Substring Differences in Genomes*

Hendrik Jan Hoogeboom Walter A. Kosters
Jeroen F. J. Laros
Leiden Institute of Advanced Computer Science
Universiteit Leiden, The Netherlands
jlaros@liacs.nl

December 22, 2008

Abstract

We introduce a new way of determining the difference between full genomes, based upon the occurrence of small substrings in both genomes. Basically we count the number of occurrences of all substrings of a certain length and use that to determine to what extent two genomes are alike. Based on these numbers several difference measures can be defined, e.g., a Euclidean distance in the vector space that has the same dimension as the number of possible substrings of a certain length, a multiset distance, or other measures. Each of these measures can be applied for phylogenetic tree generation. We also pay attention to some visualizations and several statistics.

1 Introduction

Determining how one species relates to the other can be done in many ways. One of the many techniques is to look at the DNA. At this moment, many genomes can be downloaded from the internet [11], although not all genomes are complete yet. Also note that the full genomes of many individuals of a given species become publicly available.In this document, we do not look for genes or markers in the genome, or any other annotation whatsoever, but just at the occurrence of substrings. Therefore the techniques described here can be used for various other problems, ranging from chromosome resemblance to detection of plagiarism or document similarities for search engines.
In Section 2, we describe a way to compare two long strings by counting rare substrings. This seemingly simple approach is highly non-trivial because the number of substrings is enormous. We have tried a number of ways to do the computation efficiently, like caching, using trees and hashtables, but all these

*Presented at Benelux Bioinformatics Conference 2008 (BBC 2008), Proceedings pp. 62, Maastricht, The Netherlands, 15-16 December 2008.

methods need far too much memory. Note that if the substrings are small, these methods are just fine and solve the problem in linear time, but we deal with rather large substrings (length 14 and above). The only way to count large substrings was to use an exponential (in memory) approach, but with the adaption that if the amount of available memory is not enough, we make multiple passes over the data; in each pass we search for all substrings with a preset prefix.
The determination of non frequent substrings is reintegrated in the genome data. So we know the exact position. Unfortunately, the DNA at a certain position in the genome of one species does not have to correspond to the DNA at the same position of the genome of another species. This annotated genome can be used in a large variety of application. One of them we shall discuss in this paper.
We will show some elementary statistics and visualizations in Section 3, a distance measure for the comparison of two species in Section 4 and from this the generation of phylogenetic trees in Section 5. In Section 6 we introduce a distance measure for multisets which we apply to our data. We conclude in Section 7.

2 Determining rare factors

The discovery of (almost) unique substrings of a given length n itself is certainly not as trivial as it might seem. In principle we do the following:

- Convert the entire genome to a binary sequence, using a suitable encoding scheme.
- Use a sliding window to get all subsequences of length n.
- Count these subsequences and remember in which genome each of the subsequences were found.

2.1 Conversion

In Table 1 we see one of the possible binary encodings for nucleotides. However, these values were not chosen at random. Note that the complementary letters are also complementary in the binary encoding, i.e., A and T are complementary, and so are 00 and 11. There is an advantage in such a scheme, because the calculation of the complement of such a string is a very simple and fast operation. We shall see further on why this is important.

nucleotide	A	C	G	T
encoding	00	01	10	11

Table 1: Binary encoding of the nucleotides

2.2 Sliding window

After we have converted the DNA to binary data, we use a sliding window to get all subsequences of a certain length. If we are searching for substrings of length n, we make an array of size 4^n. Each time we move the window we get another position in the array, and we simply increase the value of that array element. An advantage of this sliding window is that we only have to read one value to generate the next index. We simply shift the old index to the left and concatenate the newly read value to the end of the sequence.
The size of the array is the main difficulty in this approach. To give an indication: for $n = 16$, we have to make an array with 4^{16} entries, and if each entry consists of one byte, the array will be 4 Gigabytes large. The size of the input files have no influence on this array.
To make sure that all random access is done in memory, we use a memory locked part of the main memory. In practice, this will probably be smaller than the amount of required memory. Therefore, we make multiple passes over our input where in each pass a prefix is fixed. For example, if we require 4 Gibabytes of memory and we can only lock 2 Gigabytes, we make two passes over the input. In the first pass the fist bit is fixed and has the value 0. This means that in the first pass all substrings are counted that start with an `A` or `C`. This implies that the amount of physical memory used must always be a power of 2.
For substrings of length 18 and below, this is probably the most efficient datastructure to work with, this is because the genome of most species is so large that most (if not all) combinations occur. For the human genome we know that about 95% of the substrings of length 18 are unique. If we put this data in a space-efficient datastructure like a trie, we need about 650 Gigabytes (twice as much as one might expect, but this is because we need to use 64 bits pointers). If we use a PATRICIA tree [7], we still need about 100 Gigabytes (we base this assumption on the fact that the branching factor of the tree is very high).
The reason that we chose to use strings with a length between 12 and 18, is because using larger strings than 18 is not really needed (most of the substrings of this length are unique) and below 12 there are too few unique substrings.

