

# **The Hunt for Genomic Dark Matter: Aligning Non-coding Functional DNA**



Naila Mimouni  
LSI DTC  
University of Oxford

Gerton Lunter and Jotun Hein  
Department of Statistics  
University of Oxford

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Motivation . . . . .	2
1.1.1	Alignment Algorithms . . . . .	2
1.1.2	Assessing Performance of Alignment Algorithms . . . . .	3
1.2	Aims of the Work . . . . .	4
<b>2</b>	<b>Biological Background</b>	<b>6</b>
2.1	Mutations in the Genome . . . . .	6
2.1.1	Origins of Mutations . . . . .	6
2.1.2	Consequences of Mutations . . . . .	7
2.1.3	Location and Frequency of Mutations . . . . .	8
<b>3</b>	<b>Overview and Assumptions of the Model</b>	<b>9</b>
3.1	The Number of Indels . . . . .	10
3.2	The Distribution of Ungap Lengths . . . . .	10
3.2.1	What are Ungaps . . . . .	10
3.2.2	Ungaps and Alignment Error . . . . .	10
3.2.3	Distribution of Ungap Lengths . . . . .	12
3.3	Gap Attraction . . . . .	13
3.4	Confidence Intervals . . . . .	13
3.5	The Methodology . . . . .	14
3.5.1	The Data . . . . .	14
3.5.2	The Algorithms . . . . .	17
3.5.3	Gap Attraction . . . . .	19
<b>4</b>	<b>The Results</b>	<b>21</b>
4.1	Verification of the Hypothesis . . . . .	21
4.1.1	The Genome . . . . .	21
4.1.2	Blastz . . . . .	22
4.1.3	Clustalw . . . . .	23
4.2	Confidence Intervals . . . . .	24
4.2.1	The Genome . . . . .	24
4.2.2	Blastz . . . . .	24
4.2.3	Clustalw . . . . .	24
4.3	Comparison between Blastz and Clustalw . . . . .	25
4.4	Gap Attraction . . . . .	25

*CONTENTS*

<b>5</b>	<b>Discussion and Future Work</b>	<b>35</b>
5.1	Weighted Linear Regression . . . . .	36
5.2	Other Alignment Algorithms . . . . .	36
5.3	Pair-HMM Alignment . . . . .	37
5.4	Data Simulation . . . . .	37
5.5	Estimating Functional DNA . . . . .	37
<b>A</b>	<b>The Python Database</b>	<b>41</b>

# List of Figures

2.1	A graphical representation of slippage; the mispairing of the complementary DNA strands during replication, resulting in mutation (adapted from (Strachan and Read, 2004)). Slippage involves a region of non-pairing (shown as a bubble) containing one or more repeats of the newly synthesised strand (backward slippage), or of the parental stand (forward slippage), causing, respectively, an insertion or a deletion in the newly synthesised strand. . . . .	7
3.1	Indels "raining down" uniformly on the genome. Each arrow represents an indel event. . . . .	10
3.2	An alignment with the resulting gap and ungap regions. An indel results in a gap in the alignment. Given an alignment, an ungap is the region between two neighbouring gaps which were introduced by the alignment algorithm. . . . .	11
3.3	A graphical representation of the alignment error measure. The ungap lengths are plotted against their log frequencies. The best linear fit is shown in red, and the 95% confidence intervals for the histogram counts assuming the true distribution shown in green. Here the model is accurate between 12-80, as the data points lie within the 95% confidence intervals . . . . .	12
3.4	An example of the alignment format produced the blastz algorithm (alignment 18518 of human chromosome 19), including the summary line . . . . .	15
3.5	An illustration of the different gap options for the Gap Attraction measure. Each gap option represents the number of nucleotides subtracted from the overall number of nucleotides. In the first option, the first and last chunks are discarded because the alignment stops, and there is no information regarding ungaps before the first chunk or after the last chunk. This includes alignments where no ungaps occur. The second option subtracts the number of nucleotides in the first gap and last gap. Finally, the third option subtracts half the number of nucleotides in the first and last gap. . . . .	20
4.1	The distribution of ungaps for the whole human genome (including genes and repeats) . . . . .	27

*LIST OF FIGURES*

4.2	The distribution of ungaps for the whole human genome with the different gene and repeat options . . . . .	28
4.3	Blastz: The distribution of ungaps for chromosome 21 for the different gene and repeat options . . . . .	29
4.4	Clustalw: The distribution of ungaps for chromosome 21 . . . . .	30
4.5	The ungap Vs log frequency (dotted black) with the linear regression (red) and 95% the confidence intervals (green) shown for the whole human genome. . . . .	31
4.6	Blastz: The ungap Vs log frequency (black dotted) with the linear regression (red) and the 95% confidence intervals (green) shown for chr 21. . . . .	32
4.7	Clustalw: The ungap Vs log frequency (black dotted) with the linear regression (red) and the 95% confidence intervals (green) shown for chr 21. . . . .	33
4.8	A graphical comparison of the Gap Attraction measure for Blastz and Clustalw . . . . .	34

# List of Tables

3.1	A summary of the genome data. The number of alignments (as shown in figure 3.4) for each chromosome and the size of the file in bytes . . .	16
3.2	The scoring matrix for blastz . . . . .	17
3.3	The DNA identity matrix for clustalw . . . . .	18
4.1	A summary of the ungap lengths and their corresponding frequencies for the blastz alignment of the human genome and the mouse genome for the +gene + repeat option. . . . .	22
4.2	Gap Attraction measure for Blastz and Clustalw and the ratio between them for the different gene and repeat options. . . . .	26
A.1	The list of available DV views , with a short description of usage . . .	42

*LIST OF TABLES*

**”It is an order of magnitude easier to design two good algorithms than to tell which one is better”**

**- Maxim**

# Abstract

**Motivation:** Statistical analysis of conservation patterns in human and mouse genomes suggests that as much as 5% of human genomic DNA is under purifying selection. Known genes account for 1.5%. Identifying and characterising the remaining 3.5% is still a major challenge.

With the availability of more genomes, an intuitive way to detect conservation patterns is to align the human genome with genomes of other species such as mouse or rat.

Compared to coding DNA, aligning non-coding DNA is a hard task because of the relatively high sequence divergence and the absence of codon structure allowing a higher incidence of insertions and deletions. So far, it has been difficult to assess or compare the performance of alignment algorithms because no suitable gold standards and no evaluation procedures have been proposed.

**Results:** We propose an objective measure of the alignment accuracy of non-coding DNA called "Gap Attraction". "Gap Attraction" gives a measure of the proportion of ungaps, the conserved regions between two random neighbouring indel events, that have been misaligned. The "Gap Attraction" measure is derived from a model that assumes that insertions and deletions (indels) rain on the genome independently of each other and uniformly along the sequence. From that, it follows that the length of ungaps is geometrically distributed. This hypothesis is verified by the data for the alignment of human chromosome 21, and then the whole human genome, against the mouse genome. The histogram counts for ungaps of medium length lie within the 95% confidence intervals confirming the hypothesis.

"Gap Attraction" does not require knowledge of the true alignment. We measured the "Gap Attraction" index for two widely-used alignment algorithms; Blastz which was developed specifically for aligning human-mouse DNA, and Clustalw; a global alignment algorithm for pairwise protein and DNA sequences. As expected, Blastz performs better than Clustalw according to our evaluation measure.

**Contacts:** [naila.mimouni@bnc.ox.ac.uk](mailto:naila.mimouni@bnc.ox.ac.uk), [lunter@stats.ox.ac.uk](mailto:lunter@stats.ox.ac.uk)

# Chapter 1

## Introduction

### 1.1 Motivation

Statistical analysis of conservation patterns in human and mouse genomes suggests that as much as 5% of human genomic DNA is under purifying selection (Mouse Genome Sequencing Consortium, 2002). Known genes account for 1.5%. Identifying and characterising the remaining 3.5% is still a major challenge.

With the availability of more genomes, an intuitive way to detect conservation patterns is to align the human genome with genomes of other species such as mouse or rat. In the case of coding DNA, evaluation of alignment is conducted against existing gold standards such as protein structures and experimentally verified transcripts.

Non-coding DNA is harder to align than coding DNA, because of the relatively high sequence divergence and the absence of codon structure, allowing a higher incidence of insertions and deletions compared to coding areas. Moreover, it has been difficult to assess or compare the performance of alignment algorithms so far because no suitable gold standards are available and no evaluation procedures have been proposed. Therefore, there is a need for an objective measure of alignment algorithm performance.

#### 1.1.1 Alignment Algorithms

Pairwise alignment between two or more DNA or amino acid sequences is a comparison that aims at finding evolutionary related sites by looking for common structural information or conserved regions, sometimes with the aim of inferring function (although this is a hard task). Alignment algorithms usually take into account only the most common genetic mutations: insertions, deletions and substitutions (explained in the next chapter), and perform a local or global alignment.

Alignment algorithms are used to compare the similarity of sequences of different lengths, allowing gaps. Global alignment presumes that the entire sequences are related. The most widely used algorithm for obtaining optimal global alignment in biology is the Needleman-Wunsch (Needleman and Wunsch, 1970), a more efficient version was described by Gotoh (1982). This algorithm is used by clustalw for slow pairwise alignment, and will be explained in chapter 3.

Local alignment, on the other hand, is useful to compare subsequences of two or more sequences. It is usually used to identify a known domain/sequence motif in a new protein, or identify local similarities between a long DNA sequence and a shorter one or between two highly divergent sequences. The most widely known local alignment algorithm in biology is the Smith-Waterman (Smith and Waterman, 1981).

The time complexity of both these algorithms is  $O(mn)$  for two sequences of lengths  $m, n$ . In practice, this is too slow to search large database, or align multiple sequences. This is why, there has been attempts to implement faster algorithms than straight dynamic programming. These usually incorporate heuristics that sacrifice some sensitivity.

### Alignment Score

The goal of any alignment algorithm is to find, or sometimes approximate, the optimal alignment. However, given the output of an alignment algorithm, it is very hard to judge how accurate it is. Most alignment algorithms produce a score along with the alignment. This score takes into account the basic mutational processes of residue substitution, insertions and deletions. The sum of terms for each aligned pair of residues, plus terms for each gap make up the total score. In probabilistic terms, this corresponds to the likelihood that the sequences are related, compared to being unrelated (Durbin et al., 1998).

For a single algorithm, the significance of scoring is not that informative. Aligning a long sequence with a short one with very good accuracy could lead to the same score as aligning two long sequences with low accuracy. In this case, using score alone, it is very hard to tell which alignment is more correct. Additionally, it is usually the case that for a single algorithm, the higher the score the better the alignment. This is not valid when comparing two different alignment algorithms, as they usually have different scoring methods. The next section presents the currently-available methods of alignment evaluation.

## 1.1.2 Assessing Performance of Alignment Algorithms

### Coding DNA

**Structural Alignment** A way of evaluating alignment of coding DNA is to translate it into protein, and then check a 'standard-of-truth' database, and superimpose the corresponding structures.

Structural alignment suffers from a number of drawbacks. There is a large discrepancy between the number of known protein sequences and that of solved three-dimensional structures. Also, despite the number of available databases (de Bakker et al., 2001; Gerstein and Levitt, 1998; Holm and Sander, 1993, 1999; Mallika et al., 2002; Marti-Renom et al., 2001; Mizuguchi et al., 1998; Shindyalov and Bourne, 2001; Thompson et al., 2001), they often do not focus on the quality of structural alignment, and rely on sequence alignment algorithms which they are trying to evaluate in the first place (Venclovas, 2003). Finally, structures are deformable, which would lead to inaccurate alignment (Marsden and Abagyan, 2004).

**Experimentally Verified Transcripts** An alternative approach is to benchmark against a thoroughly filtered, biologically validated dataset of sequences. Examples include the HMR195 mammalian sequence (Rojic et al., 2001) and the *adh* region for *Drosophila* (Ashburner et al., 1999). While this approach would reflect the biology, identifying such an experimentally validated dataset usually requires a substantial investment in time and resources, and is only limited to certain regions of the genomes which might not be representative.

### Non-coding DNA

Simulating alignments over a range of divergence times estimated from the genus *Drosophila* has been used for benchmarking non-coding DNA (Pollard et al., 2004). However, to our knowledge, no objective evaluation measure for non-coding DNA has been proposed in the literature as yet.

## 1.2 Aims of the Work

The last section pointed at the fact that evaluation measures are usually targeted towards coding DNA. Aligning non-coding DNA is a hard task to achieve, and evaluating the resulting alignment is even harder. In the field of algorithmics, "it is an order of magnitude easier to design two good programs than to tell which one is better" (Schwartz et al., 2003). The aim of this work is to propose an objective evaluation of alignment of non-coding DNA, and then eventually to build an alignment algorithm given the evaluation measure.

