A Fast Algorithm for the Exhaustive Analysis of 12-Nucleotide-Long DNA Sequences. Applications to Human Genomics.

Vicente Arnau
Departamento de Informática
Universidad de Valencia. Campus de Burjassot
Avda. Vicent Andrés Estellés, s/n.
46100 Burjassot. Valencia. Spain.
Vicente.Arnau@uv.es

Ignacio Marín
Departamento de Genética
Universidad de Valencia. Campus de Burjassot
Calle Doctor Moliner, 50.
46100 Burjassot. Valencia. Spain.
Ignacio.Marin@uv.es

Abstract

We have developed a new algorithm that allows the exhaustive determination of words of up to 12 nucleotides in DNA sequences. It is fast enough as to be used at a genomic scale running on a standard personal computer. As an example, we apply the algorithm to compare the number of all 12-nucleotide long words in human chromosomes 21 and 22, each of them more than 33 million nucleotides long. Sequences that are chromosome specific are detected in less than 2 minutes, being analyzed any pair of chromosomes at a rate of 45 millions of nucleotides (45 Mb) per minute. The size of the words is long enough as to allow further analyses of all significant sequences using conventional database searches. This allows to very simply establish the location and, many times, the biological meaning of the selected words. As an example, we show here, for the comparison between human chromosomes 21 and 22, that all the sequences that are found at least 40 times in one chromosome but are absent in the other belong to just two different classes, namely tandem repeats or genes with characteristic, internally repetitive, coding regions. Other available versions of this program and further applications are discussed.

1. Introduction

Genomic analysis is usually performed using brute force strategies, with a generalized use of multiple supercomputers and parallel processing of data. However, the improvement in frequency (e. g. 2.5 GHz), hard disk capacity (e.g. 80 Gbytes) and size of main memory (e. g. 1.5 Gbytes) in standard personal computers has opened new possibilities. It has become evident that to perform many types of complex genomic analyses, it is often more important to develop tools that optimize processing time that to buy new and expensive equipment. The purpose of this study is to show one of those new applications, that allows a very fast and exhaustive determination of all the DNA ”words” that exist in sequenced pieces of DNA of any size, including whole chromosomes or even genomes. There is a large literature of DNA word estimation and analysis (reviewed in [1]), but most of it is concentrated on short motifs. Thus, there are many works that analyze dinucleotides (for a recent example, see [2]), being one of the most significant results found the underrepresentation of the dinucleotide CG in vertebrate genomes, due to its conversion into CA or TG associated to DNA methylation [1]. Oligonucleotide composition has been used, together with other types of information, to establish gene promoters or coding regions [1]-[7] or to detect sites that are characteristic of regulatory regions of the genes [8] [9]. They can also be used to establish species-specific genomic signatures [1][10]. Therefore, a fast procedure to detect all words of a certain size may be of very general interest, especially if it can be applied to large sequences, as complete chromosomes or even genomes. In this study, we show the feasibility of fast analyses of words of up to 12 nucleotides on a personal computer. As a model for testing our procedures, we will show results from a comparison of human chromosomes 21 and 22. Notably, comparisons of those two chromosomes, each one of them about 33 Megabases (33 Mb, 33 million nucleotides) long, can be performed in a few minutes in a standard personal computer.

In the following section, we will detail the new algorithm, showing its general properties. In section 3, we will describe and validate the results when the method is applied to a real case: the search in whole human chromosomes for singular, chromosome-specific sequences. Section 4 contains some concluding remarks about the potential of this method.

0-7695-1926-1/03/$17.00 (C) 2003 IEEE

2. An Algorithm That Performs a Fast, Exhaustive Search of up to 12 Nucleotides-Long Words at a Genomic Scale and Using a Personal Computer

We will proceed now to explain the basics of our procedure. As model sequences, we will use human chromosomes 21 and 22. These two chromosomes have been chosen because they are fully sequenced and are the best studied human chromosomes in terms of structure, number and location of the genes, repetitive DNA, and other interesting characteristics (e. g. [11]-[14]).

Let us first consider the complexity of the problem. On one hand, although chromosomes 21 and 22 are the smallest human chromosomes, their size is considerable. Each one has more than 33 millions of nucleotides (33 Mb). That is much larger than the size of the whole genome of other eukaryotes, as the fully sequenced and extensively analyzed yeast *Saccharomyces cerevisiae* (about 12 Mb) and many times larger than the genome of most prokaryotes (e. g. *Escherichia coli*, the most analyzed bacterium, has a genome size of about 5 Mb). Therefore, the comparison of these two chromosomes is a good test at a genomic scale. On the other hand, there are four different nucleotides, and thus the total number of different sequences of 12 nucleotides is 4^{12} or about 16.8 millions. Considering these numbers, it is evident that any exhaustive search algorithm based on sequential reading and adscription of all the words found in each of those chromosomes to one of those 16.8 millions of alternative possibilities will be too slow to be useful.