2.3 Counting

Since DNA is double stranded, we can not simply count all substrings, because the reverse complement of a string is essentially the same as the original string. Therefore we calculate the reverse complement of each index as well. One of these two indexes has the smallest numerical value, and this one will be used as representative of the pair. Keeping track of the reverse complement is just as easy as keeping track of the original string. The difference is that we shift the old index to the right and insert the inverted newly read value to the beginning of the sequence. If we now encounter either of those sequences, we increase the value of the sequence with the lowest binary representation. This way both the sequence and its reverse complement are mapped to the same position in the counting table.

This results in a table where every possible subsequence is counted, however due to physical limitations (the size of the table in memory and the size of the annotated genomes that will be written to disk afterwards), we chose to count up to three, so if there is a three in this table, it means the corresponding substring is present three or more times. To be precise, each nibble in the counting table is used as an entry. This reduces the amount of memory for the table by a factor of two, but it complicates writing and reading in the table slightly; for a even substring (ending in `A` or `G`) we have to do an AND with the value $0 \times 0F$ and in the other case we do a SHIFT RIGHT of 4 bits.
Another technical detail is that we use the first half of the value of each element in the counting table for species A and the second half for species B (leaving only two bits for each species). This is quite convenient since we can look up the number of occurrences of a certain substring in both species at once.

3 Elementary statistics and visualizations

Now we can make all sorts of elementary statistics and visualizations. For example, we can give the number of unique strings of a given length and even the position of these strings. As a first example, in Table 2 the number of unique substrings of a certain length is shown. Note that each unique string of length n automatically accounts for $(m - n) + 1$ unique strings of length m, if $m > n$.

size	11	12	13	14	15	16
Human	210	47,668	1,335,256	15,412,176	85,793,791	346,600,204
Chimp	300	62,149	1,509,471	16,636,054	87,029,038	346,319,725

Table 2: Number of unique substrings in Human and Chimp

Of course, this does not account for all unique strings of high length as can be seen in Table 2. The values grow far more rapid than the ones dictated by the formula above. Statistically, given a random string, the larger the length of the substring, the higher the chance is that a given substring is unique. This is the reason we find far more unique strings of higher length.
In Figure 1 we see the occurrence of unique strings in a Human. We visualize the number of unique strings of length 12, for each consecutive series of 100,000 basepairs. The vertical dotted lines denote the chromosome boundaries; these are ordered as follows: 1 to 22, then X, Y, and finally the mitochondrial DNA. This mitochondrial DNA is so small that it does not show up in these graphs. The white bands located at offset 1300 and 15900, for example, are due to unsampled or highly unstable DNA, and in either case it is missing from our input.
In Figure 2 we only plot the occurrences above 15. In Figure 3 we have plotted the number of occurrences of strings that are unique in the human genome and not present in the genome of a chimp. Figure 4 shows unique strings for a chimp. The chromosomes are ordered 1 to 22, then Un, X, Y, and finally the