The goal of this work is the hardest according to the maxim above; to produce an objective quality measure for non-coding DNA alignment called "Gap Attraction". Gap Attraction is a measure of the sparsity of short *ungaps*, which we define as ungapped aligned regions lying between two consecutive indels. We call it "gap attraction" because it represents the amount two consecutive gaps, separated by a short ungap, attract each other, pulling the ungap in the middle towards one of the two sides, and resulting in a single longer gap. This is indicative of alignment error.

The model is based on the assumption that insertions and deletions (indels) rain down on the genome uniformly along the sequence and independently of each other. While the first assumption does not hold for 5% of the genome that is under selection, the second one holds to a large extent for all genomic areas. From that, it follows that the distribution of the length of ungap is geometric. The model was verified on the blastw alignment of the whole human genome and chromosome 21 with the mouse genome, as well as the clustalw alignment of chromosome 21 with the mouse genome. The histogram counts for ungap of medium length (15-80) lie within the 95% confidence intervals as expected if the distribution is indeed geometric. Longer ungap (more than 80 nucleotides) do not fit the model. These are conserved regions with a lower rate of indels due to selection which is not taken into account by the model. Shorter ungap (0-15 nucleotides) do not fit the model, their frequency is lower than expected. While there is no evidence discarding a biological mechanism acting on short conserved regions, it is certainly very unlikely. Assuming there is not such a mechanism, the short

ungaps not fitting the model represent those not aligned properly by the alignment algorithm. "Gap Attraction" measures the proportion of misaligned ungaps under our model, one aspect of the errors made by alignment algorithms. It is by no means a measure of all the errors made by alignment algorithms. Rather, it provides a way of classifying and comparing algorithms, a task so far not achieved.

Once "gap attraction" is clearly understood, and justified through simulated data, it will be used towards designing our own alignment algorithm. Given our evaluation measure, we are planning on implementing a pair-HMM alignment algorithm which should perform better than existing algorithms. As a first step, the parameters will be extracted from the blastz alignments, as blastz was developed for the task of aligning non-coding functional DNA, and has a lower gap attraction measure than clustalw. Eventually, the optimal parameters for insertion and deletion rates (those that minimise gap attraction) will be inferred from the model, and the resulting alignment algorithm could be used to help the identification of conserved regions between human and other genomes. This will hopefully result in characterising functional non-coding DNA.

Chapter 2 describes the biological concepts underlying the mathematical model. Chapter 3 explains the statistical and computational parts; the assumptions and implications of the proposed model, as well as the implementation methodology and the data used. The results are shown in chapter 4. Finally, a discussion of the model, methodology, results, and future work are presented in chapter 5.

## Chapter 2

# Biological Background

Evolution of the genome is controlled by two forces: *mutation* and *selection*. Mutations are the changes proposed to DNA, and selection is accepting or rejecting these changes. Mutations are important for several reasons; they may have deleterious or (rarely) advantageous consequences to an organism or its descendants, they are the most common way to study an organism through making a variant (mutant) lacking the ability to perform the task under scrutiny, and related to this work, they are the major source of genetic variation which fuels evolutionary change. This is why genomic comparison, including alignment, relies on modelling and estimating mutation.

### 2.1 Mutations in the Genome

The genome is not a static entity, rather it is subject to different types of heritable changes called mutations. Large-scale chromosome abnormalities involve loss or gain of chromosomes, or breakage and rejoining of chromatids. We are not concerned with these in this project. Instead, smaller scale mutations can be grouped into different classes according to their effect on the DNA sequence as follows (Strachan and Read, 2004, chap. 11):

**Base substitutions** involve replacement of usually a single base, and are sometimes referred to as point mutations. These are of two types: transitions and transversions. Transitions are from purine to purine ( $A \leftrightarrow G$ ) or pyrimidine to pyrimidine ( $C \leftrightarrow T$ ) and transversions are from purine to pyrimidine or pyrimidine to purine.

**Insertions** one or more nucleotides are inserted into a sequence.

**Deletion** one or more nucleotides are eliminated from a sequence. Insertions and deletions are collectively referred to as indels.

#### 2.1.1 Origins of Mutations

Some mutations are due to the integration of segments of DNA from elsewhere in the genome. Often, these exogenous segments of DNA have specifically evolved the

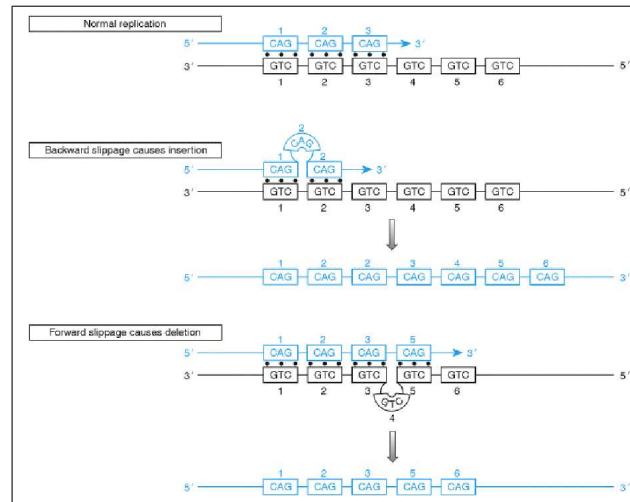


Figure 2.1: A graphical representation of slippage; the mispairing of the complementary DNA strands during replication, resulting in mutation (adapted from (Strachan and Read, 2004)). Slippage involves a region of nonpairing (shown as a bubble) containing one or more repeats of the newly synthesised strand (backward slippage), or of the parental stand (forward slippage), causing, respectively, an insertion or a deletion in the newly synthesised strand.

capability to integrate elsewhere in the genome, and hence are called *mobile elements* or *transposable elements* (Griffiths et al., 2002, chap. 10).

The molecular mechanisms underlying mutations include *mutagens*. Mutagens are physical or chemical agents, present in the internal or external environment, which induce mutations or increase their chance of occurring. Mutagens include: nitrite which converts C to U, intercalating agent; a chemical that resembles a base pair of DNA and can insert between two base pairs, and UV radiation and X-rays.

Spontaneous mutations are those that arise without a known mutagen. They are usually the result of spontaneous errors in DNA replication and repair. Figure 2.1 (adapted from (Strachan and Read, 2004)) illustrates slippage; the mispairing of the complementary DNA strands during replication, resulting in mutation. Slippage involves a region of nonpairing (shown as a bubble) containing one or more repeats of the newly synthesised strand (backward slippage), or of the parental stand (forward slippage). Backward slippage causes an insertion, while forward slippage causes a deletion in the newly synthesised strand.

### 2.1.2 Consequences of Mutations

The consequences of base substitutions in protein coding regions of a gene depend on the type of substitution and its location. They may be silent; *synonymous mutations*, in which case they do not result in a new amino acid in the protein sequence, or cause amino acid change; *missense mutation*. Missense mutations may have very serious consequences, as in the case of sickle-cell anaemia, mild consequences such as a substitution in position 6 of beta-globin, or no phenotype such as two known amino

acid substitutions at position 7 of beta-globin. Finally, substitutions in a protein coding region may mutate an amino acid codon to a termination codon. This results in a prematurely shortened protein, and is referred to as a *nonsense mutation*, the effects of which are variable depending on how much of the truncated protein is present and required for its function. Substitutions may also occur in promoters or regulatory regions of genes, and may affect their transcription, translation, or splicing. Many of the beta-thalassemias are the result of these types of non-structural mutations that affect the level of expression of the globin genes.

Insertions and deletions of one or more (not in multiples of three) nucleotides in the coding region of a gene result in *frameshift mutations*. Such a deleterious gene mutation in a cell leads to the death of that single cell. However, sometimes the mutation can lead to abnormal continuation of cell division, causing cancer (Strachan and Read, 2004, chap. 11).

### 2.1.3 Location and Frequency of Mutations

During the average human lifetime in the order of  $10^{17}$  cell divisions are estimated to take place. As each cell division requires the incorporation of  $6 \times 10^9$  new nucleotides, error-free DNA replication would require a repair mechanism that would ensure the correct nucleotide is inserted on each of the  $6 \times 10^{26}$  occasions. Throughout a human lifetime, each gene will be a locus for about  $10^8 - 10^{10}$  mutations (but for any one gene, only a tiny minority of cells will carry a mutation).

Because the substitution of alleles in a population takes thousands or even millions of years, its rate can only be estimated from comparison to other DNA molecules that share a common ancestor, such as orthologs in different species. Most of the human mutation rates estimated are obtained by comparison to the mouse genome (Mouse Genome Sequencing Consortium, 2002). Collins and Jukes (1994) found out that the transition rate exceeded the transversion rate; 1.4:1 and 2:1 for substitutions not resulting and resulting in an amino acid change respectively. Transitions may be favoured over transversion because they lead to a more conserved protein sequence.

Estimating divergence from the draft human (International Human Genome Sequencing Consortium, 2001) and mouse (Mouse Genome Sequencing Consortium, 2002) genomes indicated that humans undergo less substitutions per site (0.17) compared to 0.34 for mice. Assuming the human-mouse split occurred 75 Million years ago, the average substitution rates would have been  $2.2 \times 10^{-9}$  and  $4.5 \times 10^{-9}$  for human and mouse respectively. These are the *average* rates since the human-mouse divergence and the current substitution rate per year in the mouse genome is thought to be much higher. The same comparison allowed estimation of indels of less than 50 nucleotides long. Both species showed a net loss, 2:1 to 3:1 range for deleted vs inserted, but the overall loss was at least twice higher in the mouse.

## Chapter 3

# Overview and Assumptions of the Model

This chapter explains the assumptions underlying our model. It also highlights the results and the measure of alignment error expected, should the data fit the model.

Chapter 1 stressed that there is no evaluation measure for neutrally evolving DNA alignments. In this chapter, we show that under plausible hypotheses, the gold standard alignment will have certain statistical properties. The extent to which the actual alignment differs from these properties is a measure of alignment error.

The first assumption made by the model is that indels on both genomes occur *uniformly* along the sequence. In other words, we hypothesise that there is no position or sequence content bias and no selection placed on human and mouse genomes. There is evidently selection constraints on functional regions, accounting for 5% of the human genome (Mouse Genome Sequencing Consortium, 2002). In this study, we filter out annotated genes, which account for 1.5% of the genome. Within the remaining data which consists of 98.5% of the genome, the non-coding functional 3.5% is still included, but the large majority (96.5%) is likely to be under no selection constraints at all.

The second assumption is that indels occur *independently* of other indels. This means that the probability of an indel occurring does not depend on any other probability. This assumption holds very accurately, even under selection constraints. Assuming the opposite, i.e. dependence, would mean that an indel event is dependent on another indel event that happened millions of years prior. While there is no evidence discarding this, it is certainly very unlikely. Note that independence does not require that indel rates stay constant over time.

From a modelling perspective, our hypothesis is a simplifying one, which allows for investigating the underlying mechanisms, otherwise impossible to model. However, as will be seen further down in the document, the data for human chromosome 21 and the whole human genome fits the resulting model for medium length ungnaps (around 15-80 nucleotides). Since alignment error and a small amount of DNA under selection



Figure 3.1: Indels "raining down" uniformly on the genome. Each arrow represents an indel event.

plausibly explain these deviations on respectively the small and large ungap lengths, we conclude that the simplified description we propose is not too far from the real evolutionary process.

### 3.1 The Number of Indels

The assumptions are summarised here. Indels occur:

- Uniformly along the genome. That is, the probability of an indel occurring in any part of the genome is the same. This assumption is clearly debatable, as there is certainly selection bias for functional regions.
- Independently of other indels.

These assumptions imply that indels are distributed according to a Poisson process, with the number of indels in a region of fixed size given by a Poisson random variable. Figure 3.1 is a graphical representation of indels (shown as arrows) "raining down" along a sequence. This model is reminiscent of that representing the number of misprints on a page, the number of requests a WWW server receives in a day, or the number of  $\alpha$ -particles discharged in a fixed time from some radioactive material.

Given that the average number of indels in an interval is  $\lambda$ , the probability  $P$  of finding  $x$  indels in the interval is:

$$P(X = x) = \frac{\lambda^x e^{-\lambda}}{x!}$$

To note is, for the purpose of this project, measuring the number of indels is not as relevant as the fact that they are represented as a Poisson process.

### 3.2 The Distribution of Ungap Lengths

#### 3.2.1 What are Ungaps

Biologically, ungaps are the maximal genomic regions in two species that are homologous and have undergone no insertion or deletion events since their most recent ancestor. An indel results in a gap in the alignment. A correct alignment should identify ungaps as the ungapped aligned regions between two neighbouring gaps introduced by the alignment algorithm. Figure 3.2 shows an alignment of two algorithms, with the resulting gap and ungap regions.

