes. We plan to create an Open Repository for the data structures and patterns of all the genomes, to be uploaded as and when computed, making them available for interpretation and discovery by the genomics community.

6. Methods

6.1. Materials

Chromosomes of the human genome from the reference assembly build 36 version 3 were obtained from the NCBI FTP site (ftp://ftp.ncbi.nih.gov/genomes/H_sapiens/). Only Chromosome 19 and X are discussed in this paper. This build of chromosomes are not yet complete, and contain gaps. Chromosome sequences have been
downloaded in FASTA format. Each file contains sequences of all the contigs of a given chromosome in multiple sequence FASTA format.

### 6.2. Implementation details

Each of the tools processes the suffix array, longest common prefix array and rank array (the three arrays together are similar to enhanced suffix array), to achieve fast and scalable computation of the features. The detail of implementation of each of the tools is described in supplementary material.

### 7. Authors’ Contributions

MKG developed most of the tools. TPM developed perplexity computation tools and windowing capabilities. ADM computed most of the patterns. KM developed capability to extend suffix array computation to large genomes by merging smaller suffix arrays. SA co-developed the code for linear time implementation of suffix array construction with MKG.

### Acknowledgments

MKG would like to thank her thesis advisors Judith Klein-Seetharaman and Raj Reddy for ideas, discussions, support and training during her Ph.D., which formed the core of n-gram based work for genome sequence analysis and led to the development of the biological language modeling toolkit, a precursor to the work presented here. This work was supported mostly from the start-up funds awarded to MKG by Department of Biomedical Informatics of University of Pittsburgh (Pittsburgh, PA, USA), whereas SA has been supported by the University of Pittsburgh Arts and Science Fellowship and KM supported by Supercomputer Education and Research Centre of Indian Institute of Science (Bangalore, India). The authors would like to acknowledge that the computational resources for this work were provided by PittGrid at University of Pittsburgh (www.pittgrid.pitt.edu).

### Additional files

**Additional File 1—Implementation details**

This file describes the implementation details of each of the tools.

### References


Madhavi K. Ganapathiraju is an Assistant Professor at Department of Biomedical Informatics, University of Pittsburgh School of Medicine, Pittsburgh, USA. She completed her Ph.D. in Language and Information Technologies from Carnegie Mellon University School of Computer Science, Pittsburgh, USA, and M.Engg in Electrical Communications Engineering from Indian Institute of Science, Bangalore, India. Her doctoral thesis was on applying language processing algorithms such as statistical n-gram modeling and latent semantic analysis to biological sequences, to answer questions pertaining to protein sequence and structure. These methods have subsequently been applied widely in the domain of computational biology. Currently, her research focus is on computationally discovering protein–protein interactions in the human interactome.

Asia D. Mitchell has a B.S. in Biochemistry from Chatham University, Pittsburgh, USA, as well as a Certificate in Biomedical Informatics from the University of Pittsburgh School of Medicine, Pittsburgh, USA. She is currently a Ph.D. student, studying Medical and Molecular Genetics at Oregon Health and Science University, Portland, Oregon. Her research interests include genomics of complex disease, bioinformatics and signaling disruption in cancer.

Mohamed Thahir has an M.S. in Intelligent Systems from University of Pittsburgh, Pittsburgh, USA, and B.E. degree in Computer Science and Engineering from Anna University, India. He worked as a Software Engineer in Trilogy E-Business
Software in Bangalore, India for two years. He is currently a Software Development Engineer at Amazon.com. His research interests include Bioinformatics, Machine Learning and Information Extraction.

**Kamiya Motwani** received M.S degree in Computer Science in 2011 from University of Wisconsin Madison, USA, and B.E. (Hons.) from Birla Institute of Technology and Science, India. She worked as a Summer Research Fellow at Supercomputer Education and Research Centre, Indian Institute of Science, Bangalore. Currently, she is working as a software developer at Oracle America Inc., CA. Her research interests include machine learning and computer vision.

**Seshan Ananthasubramanian** has an M.S. in Intelligent Systems from University of Pittsburgh, Pittsburgh, USA. He has a B.E. degree in Computer Engineering from Maharashtra Institute of Technology, Pune. His research interests include machine learning.