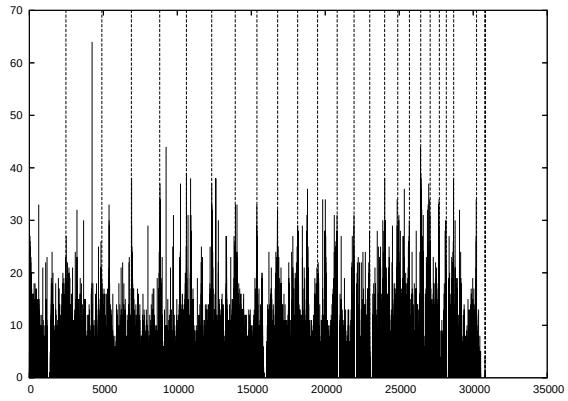

Figure 1: Occurrence of unique strings in Human

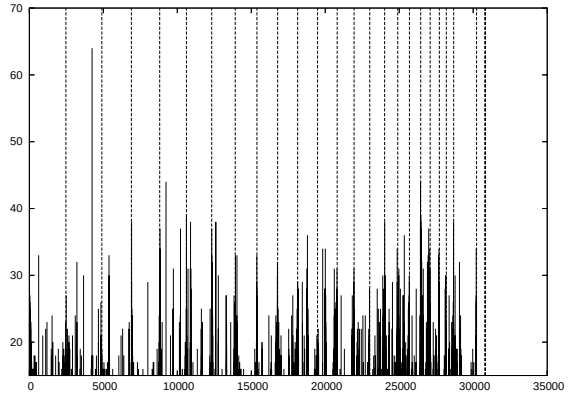

Figure 2: Occurrence of unique strings (length above 15) in Human

mitochondrial DNA. This can for example be used to select markers to put on a *microarray* [9] to make a distinction between two (or more) species. In Figure 3 and 5, we see where these markers can be found. Another application is the selection of *primers* [4], commonly used in techniques like *Multiplex Ligation-dependent Probe Amplification* (MPLA) [10] and *Polymerase Chain Reaction*, or PCR [2]. We see the regions where the number of primers are abundant in Figure 2 for the human and in Figure 4 for the chimp.

4 Distances and weights

After the substrings of length n have been counted, we make a matrix where two species are represented by counting the number of strings that occur a times in

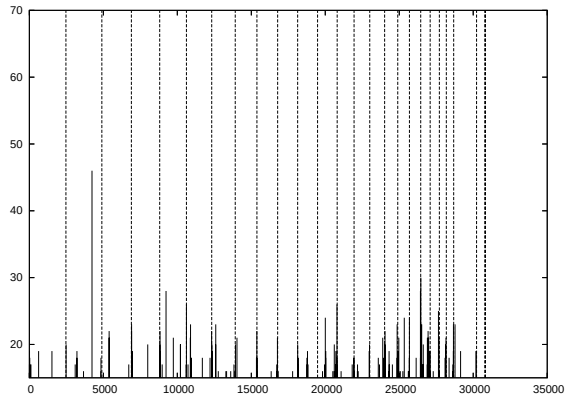

Figure 3: Occurrence of unique strings present in Human and not in Chimp

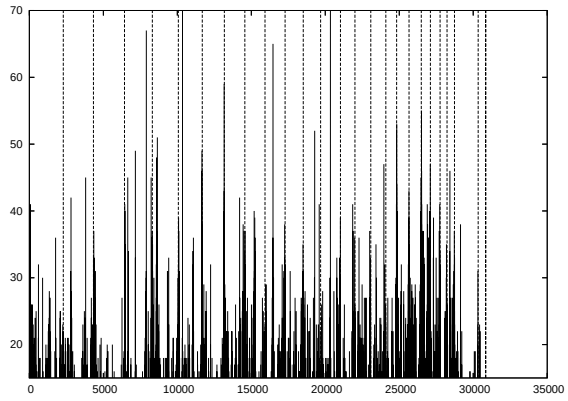

Figure 4: Occurrence of unique strings in Chimp

species A and b times in species B for $0 \leq a, b \leq 3$. This is a method that loses a lot of information, but we have a very small matrix left to work with and as we shall see further on, the information in this matrix is still sufficient to make a difference between species. The 4×4 matrix M, referred to as the *counting matrix*, contains the data:

$$M = M(A,B) = (m_{i,j}) = \begin{pmatrix} m_{0,0} & \underline{m_{0,1}} & \underline{m_{0,2}} & \underline{m_{0,3}} \\ \underline{m_{1,0}} & m_{1,1} & \underline{m_{1,2}} & \underline{m_{1,3}} \\ \underline{m_{2,0}} & \underline{m_{2,1}} & m_{2,2} & \underline{m_{2,3}} \\ \underline{m_{3,0}} & \underline{m_{3,1}} & \underline{m_{3,2}} & m_{3,3} \end{pmatrix}$$