#### 3.2.2 Ungaps and Alignment Error

As was pointed out in section 1.1.1, alignment algorithms maximise their score in the aim of producing an optimal, or suboptimal alignment. The score is somewhat

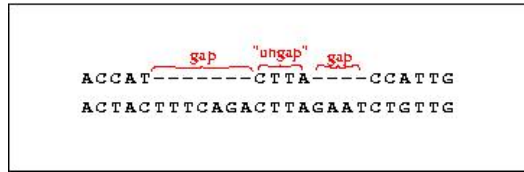
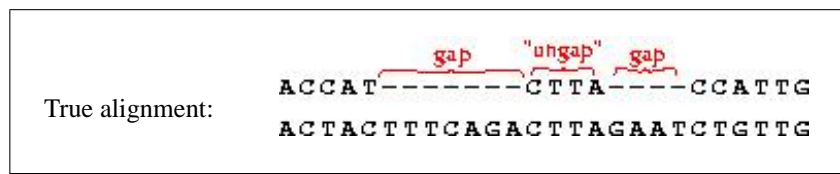


Figure 3.2: An alignment with the resulting gap and ungap regions. An indel results in a gap in the alignment. Given an alignment, an ungap is the region between two neighbouring gaps which were introduced by the alignment algorithm.

heavily penalised when a gap is introduced, and less penalised when a gap is extended. This was incorporated in alignment because it has been observed that in sequence evolution, insertions and deletions very often affect several nucleotides at a time. It is not a characteristic that could be excluded from alignment algorithms. For short ungaps for example, there is too little information to decide whether it is better to open a new gap or extend the current one, this feature helps formalise this decision in terms of score.

An implication is that algorithms usually avoid introducing a new gap in favour of extending an existing gap. In other words, a single long indel is more likely, and therefore cheaper, than two short ones. This results in joining the ungap (in the true alignment) to an ungapped stretch on either side.

Suppose the true alignment is the one pictured below:



A consequence of the alignment scoring system would join the ungap (in red) to the ungapped region on the left. This results in a single longer gap rather than two short gaps, as pictured below:



A results of this is the sparsity of short observed ungaps. However, the better the alignment procedure is at recovering the true alignment, the less the sparsity of short ungaps becomes. We propose to use this proportion as a measure of alignment error.

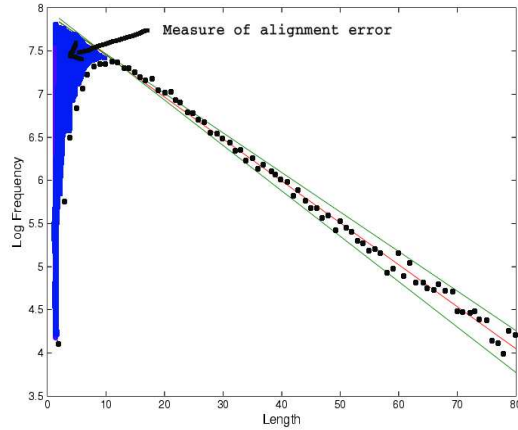


Figure 3.3: A graphical representation of the alignment error measure. The ungap lengths are plotted against their log frequencies. The best linear fit is shown in red, and the 95% confidence intervals for the histogram counts assuming the true distribution shown in green. Here the model is accurate between 12-80, as the data points lie within the 95% confidence intervals

### 3.2.3 Distribution of Ungap Lengths

To be able to measure how many ungap are being moved and thereby aligned incorrectly as indicated above, it is imperative to model the expected true distribution of ungap lengths. From the Poisson distribution of indels, it follows that the distribution of ungap lengths is geometric. Therefore, given the probabilities  $p$  of an ungap per site, and  $1 - p$  of opening a gap per site, the probability of an ungap of length  $L$  is given by:

$$P(L) = (1 - p)p^{L-1}$$

Given independent indel events, each having a probability  $1 - p$  of being a success, the geometric distribution measures the probability that after  $L$  trials, the first  $L - 1$  are failures, and the last trial is a success. In our model, the success is the occurrence of an indel, and the failures are the ungap. By analogy, the probability of an ungap of length  $L$  is given by having a gap after an ungap of length  $L - 1$ .

The geometric model implies a straight line in logarithmic coordinates. Figure 3.3 shows the expected result from plotting the ungap lengths between 0 and 80 nucleotides against their log frequencies (dotted). The red line represents the best linear fit. The 95% confidence intervals where 95% of the data should lie if it fits the model are shown in green. The region highlighted corresponds to ungap lengths whose frequency is smaller than expected, which corresponds to those ungap that have been joined to ungapged regions due to the "gap attraction" phenomenon (as explained above). It is the graphical representation of the measure of the alignment error we propose, called "Gap Attraction".

### 3.3 Gap Attraction

The gap attraction measure is the surface of the region highlighted in blue in figure 3.3 divided by the total number of nucleotides  $S$  (depending on the gene, and repeat options). The observed frequencies on the graph  $C(i)$  represent the frequencies of those alignments that have been aligned correctly by the algorithms, while the y-coordinates on the line (obtained by linear regression)  $\log freq(i)$  represents the frequency of alignments that should have been aligned correctly (according to our hypothesis). Subtracting the former from the latter results in those alignments that have not been aligned correctly by the algorithms according to our model. Accordingly, if  $n$  is the first ungap length whose log frequency lies on the best linear fit line, and  $C(i)$  is the observed count for ungap length  $i$ , then for each ungap length  $i \leq n$ , we calculate the difference between the estimated counts and the observed counts and multiply it by the ungap length. The estimated counts are the exponential of the log frequencies (giving the true frequency). The total area is the sum of all these values divided by the total number of nucleotides  $S$ . It is given by:

$$\frac{\sum_i^n [(est(i) - C(i)) * i]}{S} \quad \text{where} \quad est(i) = exp(logfreq(i))$$

### 3.4 Confidence Intervals

If our model were to be reflective of the data, what are the 95% confidence intervals in which the count of the number of ungap of length  $L$  lies?

According to our model, the count is a Bernoulli process. The probability that an ungap has a certain given length  $L$  is some fixed probability  $q$ , and each successive ungap has a length that is independent of the others. From section 3.2.3:  $q = P(L) = p(p - 1)^{L-1}$ . If  $c$  is the count of ungap of length  $L$ , then the probability to find precisely  $n$  ungap of length  $L$  is:

$$P(c = n) = \binom{N}{n} q^n (1 - q)^{N-n}$$

where  $N$  is the total number of ungap, and  $q$  is the probability of seeing an ungap of length  $L$ . From that, the expected value of the count  $c$  is

$$E(c) = Nq$$

and the variance of count  $c$  is:

$$Var(c) = Nq(1 - q) \approx Nq \quad \text{for small } q$$

Estimating  $q$  from experimental data given the observed count  $C$  gives  $q \approx C/N$ .

The variance of  $c$  given the estimate of  $q$  is:  $Var(c) \approx Nq \approx NC/N \approx C$ .

The standard deviation ( $sd$ ) given the estimate of the variance  $sd \approx \sqrt{C}$ .

The 95% confidence interval =  $[C - 2sd, C + 2sd]$ .

Using the estimated value of  $sd$ , the confidence interval =  $[C - 2\sqrt{C}, C + 2\sqrt{C}]$ .

**Plotting Confidence Intervals** The 95% confidence intervals are :

$$C \pm 2\sqrt{C} = C\left(1 \pm \frac{2}{\sqrt{C}}\right)$$

We need to plot the log, therefore:

$$\log C \pm \frac{2}{\sqrt{C}} = \log C + \log\left(1 \pm \frac{2}{\sqrt{C}}\right) \quad (3.1)$$

Using first order Taylor series:  $\log\left(1 \pm \frac{2}{\sqrt{C}}\right) \approx \frac{2}{\sqrt{C}}$ .

Replacing this approximation in (3.1) gives:

$$\text{The log confidence intervals: } \log C \pm \frac{2}{\sqrt{C}} \quad (3.2)$$

We now deal with each of the two terms in (3.2). First, according to our model:

$$C = kp^L \text{ where } k = 1 - p = \text{the probability of opening a gap per site} \quad (3.3)$$

Log (3.3):

$$\log C = k' + L \log p \text{ where } k' = \log k \quad (3.4)$$

Second:

$$\frac{2}{\sqrt{C}} = \frac{2}{\sqrt{kp^L}} = \frac{r}{p^{L/2}} \text{ where } r = \frac{2}{\sqrt{k}} \quad (3.5)$$

Replacing in (3.2) gives:

$$\text{The log confidence intervals: } k' + L \log p \pm \frac{r}{p^{L/2}}$$

For  $p \approx 1$ , this is more or less linear in  $L$ .

Figure 3.3 shows the two 95% confidence intervals in green. Here the model is accurate between 12-80, as most of the counts lie within the confidence intervals.

## 3.5 The Methodology

The previous section highlighted the model and the results to expect under the assumptions made. This section explains the methodology of the work including the data and algorithms, which are available from the author on request.

### 3.5.1 The Data

#### The Genome

The genome data was downloaded from the ucsc website (<http://www.genome.ucsc.edu/>). It contains the whole June 2003 human assembly (also known as build 34) aligned against the October 2003 mouse assembly (also known as mm4 or NCBI Mouse Build 32). The alignments were obtained using the blastz algorithms (workings and evaluation presented in the next section). Blastz is available from Webb Miller's group at Pennsylvania State University ([www.bx.psu.edu/miller\\_lab/](http://www.bx.psu.edu/miller_lab/)). The alignments are in

```

18518 chr19 50930292 50930752 chr7 121386812 121387235 - 12369
AAGGTTTAGCTGTTTAAATTTGGGTCTTTGGGCCCATAAATTTGTTGGGGCTAG
GGACTCTTATTTTGGGATCTCTGGGTGCTCA--AAGGAGCATCATAGGGAAG
ATGAAGGGTGCCCATCAGGGTTCCGCCCATCCTTCCCTGCAATCCCAGAGGT
CACTAATCATTCTTTTTCCCTCTTTGCAGATGTTTGGTTCCAGCAGATACTT
GGGTTCTCTGAACAGCCCAGAGCTAATTCACTGGGTCCAGTGACAGGTAA
GGGCTTGCCCATCTCCCTTATAGCCCCAGCCACTCTCATTGAATaataata
acaaaaacat-ttgctgagcatttaccgtttgctagactctttgccaaacat
tttgtttgta--tgataactgttaatccatacaacaatctcatgagggaagt-
--tgtgtacaaagatgcctaccatthccagatgagagtactctgaggTGATC

AGAGTTTAGGGTTTTAATTTAGGTTCTGAACCCATA-TTTATG-----
----TCCTATTTTAGGAACTCTAGGTGCTTTTTAAATGTACCATGGGGGTA
GGGAGGTTCGTACATTA----CTTGGCTGATTCACTTTGGAGTCCCAAAGGTC
ACTAATAATT-TTTTTCCCTCTTTGCAGATGTTGGGTTCCAGACAAATCTCA
TGTTTCTCTGATCAGCACAGAGCTAGATCTCCAAGTCTACTGACAGGTAAT
TGTT--CTCTACCTTCAGTGCAGCTCTACCCACTC----TGAACAATAGTAG
CAACAAGGTGTTACTAAAC----ACTGTTT-CTAGACTGTTTGTAGTCCT
TCCATTGTATCTAATAGCTAT-----TACAACCACCAGACAAGGAAGTAG
GTGCGTATA-----GTCCTTCATTTCCA----GAGACGGTTCTGAGGTGATTC

```

Figure 3.4: An example of the alignment format produced the blastz algorithm (alignment 18518 of human chromosome 19), including the summary line

the 'axt' format. Each alignment contains a summary line, the human sequence and the mouse sequence with gaps, and is separated from the next alignment by a space. The summary line contains information about the alignment number, the base numbers on the human and mouse chromosomes and the score. Figure 3.4 shows an example of the alignment format produced by the blastz algorithm, including the summary line. Table 3.1 summarises the genome data, and for each chromosome gives the number of alignments (as shown in figure 3.4) and the size of the file in bytes.

## Chromosome 21

**Description** Chromosome 21 is the smallest human autosome, and accounts for 1-1.5% of the human genome. This makes it practical to analyse, while retaining its representative quality. A recent, high quality sequence covering 99.7% of the 33.5 Megabases of the long arm has been produced. The sequencing has revealed a low gene density (127 known genes, 98 predicted genes, and 59 pseudogenes) compared to other chromosomes of similar size, such as chromosome 22 (545 genes) (Hattori et al., 2000). Additionally, from the human chr21-mouse alignment, it has been found that a large fraction (about 40%) of the conserved elements on chromosome 21 do not encode for known genes, and do not have a known function (Frazer et al., 2001; Mural et al., 2002). This is an important observation when studying conserved non-coding DNA.

**Blastz and Clustalw Data for Chr 21** For the blastz alignment, the data was obtained in a similar way to the whole genome above, i.e. downloaded from the ucsc website (<http://www.genome.ucsc.edu/>). This time, only the human chromosome 21

<i>chromosome</i>	<i>Num. of alignments</i>	<i>size of file (bytes)</i>
Chr 1	151562	205947699
Chr 2	159268	219493213
Chr 3	129436	182892151
Chr 4	110342	152989116
Chr 5	111367	158887519
Chr 6	112489	149310640
Chr 7	98008	132169700
Chr 8	88414	119901736
Chr 9	75274	104840956
Chr 10	90586	119213569
Chr 11	83058	117577692
Chr 12	85749	109539174
Chr 13	57672	78806600
Chr 14	60899	82383669
Chr 15	58690	77434675
Chr 16	58246	71910480
Chr 17	61732	75778889
Chr 18	45285	61884700
Chr 19	31212	30075716
Chr 20	39708	55761987
Chr 21	20051	26136796
Chr 22	21435	26754865

Table 3.1: A summary of the genome data. The number of alignments (as shown in figure 3.4) for each chromosome and the size of the file in bytes

(hg16) is aligned against the mouse genome (see table 3.1 for the number of alignments).

On the other hand, a clustalw alignment of human and mouse genomes is not available, therefore we had to build it ourselves in the following steps:

1. We were only interested in the final local alignment, i.e. the detailed placement of the gaps introduced, not the global alignment (which stretch of the mouse genome corresponds to which stretch of the human genome). For this reason, I used the blastz data, and wrote a Perl script that removes the gaps for each of the blastz alignments. This resulted in non-aligned ungapped stretches of human and mouse genomes for clustalw to align.
2. I installed clustalw, and wrote a Java script to divide the alignment file (20051 alignments) into 20051 files each containing a single alignment to be aligned. Converting the blastz summary line format into one that is accepted by clustalw was also performed. Please note that clustalw does not accept lowercase notation for repeats, so a Java program keeping track of the repeats was also written.
3. Running the clustalw on each of the 20051 alignment files.
4. Converting the results of the clustalw alignments into the axt format (summary line etc), and introducing the repeats where appropriate.

Once all of these steps were performed, we had a clustalw alignment file containing the 20051 alignments in an axt format ready to be processed by our algorithms.

	A	C	G	T
A	91	-114	-31	-123
C	-114	100	-125	-31
G	-31	-125	100	-114
T	-123	-31	-114	91

Table 3.2: The scoring matrix for blastz

### 3.5.2 The Algorithms

#### Blastz

The blastz algorithm, used by the PipMaker webserver (Schwartz et al., 2000), was implemented in an effort to align neutrally evolving DNA in the human and mouse genomes. It follows the three-step strategy used by Gapped Blast (Altschul et al., 1997): finding short near-exact matches, extending each short match disallowing gaps, and extending each ungapped match that exceeds a certain threshold by a dynamic programming procedure that permits gaps.

First, the initial short seeds are determined by looking for runs of 19 consecutive nucleotides in each sequence, within which the 12 positions indicated by a 1 below are identical (Ma et al., 2002).

1110100110010101111

The scoring matrix blastz uses is shown in table 3.2, with a gap open penalty of 400 and a gap extension penalty of 30. The minimum score for an alignment to be kept is 3000 for the first pass, and then 2200 for the second pass, which serves to restrict the search space to the regions between two alignments found in the first pass. Although the gap introduction penalty is high (3000), once started the alignment keeps extending as long as the average score remains positive. Finally, the axtBest program automatically selects for an alignment likely to be the orthologous one, and produces the 'axt' files (Schwartz et al., 2003).

Obtaining a measure of the performance of blastz was not possible (hence this project). Therefore, the blastz alignment evaluation consisted mainly of changing the blastz parameters (gap penalties) and observing the differences, aligning the human genome to the inverse mouse genome (0.16% aligned as opposed to the original 40%), and noting the conserved syntenicity between human chr20 and mouse chr 2 (only 3.3% of human chr 20 aligned outside human chr2) Schwartz et al. (2003).

#### ClustalW

Contrary to Blastz, clustalw was not specifically developed for human-mouse DNA alignment. It is a progressive multiple alignment profile-based algorithm, which succeeded an earlier version called clustalv (Higgins et al., 1992). It works in much the same way as the Feng-Doolittle method (Feng and Doolittle, 1987), except it is tuned for profile alignment methods.

We used the pairwise option for clustalw, which is based on the global alignment algorithm by Needleman and Wunsch (1970). The idea behind this algorithm is to find

	A	C	G	T
A	10	0	0	0
C	0	10	0	0
G	0	-0	10	0
T	0	0	0	10

Table 3.3: The DNA identity matrix for clustalw

the global optimum by recursion from optimal alignments of smaller sequences. It involves the construction of a dynamic programming matrix,  $M$ , where the optimal score is computed and stored in the bottom right cell. The optimal alignment is obtained through traceback of the matrix. The matrix score for row  $i$  and column  $j$  are computed recursively, using a scoring matrix  $s$ , and a gap penalty  $d$  as follows:

$$M(i, j) = \max \begin{cases} M(i-1, j-1) + s(x_i, y_i) & \text{residue } x_i \text{ is aligned to residue } y_i \\ M(i-1, j) - d & \text{residue } x_i \text{ is aligned to a gap} \\ M(i, j-1) - d & \text{residue } y_i \text{ is aligned to a gap} \end{cases}$$

We are not aware of any literature comparing blastz to clustalw. However, when evaluated for protein alignments, clustalw performs worse than other alignment algorithms such as Dialign (Morgenstern, 1999), Poa (Lee et al., 2002) and T-Coffee (Notredame et al., 2000). The difference between the highest scoring (Dialign/Poa depending on the domain arrangements) and clustalw is about 5%. Clustalw is the second fastest, and its speed is not affected by sequence length (Lassman and Sonnhammer, 2002).

We obtained the clustalw alignments using the default values for the full pairwise alignment. The scoring matrix was the DNA identity matrix (figure 3.3), the gap opening penalty (GAP OPEN) is 10, the gap extension penalty (GAPEXT) is 0.05, and the gap separation penalty (GAPDIST) is 8.

### DBView

This section summarises the Python database DBView that was written by Gerton Lunter. Some time was spent learning Python and getting familiar with this database to be able to use it to support the novel code.

Reading and parsing a large database of files usually takes more time and labour than the actual algorithms used on the data. This is why it is useful to cache the parsed data, especially if there is need to access it often, as is the case for the genome and chromosome 21 for this project. The caching is automatically done by DBView, which also offers different "views" of the data, namely: *DBView.py*, *AlignmentParView.py*, *AlignmentView.py*, *RepeatView.py*, *AlignTools.py*, *ChromosomeView.py*, *SegmentView.py*, *SegmentIndexView.py*, *Seg.py*, *SeqStatView.py*, *SequenceSelectView.py*, *EnsGeneView.py*, *pydb.py*, *SubsetView.py*, *RepBaseView.py*, and *Vec.py*.

For example, *EnsGeneView* is used to open a view onto the *ensGene.txt* file, which lists all genes annotated in the Ensembl database. From that, filtering can be done so

that only the genes in a single chromosome (chr 21) are accessed. The list of available views, with a short description of usage is available in the appendix.

### **Ungap Count**

This is the first Python script that was written. It takes in the blastz or clustalw file, counts the number of ungaps, and produces a file of two columns recording the length of an ungap and its corresponding frequency respectively. For each alignment in the blastz or clustalw files, a mapping of both human and mouse alignments is created, the gene, exon and repeat regions are removed/added depending on the option. This Python mapping is then processed and the ungaps lengths and frequencies are recorded in the file. For the whole genome data, a loop over all chromosomes is used.

### **Plotting**

To be able to generate a plot from python, the 'matplotlib' (<http://matplotlib.sourceforge.net/>) was used. It allows to use matlab notation within python code, and perform mathematical analysis and produce plots. The linear regression was implemented using the matlab function *polyfit*.

### **Confidence Intervals**

Calculated and plotted the 95% confidence intervals explained in section 3.2.5.

### **3.5.3 Gap Attraction**

The gap attraction measure is the surface of the region highlighted in blue in figure 3.3 divided by the total number of nucleotides (depending on the gene, and repeat options). It involves implementing the following calculations:

1. Estimate the region which represents the misaligned stretches (highlighted in blue in figure 3.3). The observed frequencies on the graph represent the frequencies of those alignments that have been aligned correctly by the algorithms, while the y-coordinates on the line (obtained by linear regression) represents the frequency of alignments that should have been aligned correctly (according to our hypothesis). Subtracting the former from the latter results in those alignments that have not been aligned correctly by the algorithms according to our model. Accordingly, for each of the lengths (x coordinates in the graph) that appear in the region, the observed frequency is subtracted from the y-coordinate on the line resulting from the regression line. For example, if for length 20 the observed frequency is 200, while the y-coordinate on the line (obtained linear regression) is 350, then the resulting value is  $(350-200)*20$ . The total surface is the sum of all the resulting values for all lengths in the region. This was done for each of the gene and repeat options, and for both algorithms (blastz and clustalw).
2. Calculating the total number of nucleotides, the total number of nucleotides excluding repeats, the total number of nucleotides excluding genes, and the total number of nucleotides excluding exons.

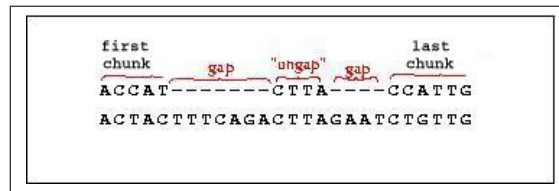


Figure 3.5: An illustration of the different gap options for the Gap Attraction measure. Each gap option represents the number of nucleotides subtracted from the overall number of nucleotides. In the first option, the first and last chunks are discarded because the alignment stops, and there is no information regarding ungaps before the first chunk or after the last chunk. This includes alignments where no ungaps occur. The second option subtracts the number of nucleotides in the first gap and last gap. Finally, the third option subtracts half the number of nucleotides in the first and last gap.

3. For each of the gene and repeat options, the gap attraction results, as will be seen below, are presented for three total nucleotide options, subtracting the following from the total number of nucleotides: the number of nucleotides in alignments that have 0 or 1 gap (this means that there are no ungaps), the number of nucleotides in the first and last gap, and the number of nucleotides in half the first gap and half the second gap. Figure 3.5 presents a graphical representation of this.

# Chapter 4

## The Results

This chapter presents the results obtained.

### 4.1 Verification of the Hypothesis

The first step was the verify the hypothesis on the data available, the whole genome first and then chromosome 21.

#### 4.1.1 The Genome

Table 4.1 gives a summary of the ungap lengths and their respective frequencies as obtained from the alignment of the human genome and the mouse genome (including genes and repeats). Figure 4.1a is a plot of the length of ungap on the x-coordinate against their respective log frequencies, for the whole genome data, and with options '+genes+repeats', that is, including genes and repeats. The plot is a geometric distribution, verifying the model. 99% of the ungap lengths lie in the region 0-3500, with log frequencies 0-4. We can also see a small number of ungap of length 4000, including a few very long ones, with low frequencies, probably representing conserved functional sites between mouse and human. This could be a step towards identifying coding as well as non-coding functional sites.

Figure 4.1b is a plot of the length of ungap on the x-coordinate against their respective log frequencies, for the whole genome data but only showing the first 80 nucleotides, and with options '+genes+repeats'. The linear regression line (in red) fits the model nearly perfectly for ungap lengths of relatively medium size (15-80). Longer ungap (more than 80 nucleotides) and shorter ungap (0-15 nucleotides) do not fit the model. The former is probably due to selection not taken into account by our model, resulting in long conserved regions. Short ungap on the other hand appear less frequently than expected. This might be explained by a biological mechanism, it is very unlikely however. After some research, we have not come across such a mechanism, and therefore we explain the sparsity of short ungap as those not aligned properly by the alignment algorithm. The black dots occurring under the line (lengths 0-15) represent the frequencies that are smaller than what is expected. The frequencies corresponding to lengths 15-25 occur above the line, which could either be an artifact of the linear re-

<i>Ungap Length</i>	<i>Frequency</i>
1-15	10078462
16-80	17179511
80-200	1175165
200-499	111623
500 - 999	6198
1000 - 1999	1055
2000-3000	136
3000-4000	20
> 4000	5

Table 4.1: A summary of the ungap lengths and their corresponding frequencies for the blastz alignment of the human genome and the mouse genome for the +gene + repeat option.

gression (which a weighted linear regression would fix), or that for these lengths the frequencies a bit higher than expected by our model. Also, deviation from the constant indel rate hypothesis (uniformity) can cause the distribution of indel lengths to become concave. This could be an explanation for the slight deviation from model for ungap lengths 15-80.