Here $m_{i,j}$ denotes how many substrings are present i times in species A and j times in species B. As mentioned before, we only count up to three, so, e.g., the element $m_{1,3}$ is the amount of substrings that are present once in species A and

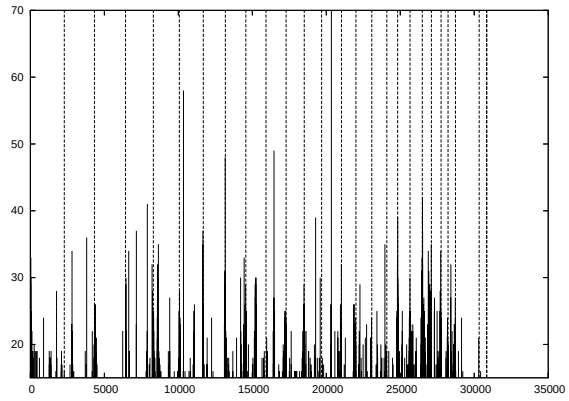

Figure 5: Occurrence of unique strings present in Chimp and not in Human

three or more times in species B. All elements that contribute to the difference are underlined to indicate the relevant elements of the matrix.
In principle, we want to use the following distance formula:

$$\text{dist}(S,T) = \frac{|S\backslash T| + |T\backslash S|}{|S \cup T|}, \tag{1}$$

where S and T are sets. We divide the symmetrical difference of set S and T by the maximum value of the numerator. If both S and T are the empty set, we let $\text{dist}(S,T) = 0$. The reason we chose for this particular distance measure is because it takes the sizes of the sets into account (also see [5]). To adjust this formula to work with our matrix, we have to rewrite it as follows (for species A and B):

$$\text{dist}(A,B) = \frac{\sum_{i=1}^{3}(m_{0,i} + m_{i,0})}{4^\ell - m_{0,0}}, \tag{2}$$

where $M = M(A,B)$ is the counting matrix of the pair (A,B) and $4^\ell - m_{0,0}$ is the total number of substrings that occur in at least one of A and B.
One of the shortcomings of this formula is that only the absolute differences are used, i.e., only the substrings present in one of the species. Another shortcoming is that all differences are weighted equally, although it is perhaps reasonable to assume that a substring that is present once has less significance than one that is present more than three times.
To compensate for these shortcomings, we use a *weighting matrix* W:

$$W = (w_{i,j}) = \begin{pmatrix} 0 & \alpha_3 & \alpha_4 & \alpha_5 \\ \alpha_3 & 0 & \alpha_0 & \alpha_2 \\ \alpha_4 & \alpha_0 & 0 & \alpha_1 \\ \alpha_5 & \alpha_2 & \alpha_1 & 0 \end{pmatrix}$$

The values of $\alpha_0, \ldots, \alpha_5$ are weights applied to the matrix M. They are ordered in ascending order of significance, e.g., we assume that the value of $m_{2,1}$ is less

significant than $m_{3,2}$, and therefore we should set α_0 to a lower value than α_1. We base this assumption on the fact that if a substring is present zero times in species A and two times in species B, this is a more significant difference than once in species A and two times in species B for example. We now define

$$\text{dist}(A, B, W) = \frac{\sum_{i,j} w_{i,j} m_{i,j}}{\max(\alpha_0, \ldots, \alpha_5)(4^\ell - m_{0,0})}, \tag{3}$$

where W is the weighting matrix and M is the counting matrix. We calculate the weighted sum of the relevant matrix elements and divide by the maximum possible difference. Note that this is a generalization of Equation 2, if we choose $\alpha_0 = \alpha_1 = \alpha_2 = 0$ and $\alpha_3 = \alpha_4 = \alpha_5 = 1$, then we get Equation 2 again.

5 Experiments and results

We have done two types of experiments. The first one is the comparison of a pair of species, the second one is extracting a distance from the first experiments and to combine a number of species in a distance matrix.

5.1 Some raw data

For the following results, we have chosen to look at sequences of length $n = 14$. Table 3 is the raw comparison matrix of the human genome and that of a chimp. Notice that the number at position $(0, 0)$ is huge and non-informative. The other numbers on the main diagonal are also relatively large. This might mean that there are lots of similarities between the two species. The number at $(3, 3)$ is also very large, but that is because it is actually the sum of all points (x, y) with $x, y \geq 3$. Actually, all points $(3, x)$ and $(x, 3)$ with $x \in \mathbb{N}$ are less informative than the other numbers in this matrix, because we can not be sure if the 3 "is actually" a 3.