We also investigated the hypothesis under different gene and repeat options. Figure 4.2 shows the results. The '+' sign means inclusion of the gene/exon/repeat, while the '-' sign means exclusion. Part a) is a plot of the length of ungaps on the x-coordinate against their respective log frequencies, for the whole genome data, and with the different options. For all the options, the distribution is more or less geometric. The total frequency for the '+gene' option is higher than the '- exon' option which in turn is higher than '-gene', which is what we expected. The graphs do not look totally identical indicating the presence/absence of functional/non-functional sites. Part b) is a plot of the length of ungaps on the x-coordinate against their respective log frequencies, for the whole genome data but only showing the first 80 nucleotides, and with all the options. Again, the same pattern is observed, lengths 0-15 the frequencies are below the expected (characterised by red line), then lengths 15-25 are above. To note is the effect of the amount of data available, the line fits the graph better in the '+gene' and '-exon' option that it does for the '-gene' option. Finally, for all the graphs in this part, (easily observed in the '-genes+repeats' graph), when there are points that are below the lines, then the ones following them happen to be above it, in a compensation-like manner.

These results suggest that our hypothesis is more or less representative of the real evolutionary process. Next, we investigate our hypothesis for one individual chromosome and both alignment algorithms.

#### 4.1.2 Blastz

Figure 4.3a is a plot of the length of ungaps on the x-coordinate against their respective log frequencies, for the blastz alignment of chromosome 21 with the mouse genome, and with the different options. It is important to note that chromosome 21 data also fits

our hypothesis. The distribution is geometric, and patterns observed for the genome are also valid for chromosome 21. The majority of ungap lengths lie between lengths 100-350, and log frequencies 0-3. A few long ones are also observed, indicating conserved DNA.

Figure 4.3b is a plot of the length of ungap on the x-coordinate against their respective log frequencies, for the blastz alignment of chromosome 21 with the mouse genome, but only showing the first 80 nucleotides, and with the different options. Again, the same pattern is observed when repeats, genes or exons are excluded but with different log frequencies obviously. The line resulting from the linear regression is shown for the data between lengths 0-80. The linear regression line (in red) fits the model for ungap lengths of relatively medium size (12-80). Longer ungap (more than 80 nucleotides) and shorter ungap (0-12 nucleotides) do not fit the model.

While the model fits the data for the blastz alignment of chromosome 21, it is important to verify the hypothesis for other alignment algorithms, such as clustalw, that were not developed for this specific task.

### 4.1.3 Clustalw

Figure 4.4a is a plot of the length of ungap on the x-coordinate against their respective log frequencies, for the clustalw alignment of chromosome 21 with the mouse genome, and with the different options. It is important to note that the clustalw data for chromosome 21 also fits our hypothesis. The distribution is geometric, and patterns observed for the genome and blastz chr 21 are also valid for clustalw. The majority of ungap lie between lengths 100-300, and log frequencies 0-3. More ungap of length near 300 are observed than in blastz alignments. This could be due to the clustalw hard-coded heuristic that gap extension is performed as long as the score remains positive. A few long ungap are also observed, indicating conserved DNA.

Figure 4.4b is a plot of the length of ungap on the x-coordinate against their respective log frequencies, for the clustalw alignment but only showing the first 80 nucleotides, and with the different options. Again, the same pattern is observed when repeats, genes or exons are excluded but with different log frequencies. The line resulting from the linear regression fits the data between lengths 0-80, although not as precisely as in the blastz case. This could be due to the algorithm itself or to errors in computing the clustalw alignments. Moreover, as there is less data ('-genes' and '-exons'), the points lie further from the line. The linear regression line (in red) fits the model nearly perfectly for ungap lengths of relatively medium size (24-80). Longer ungap (more than 80 nucleotides) and shorter ungap (0-24 nucleotides) do not fit the model, as is the case for the blastz alignments. As is the case with blastz, longer ungap are selected for, while the sparsity of short ungap can be explained as those not aligned properly by the alignment algorithm. In this case, the first ungap length fitting our model is 24, which is about double that for blastz (12), indicating a higher error for clustalw under our model. The same observation regarding the different locations of the points above and below the line in a compensating manner applies in this case as well.

Now that we have observed that the data fits the hypothesis, it is important to check

whether it lies within the 95% confidence interval of the model for each of the whole genome, the blastz alignment and the clustalw.

## 4.2 Confidence Intervals

### 4.2.1 The Genome

Figure 4.5 shows the ungap lengths vs log frequency (dotted black) with the linear regression (red) and the 95% the confidence intervals (green), for the whole human genome and the different options. Calculating the confidence intervals was explained in section 3.4. Because of the availability of the data, the two 95% confidence intervals are very close, an indication of precision. The linear regression line can hardly be seen, except for the '-gene' option, where there is less data, and the two confidence intervals (shown in green) diverge. The important observation is that 95% of the data points lie *within* the two confidence interval lines for all the options.

The next step is to check the data and the confidence intervals for both chromosome 21 under the blastz and the clustalw alignment.

### 4.2.2 Blastz

Figure 4.6 shows the ungap lengths vs log frequency (dotted black) with the linear regression (red) and the 95% the confidence intervals (green), for the blastz alignment of chromosome 21 and the different options. This time, because there is not as much data, we can clearly see the two intervals. Still, for all the options, 95% of the data points lie *within* the two confidence intervals for all the options.

### 4.2.3 Clustalw

Figure 4.7 shows the ungap lengths vs log frequency (dotted black) with the linear regression (red) and the 95% the confidence intervals (green), for the clustalw alignment of chromosome 21 and the different options. Similarly to the blastz data, we can clearly see the two confidence interval lines, whose divergence increases with the ungap length suggesting a decrease in precision. Again, for all the options, the biggest majority of the data points (except for a few odd ones) lie *within* the two confidence interval lines for all the options.

All of the results presented above verified the hypothesis presented in the last chapter. First, plotting the ungap lengths against their frequencies showed the geometric distribution that the hypothesis predicted, for the whole genome data and then the blastz and clustalw alignments of chromosome 21. Second, the linear regression performed on the frequencies of ungap lengths between 0-80 precisely fitted the data for medium length ungap. Longer ungap are conserved regions not taken into account by our model. Short ungap, unless there is a biological mechanism, are those misaligned by the algorithm. The overall patterns observed were common to the whole genome and the blastz and clustalw alignments. Finally, calculating the confidence intervals for lengths 0-80 resulted in the observation that a large majority lies within the 95%

confidence intervals, again confirming the hypothesis. Next, we perform a comparison between blastz and clustalw which will lead to introducing the gap attraction measure of alignment error.

### 4.3 Comparison between Blastz and Clustalw

As was already pointed out, the blastz algorithm was developed for aligning non-coding functional DNA, while clustalw was not. Accordingly, it would be reasonable to expect that blastz performs better at the task at hand than clustalw. However, this has not been verified so far because of a lack of an objective measure of alignment error. Figure 4.8 shows a graphical comparison of the alignment error measure for blastz and clustalw. We call this measure: "Gap Attraction". As was indicated in section 3.5.5, the region lying below the red line represents the measure of alignment error, and therefore the "gap attraction" measure (highlighted in blue). From the plots, it is clearly visible that the gap attraction region for blastz is *smaller* than the one for clustalw (which will be confirmed by numbers below). Additionally, The first ungap length whose frequency lies on the line is 12 for blastz, and 24 for clustalw. This indicates that gap attraction is *lower* for blastz, confirming our expectation.

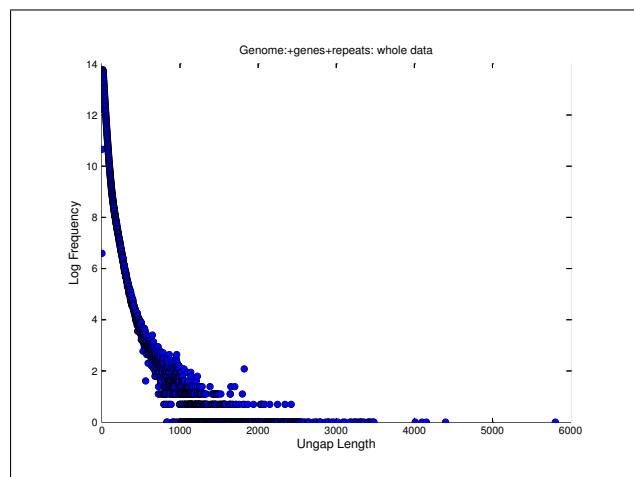
### 4.4 Gap Attraction

*Gap Attraction* is an objective measure of alignment algorithms. We call it "gap attraction" because it represents to what extent two gaps, separated by an ungap, are attracted to each other, pulling the ungap in the middle towards one of the two sides, and resulting in a single longer gap. "Gap attraction" measures the sparsity of short un-gaps. Given that we are not aware of any biological explanation for this, gap attraction is indicative of alignment error. It is by no means an exact measure of all the errors an alignment algorithm makes. Rather, it measures one of the aspects of alignment error, allowing a novel way of classifying and comparing alignment algorithms. Gap attraction is a very useful measure for evaluation when no benchmark exists, such as for aligning non-coding functional DNA, a task so far not possible to achieve.

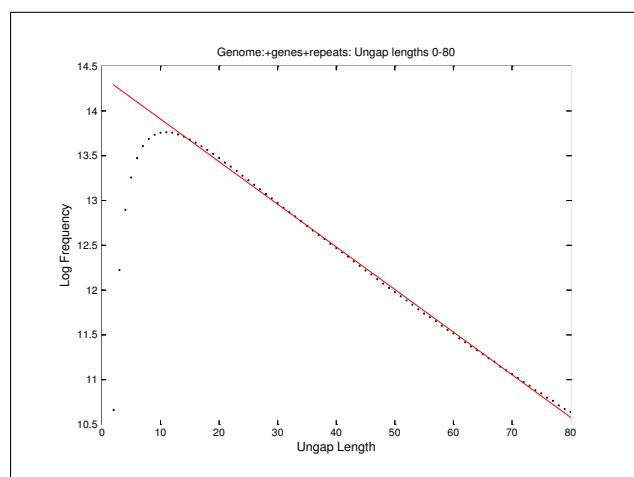
Table 4.2 presents the "gap attraction" measure for blastz (indicated by B), and clustalw (indicated by W) for the alignment of human chromosome 21 with the different options. The 'F' option is the fraction of error between blastz and clustalw. For example, if the fraction is 98% then the gap attraction of blastz constitutes 98% of that made by clustalw. The different gap options are explained in figure 3.5. The F fraction indicates that gap attraction is very close between the two algorithms (93-97%). However, in the '-repeat' case, it drops to 73-77%, this could be due to a bug in the code. Given the time constraints, it was not possible to investigate this fully.

Plot/Nuc		more than 1 gap	discard 2 gaps	discard half 2 gaps
<i>+gen+rep</i>	B:	0.8497	0.8492	0.8653
	W:	0.8671	0.9121	0.8882
	F:	<b>97.99</b>	<b>93.08</b>	<b>97.42</b>
<i>+gen-rep</i>	B:	0.6599	0.7453	0.7000
	W:	0.84599	1.008	0.9172
	F:	<b>78.00</b>	<b>73.93</b>	<b>76.31</b>
<i>-gen+rep</i>	B:	0.7819	0.7849	0.7834
	W:	0.8023	0.8064	0.8044
	F:	<b>97.46</b>	<b>97.33</b>	<b>97.39</b>
<i>-gen-rep</i>	B:	0.6194	0.6257	0.6222
	W:	0.8003	0.8108	0.8054
	F:	<b>77.43</b>	<b>77.17</b>	<b>77.25</b>
<i>-ex+rep</i>	B:	0.7820	0.7855	0.7838
	W:	0.8024	0.8076	0.8050
	F:	<b>97.45</b>	<b>97.26</b>	<b>97.36</b>
<i>-ex-rep</i>	B:	0.6192	0.6272	0.6232
	W:	0.7997	0.8149	0.8072
	F:	<b>77.42</b>	<b>76.96</b>	<b>77.19</b>

Table 4.2: Gap Attraction measure for Blastz and Clustalw and the ratio between them for the different gene and repeat options.