		Human			
		0	1	2	≥ 3
Chimp	0	150,783,349	4,486,933	1,216,093	498,090
	1	3,212,656	7,352,318	3,737,739	2,333,341
	2	602,927	2,621,970	4,011,169	4,907,515
	≥ 3	145,530	950,955	2,697,230	78,877,641

Table 3: Differences between Human and Chimp

We will compare these figures with the difference between a cow and yeast. In Table 4, we see a quite different picture. The matrix is even less symmetrical. We can think of two reasons for this. Firstly a cow and yeast are quite different species, and secondly the genome of yeast is a lot shorter than that of a cow. Therefore a given random string is more likely to be present in a cow, so the matrix is what we would expect it to be.

		Yeast			
		0	1	2	≥ 3
Cow	0	153,248,529	544,363	21,518	5,229
	1	15,023,538	548,614	25,707	5,624
	2	11,361,444	489,848	26,124	5,459
	≥ 3	78,706,409	7,293,480	876,010	253,560

Table 4: Differences between Yeast and Cow

5.2 Visualization of the raw data

For the following results, we have chosen to look at sequences of length $n = 16$. In Figure 6 we have plotted an interpolation of the values in the matrix M

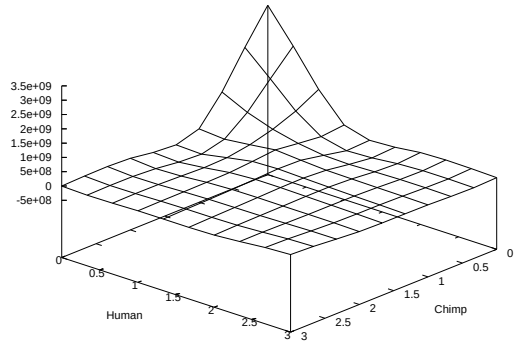

Figure 6: Human-Chimp raw data

of the human genome and that of the chimp. Notice that although the matrix contains lots of information, the graph is almost symmetric, this is also the case for the almost identical Figure 7. This is because the similarities between the two species are much larger than the differences. Another very disturbing factor is the point at $(0, 0)$: this is where all substrings that are not present in either species are. If the length of the substrings becomes too large (like here), this peak will be enormous. This is why we chose to leave out similarities in the next two pictures. In Figure 8 the difference between the human genome and that of a chimp is plotted. The main diagonal has been removed from the data to emphasize the differences (the values at these positions are interpolated). The same technique is used in Figure 9, where this time the difference between human and cow is plotted.

5.3 Comparison of many species

We have taken the genomes of the species shown in Table 5 from [11].