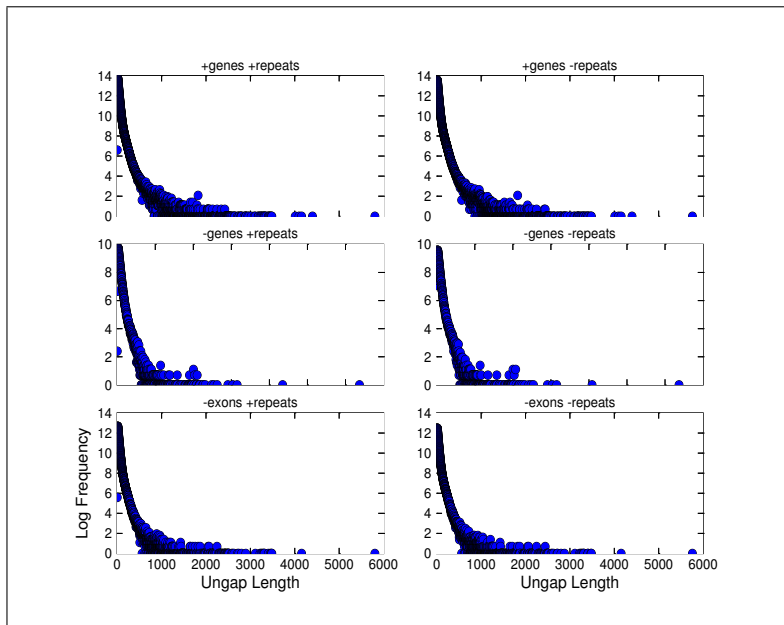


(a) The length Vs log frequency. The whole data shown

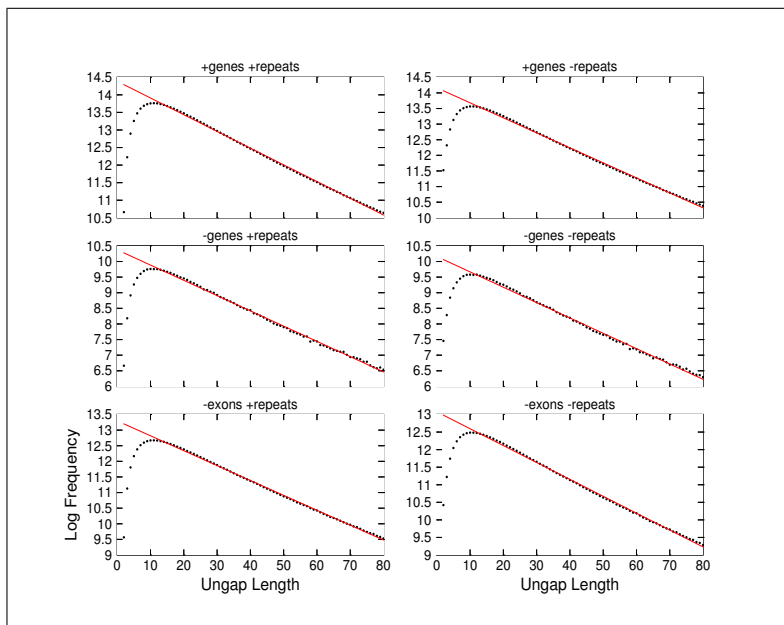


(b) The length Vs log frequency with linear regression (red line).  
Data for ungaps of length less than 80 nucleotides shown

Figure 4.1: The distribution of ungaps for the whole human genome (including genes and repeats)

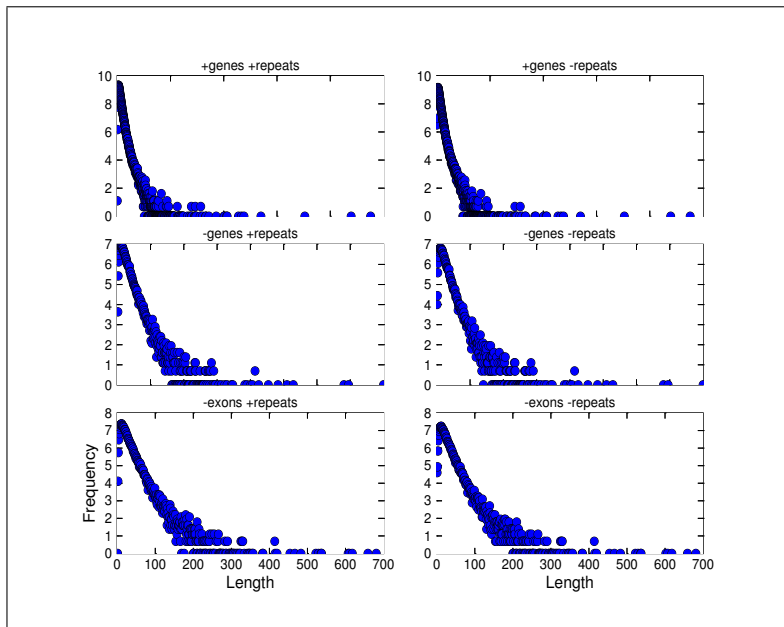


(a) The length Vs log frequency. The whole data shown.

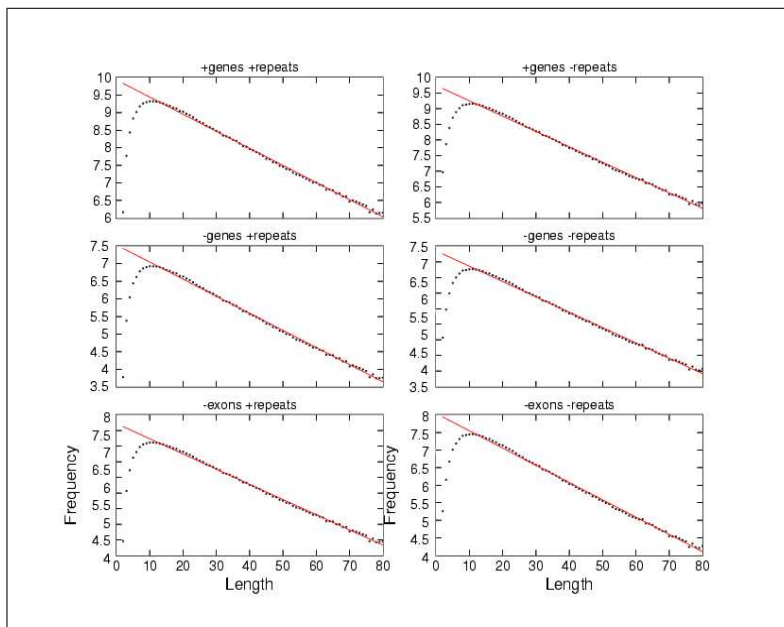


(b) The length Vs log frequency with linear regression (red line).  
Data for ungap of length less than 80 nucleotides

Figure 4.2: The distribution of ungap for the whole human genome with the different gene and repeat options

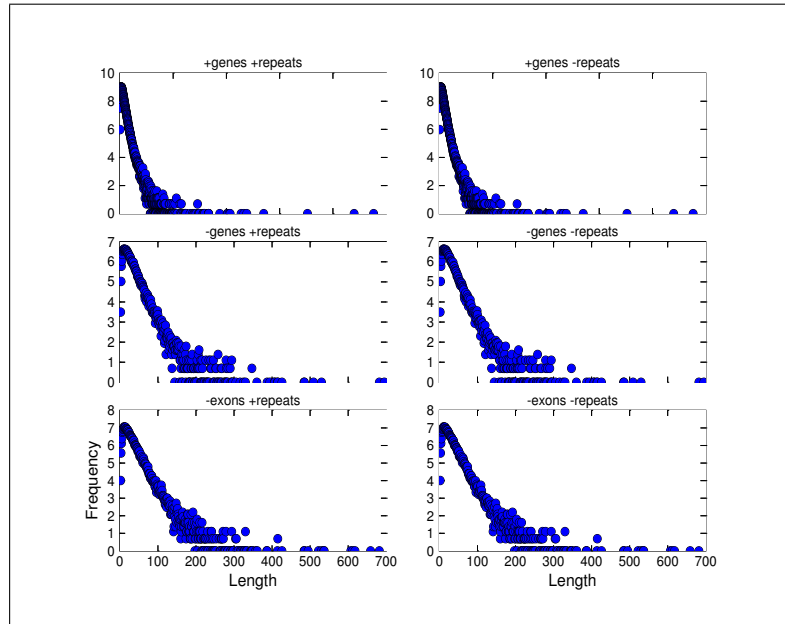


(a) The length Vs log frequency. The whole data shown

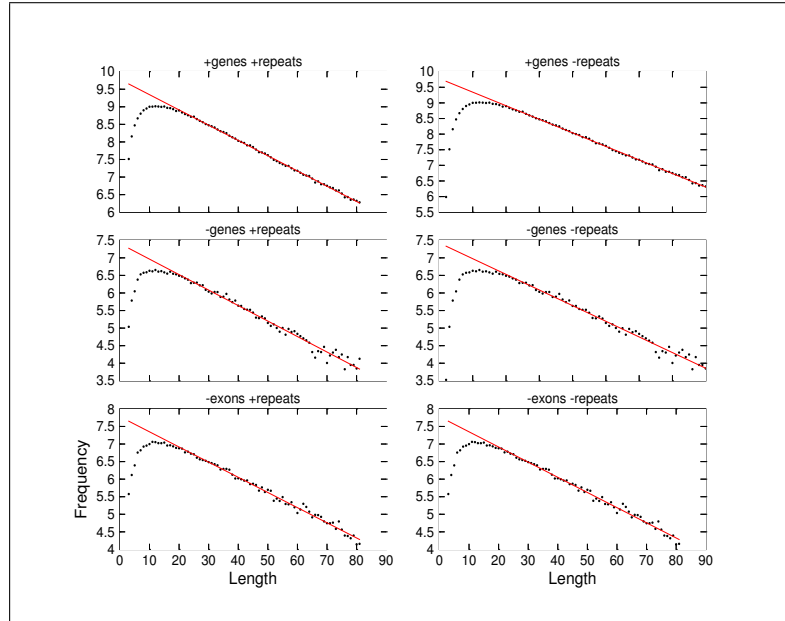


(b) The length Vs log frequency with linear regression (red line).  
Data for ungap of length less than 80 and shown

Figure 4.3: Blastz: The distribution of ungap for chromosome 21 for the different gene and repeat options



(a) The length Vs log frequency. The whole data shown for the different gene and repeat options



(b) The length Vs log frequency with linear regression (red line). Data for ungap of length less than 80 and the different gene and repeat options shown

Figure 4.4: Clustalw: The distribution of ungap for chromosome 21

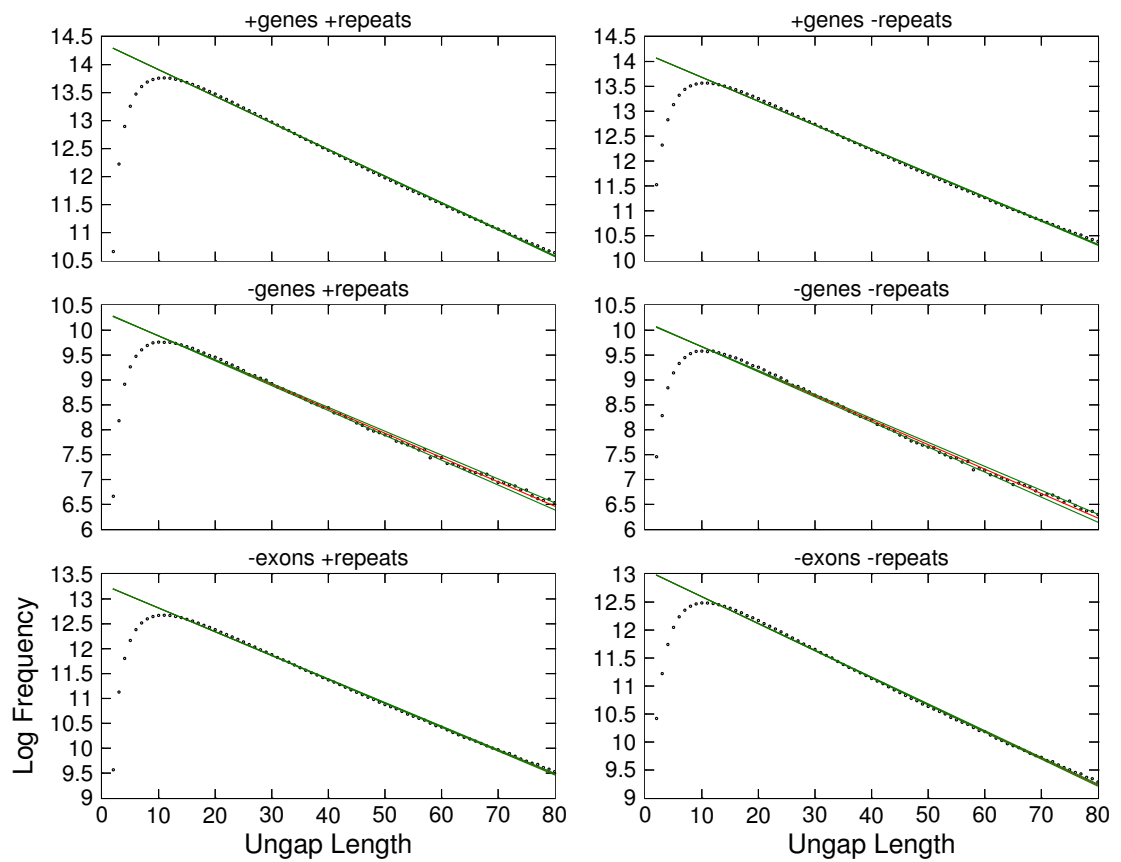


Figure 4.5: The ungap Vs log frequency (dotted black) with the linear regression (red) and 95% the confidence intervals (green) shown for the whole human genome.