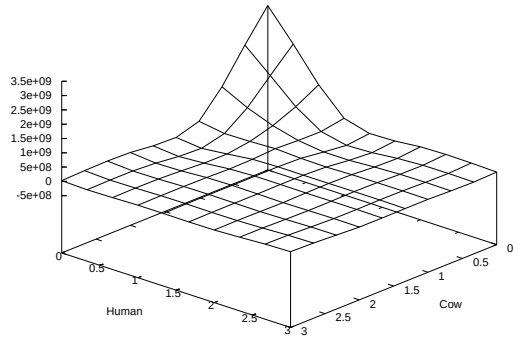

Figure 7: Human-Cow raw data

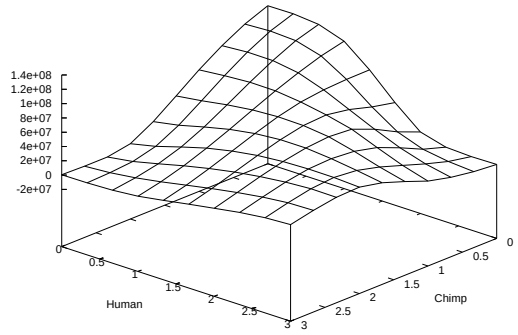

Figure 8: Human-Chimp raw data, main diagonal removed

In Table 6 we give the distance matrix, where we took $\alpha_i = 1$ for all i. Notice that since this is a distance matrix, we do not have to show the upper half of the matrix, because we can just mirror it in the main diagonal. Also note that although we have not done any weighting, some things are already remarkable, for instance: SARS is at distance 0.999 to all of the other species (as expected) and the lowest distance (0.446) is the one between a human and a chimp.
In Table 7 we took $\alpha_0 = 1, \alpha_1 = 2, \alpha_2 = 4, \alpha_3 = 10, \alpha_4 = 20$ and $\alpha_5 = 1$. These values are taken quite arbitrary, as the authors of this document are by no means genetic experts. The reason we chose for this particular set of values, is because if we assume that the two genomes are normal random strings, this would be a nice weighting scheme.
From these distance matrices we can make a *phylogenetic tree*, also see [1]. We chose to make two visualizations, one rooted tree in which the distances are

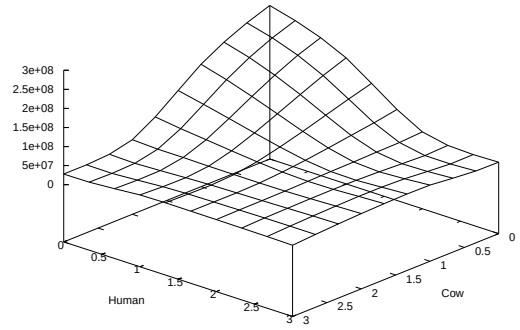

Figure 9: Human-Cow raw data, main diagonal removed

species	abbreviation	genome length
Bee	B	$5.13 \cdot 10^8$
C_elegans	Ce	$1.02 \cdot 10^8$
Chicken	Ci	$1.13 \cdot 10^9$
Chimp	C	$3.15 \cdot 10^9$
Cow	Co	$3.60 \cdot 10^9$
Dog	D	$2.58 \cdot 10^9$
Drosophila_melanogaster	Dm	$1.35 \cdot 10^8$
Human	H	$3.15 \cdot 10^9$
SARS	S	$3.69 \cdot 10^4$
Yeast	Y	$1.24 \cdot 10^7$

Table 5: Species

	Y	S	H	Dm	D	Co	C	Ci	Ce	B
Y	.000									
S	.999	.000								
H	.997	.999	.000							
Dm	.990	.999	.979	.000						
D	.997	.999	.740	.977	.000					
Co	.997	.999	.744	.977	.750	.000				
C	.997	.999	.442	.977	.748	.752	.000			
Ci	.995	.999	.834	.968	.833	.833	.830	.000		
Ce	.988	.999	.984	.959	.982	.983	.982	.973	.000	
B	.991	.999	.971	.957	.969	.969	.969	.959	.953	.000

Table 6: Distance matrix for $\alpha_i = 1$ for $i = 0, 1, 2, 3, 4, 5$

	Y	S	H	Dm	D	Co	C	Ci	Ce	B
Y	.000									
S	.507	.000								
H	.442	.443	.000							
Dm	.517	.525	.431	.000						
D	.463	.464	.293	.450	.000					
Co	.456	.457	.294	.443	.301	.000				
C	.459	.461	.142	.447	.300	.301	.000			
Ci	.514	.518	.340	.491	.349	.348	.346	.000		
Ce	.513	.524	.435	.494	.454	.447	.450	.495	.000	
B	.519	.526	.433	.494	.451	.445	.448	.488	.491	.000

Table 7: Distance matrix with weight (large α_3 and α_4, see text)

not preserved and one unrooted tree where distances are preserved as much as possible. Of course, since this is only a projection of the actual data, more trees can be drawn apart from these ones. In Figure 10 and 11 we see a rooted tree

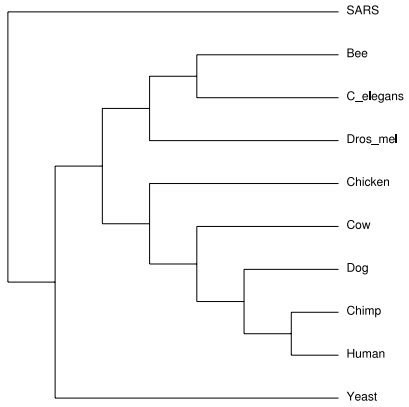

Figure 10: Phylogenetic tree of Table 6

in which distances are not preserved. This is only to give the reader a global view of the distances between the given species. In Figure 12 and 13 we see an unrooted tree with partially preserved distances. The path from yeast to SARS for example is shorter than the path from yeast to cow. We see a difference in the warm-blooded animals when we compare these trees, the triple Dog, Cow, Chicken seem to be affected by our choice of weights. These differences can be observed in both the rooted and the unrooted trees.
Figure 10, 11, 12 and 13 are constructed by means of the *Fitch-Margoliash* [3] algorithm. They are visualizations of the matrices in Table 6 and 7.