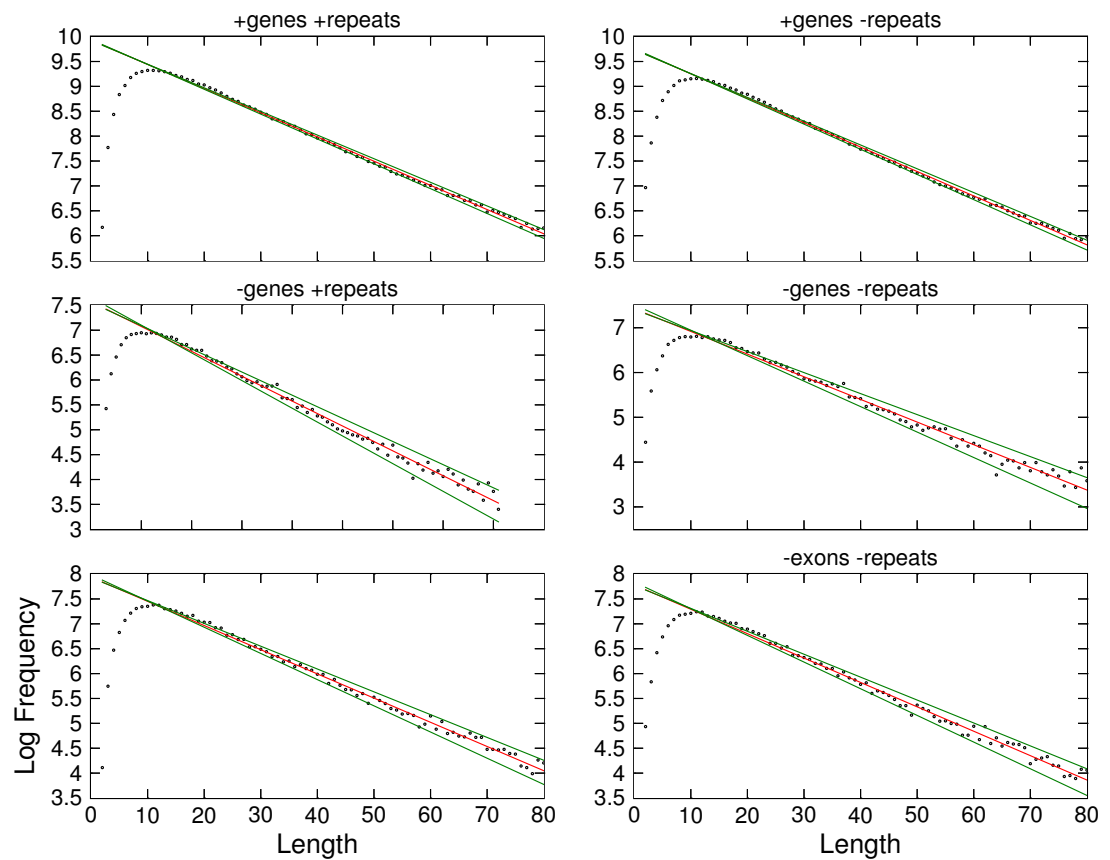


Figure 4.6: Blastz: The ungap Vs log frequency (black dotted) with the linear regression (red) and the 95% confidence intervals (green) shown for chr 21.

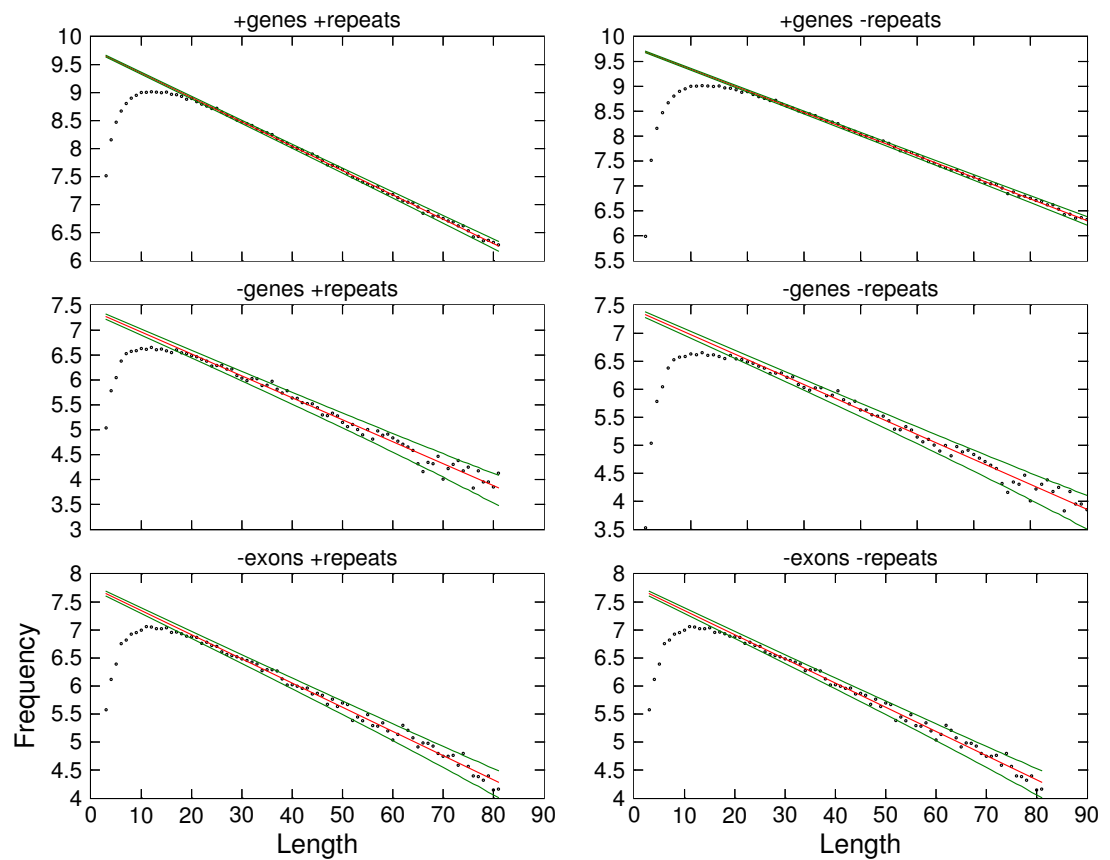
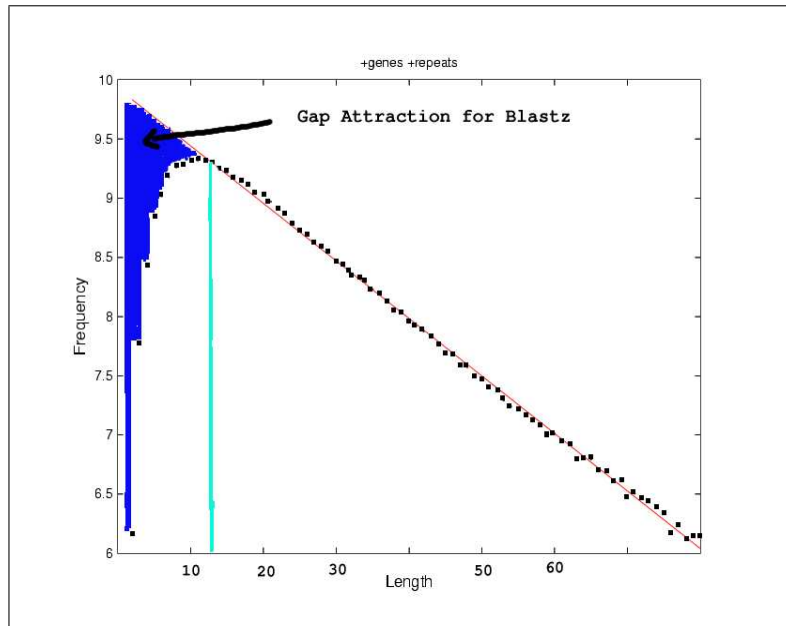
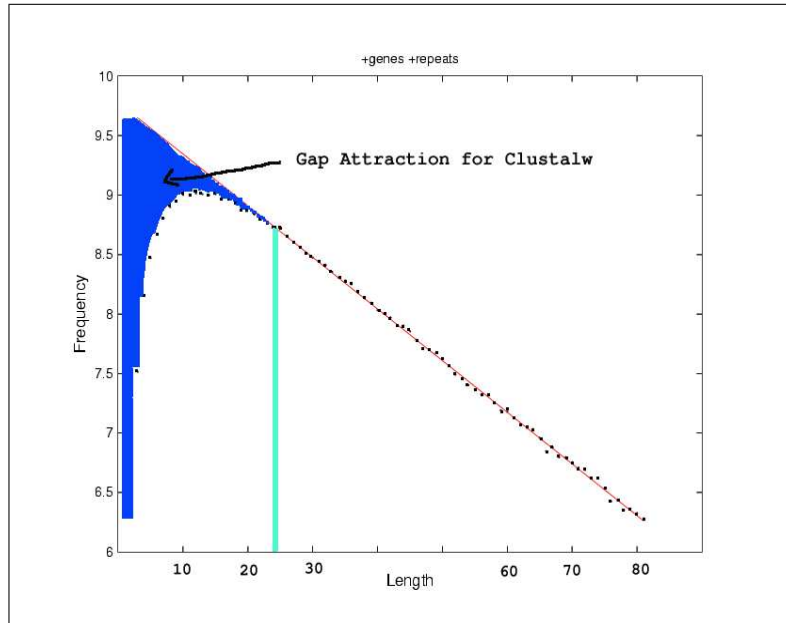


Figure 4.7: Clustalw: The ungap Vs log frequency (black dotted) with the linear regression (red) and the 95% confidence intervals (green) shown for chr 21.



(a) Gap Attraction for Blastz



(b) Gap Attraction is visibly more significant for Clustalw

Figure 4.8: A graphical comparison of the Gap Attraction measure for Blastz and Clustalw

## Chapter 5

# Discussion and Future Work

We present an objective measure of alignment algorithms called: "Gap Attraction". The gap attraction measure is based on the hypothesis that indels occur uniformly along the sequence, and independently of each other. The first assumption probably does not hold exactly, and is certainly violated in functional and thus conserved regions, this is why we filter out all annotated genes (1.5%). The second is likely to be valid for all genomic areas. Indels can be modelled as a Poisson process. From that, it follows that the distribution of ungap, the ungapped aligned regions between two random neighbouring indel events, is geometric. The hypothesis was validated by the data for the blastz alignment of the whole human genome and chromosome 21 with the mouse genome, as well as the clustalw alignment of human chromosome 21 with the mouse genome. A best linear fit for ungap lengths 0-80 was produced, and the 95% confidence intervals were calculated. For ungap lengths of medium lengths, the data, consisting of the whole genome, blastz and clustalw alignments of chromosome 21, fitted within the intervals.

The model is accurate for medium length ungap. Longer ungap, more than 80 nucleotides, did not fit the model because they represent the conserved regions not taken into account by the assumption of uniformity. The sparsity of shorter ungap is also observed. It represents those misaligned ungap pulled to one side favouring a long gap over two shorter gaps, and is therefore due to alignment error. For those data points whose frequency is smaller than expected, the difference with the best linear fit was calculated. Graphically, it is the surface of the region lying underneath the best linear fit line (figure 4.8). The resulting measure was divided by the total number of nucleotides, for the different gene and repeat options as well as the different gap options. The final value is the "gap attraction" measure.

Gap attraction was measured for two alignment algorithms; blastz which was developed for human mouse alignment and clustalw which performs global pairwise alignment. We found the gap attraction measure to be higher for clustalw indicating that it has worse performance. Although this was expected, it had never been possible to compare these two algorithms before.

"Gap attraction" represents to what extent two gaps, separated by an ungap, are attracted to each other, pulling the ungap in the middle towards one of the two sides, and resulting in a single longer gap. It is by no means an exact measure of all the errors an

alignment algorithm makes. For example, it does not measure:

1. The incorrect placing of an ungap. When the ungap is pulled one side or the other, gap attraction does not give a measure of which one results in the more accurate alignment.
2. The quality of ungap alignment. Gap attraction only measures the frequency of short ungap that is smaller than expected, assuming those whose frequency matches the expectancy are aligned correctly.
3. In measuring the frequencies of ungap that are smaller than expected, there is a certain threshold of ungap length that gap attraction applies to. For example, if an ungap is too short, then it will be pulled to the same side for both alignment algorithms, and if an ungap is too long, then it is easier to align anyway, and most alignment algorithms will get it right.

Rather, it measures one of the aspects of alignment error, allowing a novel way of classifying and comparing alignment algorithms, but not giving an exact measure of how much error they accumulate. Gap attraction is a very useful measure for evaluation when no benchmark exists, as is the case for aligning non-coding functional DNA, a task so far not possible to achieve as seen in the case of evaluating blastz or comparing it to clustalw.