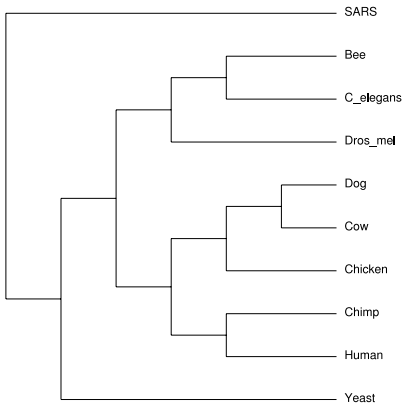

Figure 11: Phylogenetic tree of Table 7

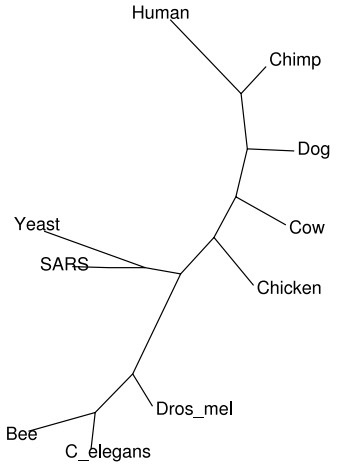

Figure 12: Unrooted phylogenetic tree of Table 6

6 Using a new distance measure

In this section we use a new distance measure designed for multisets [6]. This metric is parameterized by a function f that, given a few restrictions, will give a valid metric. We shall adhere to these restrictions.
The generic distance measure is defined as follows:

$$d_f(X,Y) = \frac{\sum_{i=1}^{n} f(x_i, y_i)}{|S(X) \cup S(Y)|}$$

The numerator is the sum of values of a function f, that indicates the difference between the number of elements in one category. In this case, the difference in

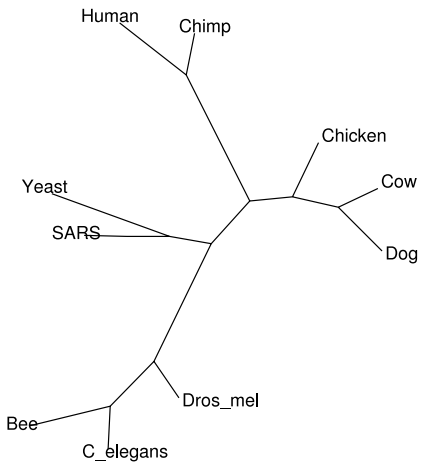

Figure 13: Unrooted phylogenetic tree of Table 7

occurrences of a particular piece of DNA. The denominator is the number of categories, in this case, the number of strands of DNA present in either of the samples.
The function we use is:

$$f(x, y) = \frac{|x - y|}{(x + 1)(y + 1)}$$

The reason for using this function is quite intuitive. If a particular strand of DNA is present once in one of the samples, and not in the other sample, the function will return distance 1/2. But when this strand is present in one sample once and twice in the other, the distance will be 1/6. In other words, the fact that two samples share a piece of DNA or not, is more important than the number of occurrences, though the latter is included.
Using the metric described above, we obtain the distance matrix shown in Table 8.
The rooted and unrooted phylogenetic trees are shown in Figure 14 and 15.

7 Conclusions and further research

We have shown that determining rare substrings in a genome is possible up to a certain length. With the result we can make an annotated genome from which we can extract lots of data. The cumulative count of strings that occur n times in species A and m times in species B, where n, m are at most 4, still contains enough data to make a phylogenetic tree.
The techniques described in this document could also be used to discover *Single Nucleotide Polymorphisms* or SNP's [8] by using two individuals of the same species as input.