Although in its infancy, the project led to very promising results. There is more work to be done, both in improving the current methods and extending them to get more results.

## 5.1 Weighted Linear Regression

While describing the ungap length 0-80 vs log frequency graphs, we mentioned the fact that ungap lengths 0-15 lied notably below the best linear fit, while lengths 15-25 lied on top of the line. A weighted linear regression that gives more weight to gap lengths 15-80 as opposed to 0-80 (only counting where data starts to fit the model) would probably produce a better fit, resulting in a higher figure for gap attraction.

## 5.2 Other Alignment Algorithms

It would be useful to evaluate more alignment algorithms such as: Lagan (Brudno et al., 2003), Dialign (Morgenstern, 1999), T-Coffee (Notredame et al., 2000), and Poa (Lee et al., 2002) etc. Some of these algorithms have already been evaluated against some existing benchmark (protein structures for example), and evaluating them according to our model could confirm/disprove what is known. Additionally, for this work, it was expected that blastz was going to perform better since it was developed for the task at hand. Also, clustalw has a gap opening penalty that is smaller than blastz, which might make up some of the gap attraction measure. Comparing blastz against these algorithms that are known to do better than clustalw, and with different gap opening penalties, will shed some light on the differences in the gap attraction

measure. Ultimately, a gap attraction measure for different algorithms for the whole of the human genome would be interesting (although not currently feasible for algorithms other than blastz).

### **5.3 Pair-HMM Alignment**

From our model, we could work at extracting the parameters that minimise gap attraction resulting in a pair-HMM alignment algorithm that performs better than existing algorithms. For example, obtaining the indel rate from the geometric distribution for a start. We could also have different parameters for insertion and deletion, resulting in a more exact model according to our hypothesis.

### **5.4 Data Simulation**

Simulating data could help achieve a better understanding of the significance of the "gap attraction" measure in two ways. First, the model is novel and has some assumptions which there is evidence against. Simulating a Poisson indel process on two sequences, aligning them with blastz, and then comparing that with the geometric distribution we obtain could confirm the validity of the hypothesis. Second, simulating sequences where the indel parameters are known, aligning them, and then comparing the resulting gap attraction with our current finding should shed some more light on the significance of gap attraction and its relation to alignment accuracy.

### **5.5 Estimating Functional DNA**

From the graph in figure 4.1a, we noted that there are a few long ungaps indicating conserved regions. This might ultimately lead to estimating functional DNA, and eventually non-coding functional DNA.

# Bibliography

- Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W. and Lipman, D. J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–402.
- Ashburner, M., Mishra, S., Roote, J., Lewis, S. E., Blazej, R., Davis, T., Doyle, C., Galle, R., George, R. and Harris, N. (1999) An exploration of the sequence of a 2.9 Mb of the genome of *Drosophila melanogaster*. The *adh* region. *Genetics*, **153**, 179–219.
- de Bakker, P., Bateman, A., Burke, D., Miguel, R., Mizuguchi, K., Shi, J., Shirai, H. and Blundell, T. (2001) HOMSTRAD: adding sequence information to structure-based alignments of homologous protein families. *Bioinformatics*, **17**, 748–749.
- Brudno, M., Cooper, G., Kim, M. F., Davydov, E., Green, E. D., Sidow, A. and Batzoglou, S. (2003) LAGAN and Multi-LAGAN: efficient tools for large-scale multiple alignment of genomic DNA. *Genome Research*, **13**, 721–731.
- Collins, D. W. and Jukes, T. H. (1994) Rates of transitions and transversions in coding sequences since the human-mouse divergence. *Genomics*, **20**, 386–396.
- Durbin, R., Eddy, S., Krogh, A. and Mitchison, G. (1998) *Biological Sequence Analysis*. Cambridge University Press.
- Feng, D. F. and Doolittle, R. F. (1987) Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *Journal of Molecular Evolution*, **25**, 351–360.
- Frazer, K. A., Sheehan, J. B., Stokowski, R. P., Chen, X., Hosseini, R., Cheng, J. F., Fodor, S. P., Cox, D. R. and Patil, N. (2001) Evolutionarily conserved sequences on human chromosome 21. *Genome Res.*, **11**, 1651–1659.
- Gerstein, M. and Levitt, M. (1998) Comprehensive assessment of automatic structural alignment against a manual standard, the scop classification of proteins. *Protein Sci.*, **7**, 445–456.
- Gotoh, O. (1982) An improved algorithm for matching biological sequences. *Journal of Molecular Biology*, **162**, 705–708.
- Griffiths, A. J. F., Gelbart, W. M., Lewontin, R. C. and Miller, J. H. (2002) *Modern Genetic Analysis*. W. H. Freeman and Co.

- Hattori, M., Fujiyama, T. D., Taylor, H., Watanabe, H., Yada, H., Park, H. S., Toyoda, A. and Ishii, K. (2000) The DNA sequence of human chromosome 21. *Nature*, **405**, 311–319.
- Higgins, D. G., Bleasby, A. J. and Fuchs, R. (1992) CLUSTALV: Improved software for multiple sequence alignment. *Computer Applications in the Biosciences*, **5**, 151–153.
- Holm, L. and Sander, C. (1993) Protein folds and families: sequence and structure alignments. *J Mol Biol.*, **233**, 123–138.
- (1999) Protein folds and families: sequence and structure alignments. *Nucleic Acids Res.*, **27**, 244–247.
- International Human Genome Sequencing Consortium (2001) Initial sequencing and analysis of the human genome. *Nature*, **409**, 860–921.
- Lassman, T. and Sonnhammer, E. L. (2002) Quality assessment of multiple alignment programs. *FEBS letters*, **529**, 126–130.
- Lee, C., Grasso, C. and Sharlow, M. F. (2002) Multiple sequence alignment using partial order graphs. *Bioinformatics*, **18**, 452–464.
- Ma, B., Tromp, J. and Li, M. (2002) PatternHunter: faster and more sensitive homology search. *Bioinformatics*, **18**, 440–445.
- Mallika, V., Anirban, B. and Sowdhamini, R. (2002) PASS2: Semi-automated database of protein alignments organized as structural superfamilies. *Nucleic Acids Res.*, **30**, 248–288.
- Marsden, B. and Abagyan, R. (2004) SAD: A normalised structural alignment database: improving sequence-structure alignments. *Bioinformatics Advance Access*.
- Marti-Renom, M., Ilyin, V. and A., S. (2001) DBAli: a database of protein structure alignments. *Bioinformatics*, **17**, 746–747.
- Mizuguchi, K., Deane, C. M., Blundell, T. L. and Overington, J. (1998) HOMSTRAD: a database of protein structure alignments for homologous families. *Protein Sci.*, **7**, 2469–71.
- Morgenstern, B. (1999) DIALIGN2: improvement of the segment-to-segment approach to multiple sequence alignment. *Bioinformatics*, **15**, 211–218.
- Mouse Genome Sequencing Consortium (2002) Initial sequencing and comparative analysis of the mouse genome. *Nature*, **420**, 520–562.
- Mural, R., Adams, M. D., Myers, E. W., Smith, H. O., Miklos, G. L., Wides, R., Halpern, A., Li, P. W., Sutton, G. G., Nadeau, J., Salzberg, S. L., Holt, R. A., Kodira, C. D., Lu, F., Chen, L., Deng, Z., Evangelista, C. C., Gan, W., Heiman, T. J., Li, J., Li, Z., Merkulov, G. V., Milshina, N. V. and Naik, A. K. (2002) A comparison of whole-genome shotgun-derived mouse chromosome 16 and the human genome. *Science*, **296**, 1661–1671.

- Needleman, S. B. and Wunsch, C. D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, **48**, 443–453.
- Notredame, C., Higgins, D. G. and Heringa, J. (2000) T-Coffee: A novel method for fast and accurate multiple sequence alignment. *J. Mol. Biol.*, **302**, 205–217.
- Pollard, D. A., Bergman, C. M., Stoye, J., Celniker, S. E. and Eisen, M. B. (2004) Benchmarking tools for the alignment of functional noncoding dna. *BMC Bioinformatics*, **5**.
- Rojic, S., Mackwoth, A. K. and Ouellette, F. B. F. (2001) Evaluation of gene-finding programs on mammalian sequences. *Genome Research*, **11**, 817–832.
- Schwartz, S., Kent, W., Smit, A., Zhang, Z., Baertsch, R., Hardison, R., Haussler, D. and Miller, W. (2003) Human-mouse alignments with blastz. *Genome Res.*, **13**, 103–107.
- Schwartz, S., Zhang, Z., Frazer, K. A., Smit, A., Riemer, C., Bouck, J., Gibbs, R., Hardison, R. and Miller, W. (2000) PipMarker: A web server for aligning two genomic DNA sequences. *Genome Res.*, **10**, 577–586.
- Shindyalov, I. and Bourne, P. (2001) Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng.*, **11**, 739–747.
- Smith, T. F. and Waterman, M. S. (1981) Identification of common molecular subsequences. *Journal of Molecular Biology*, **147**, 195–197.
- Strachan, T. and Read, A. P. (2004) *Human Molecular Genetics*. Garland Publishing.
- Thompson, J., Plewniak, F., Ripp, R., Thierry, J. and Poch, O. (2001) Towards a reliable objective function for multiple sequence alignments. *J Mol Biol.*, **314**, 937–51.
- Venclovas, C. (2003) Comparative modeling in CASP5: progress is evident, but alignment errors remain a significant hindrance. *Proteins*, **53**, 380–388.

## Appendix A

# The Python Database

<b>EnsGeneView:</b>	
descr:	List of genes in ensGene.txt file
params:	filename (txt), chromosome (txt)
fields:	{'chromosome':0,'dir':1,'start':2,'end':3,'codingstart':4,'codingend':5,'exons':6}
note:	If chromosome == "", creates view on entire file
bug:	If chromosome == "", nucleotide position still refers to beginning of chromosome
<b>AlignmentParView:</b>	
descr:	List of alignments (without data, without actual alignment) in .axt file
params:	filename (txt)
fields:	{'start1':0,'end1':1,'start2':2,'end2':3,'chr1':4,'chr2':5,'strand':6,'score':7,'filepos':8}
<b>AlignmentView:</b>	
descr:	List of actual alignments
params:	-
views:	AlignmentParView
<b>RepeatView:</b>	
descr:	List of repeats in .fa.out file
params:	filename (txt)
fields:	{'swscore':0,'percdiv':1,'percdel':2,'percins':3,'chrom':4,'start':5,'end':6,'strand':7,'repeat':8,'class':9,'repstart':10,'repend':11,'id':12}
note:	repstart and repend are in positive orientation wrt repeat (unbracketed numbers in .fa.out file) strand is '+' or 'C' percentages are actually promillages, and integers id is converted to a unique value by adding a constant at thresholds chrom is a string (e.g. 'chr21')
<b>SubsetView:</b>	
descr:	View on subset of underlying view
params:	tester (tuple): (index,test,value), test whether view[x][index] is equal or different (test '==' or '!=') from value tester (function): tester(data) returns True or False according to whether data should be included
views:	single view
<b>ChromosomeView:</b>	
descr:	The chromosome from .fa file
params:	filename (txt)
note:	supports __getslice__

<b>SequenceSelectView:</b>	
descr:	Sequence of segments, corresponding to parts of a sliceable underlying view (i.e. sequence) that satisfies some criterion
params:	<p>window (int, default 10000), size of window of underlying view that is fed to criterion</p> <p>dx (int, default window), step size for window</p> <p>criterion (function), function that takes slice from underlying view and returns list of segments that should be included in view</p> <p>inverse (segment, default None): segment wrt which list should be inverted (if exists) (e.g. Seg.allSeg)</p>
views:	view supporting <code>__getslice__</code>
<b>SegmentView:</b>	
descr:	View of segments based on another view of segments, or lists of segments
params:	<p>singleSegment (int tuple (idxL,idxR)): indices into underlying view giving endpoints of segment</p> <p>segmentList (int): index into underlying view giving list of segments</p> <p>inverse (segment, default None): segment wrt which list should be inverted (if exists) (e.g. Seg.allSeq)</p> <p>shoulder (int, default 0): number of elements by which underlying segments are extended before combining &amp; inverting</p>
views:	may be more than 1
note:	only one of singleSegment and segmentList may be present. If none are present, plain list of segments is assumed as underlying view
<b>SeqStatView:</b>	
descr:	Window-based statistics of underlying sliceable view
params:	<p>window (int), dx (int)</p> <p>statistic (function), takes data segment, provides statistic</p> <p>matchCoords (boolean, default False), whether to keep underlying coordinates, or enumerate segments</p>
views:	<p>single view supporting <code>__getslice__</code></p> <p>second view is used as segment mask if present</p>
note:	matchCoords can be changed on-the-fly by changing view.matchCoords (True or False)
bug:	last (partial) segment is not included

Table A.1: The list of available DV views , with a short description of usage