	Y	S	H	Dm	D	Co	C	Ci	Ce	B
Y	.000									
S	.505	.000								
H	.618	.623	.000							
Dm	.511	.519	.582	.000						
D	.610	.615	.315	.574	.000					
Co	.613	.618	.320	.577	.323	.000				
C	.611	.616	.150	.574	.320	.325	.000			
Ci	.581	.587	.389	.542	.388	.390	.386	.000		
Ce	.516	.528	.590	.490	.582	.584	.582	.549	.000	
B	.532	.542	.571	.493	.563	.565	.562	.531	.491	.000

Table 8: Distance matrix as calculated with the multiset metric

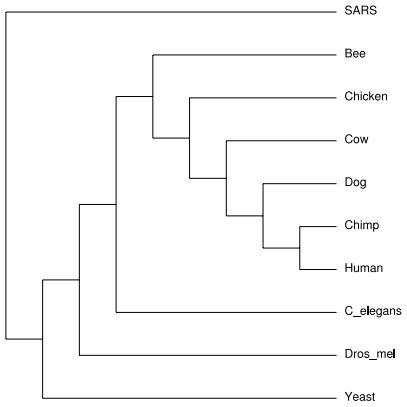

Figure 14: Phylogenetic tree of Table 8

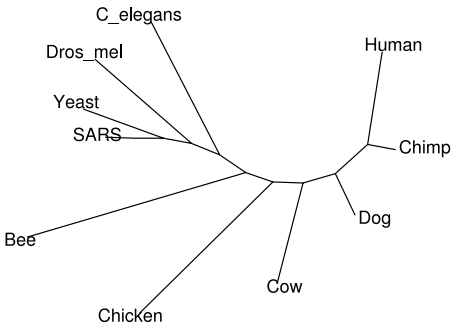

Figure 15: Unrooted phylogenetic tree of Table 8

For further research we could make a distance measure based on (some of) the unique strings themselves, not the amount of them. This way we could make a very accurate distinction between species or individuals.

8 Acknowlegements

The authors wish to thank Robert Brijder for his insightful ideas and support. This research is part of the DALE project (Data Asssistance for Law Enforcement) as financed in the ToKeN program from the Netherlands Organization for Scientific Research (NWO) under grant number 634.000.430.

References

[1] Chan, P. Y., Lam, T. W., and Yiu, S. M., A More Accurate and Efficient Whole Genome Phylogeny. Asia-Pacific Bioinformatics (2006): 337–351.

[2] Dieffenbach, C.W., and Dveksler, G.S., PCR Primer: A Laboratory Manual. CSHL Press, Cold Spring Harbor, USA, 1995.

[3] Fitch, W. M., and Margoliash, E., Construction of Phylogenetic Trees. Science 155 (1967): 279–284.

[4] Gibson, G., and Muse, S., A Primer of Genome Science (2nd Ed.). Boyle Biochemistry and Molecular Biology Education (2005): 33–313

[5] Glazko, G., Gordon, A., and Mushegian, A., The Choice of Optimal Distance Measure in Genome-wide Datasets. Bioinformatics 21 (2005): iii3–iii11.

[6] Kosters, W.A., and Laros, J.F.J., Metrics for Mining Multisets, Research and Development in Intelligent Systems XXIV, Proceedings of AI-2007, the Twenty-seventh SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence (SGAI 2007), Springer, pp. 293–303, 2007.

[7] Morrison, D. R. PATRICIA — Practical Algorithm to Retrieve Information Coded in Alphanumeric. Jrnl. of the ACM 15(4) (1968): 514–534.

[8] Oliphant, A., Barker, D. L., Stuelpnagel, J. R., and Chee, M. S., SNPs: Discovery of Markers for Disease. BioTechniques 32 (2002): S56–S61.

[9] Schena, M., Shalon, D., Davis, R. W., and Brown, P. O., Quantitative Monitoring of Gene Expression Patterns with a Complementary DNA Microarray. Science 270, no. 5235 (1995): 467–470.

[10] Schouten, J.P., McElgunn, C.J., Waaijer, R., Zwijnenburg, D., Diepvens, F., and Pals, G., Relative Quantification of 40 Nucleic Acid Sequences by Multiplex Ligation-dependent Probe Amplification. Nucleic Acid Research 30, No. 12, e57 (2002):1–13.

[11] UCSC Genome Bioinformatics, http://genome.ucsc.edu/ [version April 21, 2006].