A different strategy must be found [15][16], and the one we have developed can be summarized as shown in Figure 1 and 2. The rationale of the algorithm is to establish a tree of solutions containing all different 12-nucleotide-long sequences found in a particular DNA entry sequence, together with their frequencies. A tree is started that has a root node from which four different pointers can be established, corresponding to nucleotides A, C, G or T, that lead to the four possible level 1 nodes. This node structure is repeated for nodes of levels 1 to 11. The final nodes (level 12 nodes) have a different structure, because they must store three fields: the word of 12 nucleotides that is recognized, the frequency of that word in the first sequence (e. g. chromosome 21) and the frequency of the same word in the second sequence (e. g. chromosome 22).

The tree of solutions is dynamically generated. Eleven pointers (q1 to q11) are addressed to NULL. An additional pointer, called Q, is used to read the nucleotides. The program starts by reading the first nucleotide, addressed by Q, of the entry sequence. Then, the first pointer (q1) is addressed to the level 1 node that corresponds to the read nucleotide. When a second nucleotide is read, the second pointer, q2, is addressed to the level 2 node that corresponds to the read dinucleotide and q1 shifts to the level 1 node created by the second nucleotide. This process continues until the first eleven nucleotides are read. From then on, pointers q1 to q11 are addressed to the last strings read of lengths 1 to 11, respectively (Figure 1).

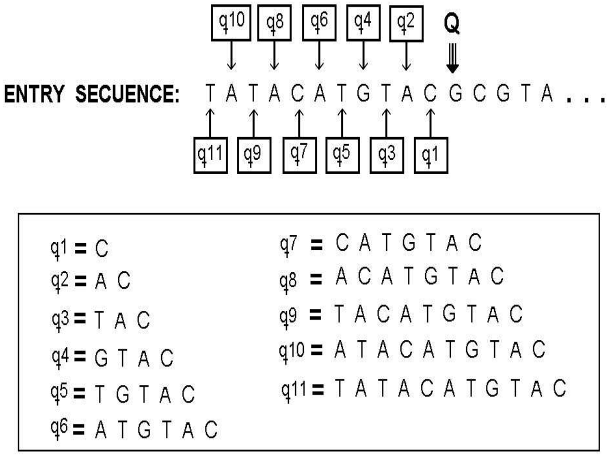

Figure 1. State of the pointers when the twelfth nucleotide is read.

Once the twelfth nucleotide is read, final nodes start to be generated. In this moment, a pointer P is addressed at the beginning of the array of final nodes and Q arrives to the twelfth nucleotide. Then, the first word is recognized and stored in the first final node, N[0]. An internal counter of that node, that we will call F_1, is then increased. The algorithm then proceeds by increasing the value of P, therefore pointing at the following final node, N[1], and moving Q to the next nucleotide. A new 12-bases long word is then recognized and stored in node N[1]. This procedure is repeated for each additional nucleotide (Figure 2).

As we said above, it is significant the fact that each word read from the file that contains the first chromosome generates all the branches of the tree that lead to its final state. A second important point is that, when a particular word has already been found before, the only action is to increase in its corresponding final node, that contains that particular word, the value of F_1. Following these steps, we get to the end of the file that contains the first sequence. At this point, not all possible words have been found. The number of different words found corresponds to the value of the index of pointer P. Figure 3 shows in detail the final nodes, and how they are being filled up when the sequence is read. A significant point is that only those branches of the tree needed to represent the words that are actually present in the file are built. This strategy, that avoids to build a tree containing all possible solutions, generates considerable savings in computing time and computer memory

This process is repeated for the second sequence. How-

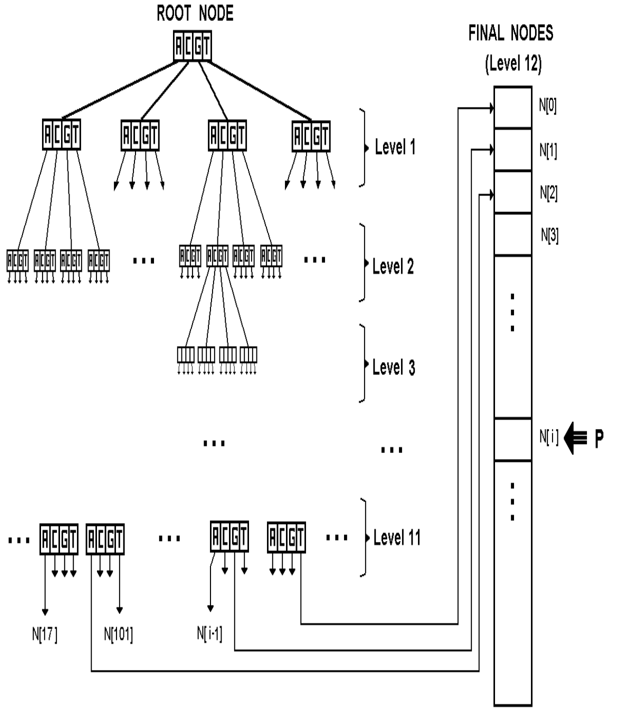

Figure 2. Structure of the dynamic tree of solutions for words of size 12 and generation of the final level nodes.

ever, in this second reading, many of the words correspond to those already found in the first chromosome. In those cases, no new final nodes are created. Simply, the value of a second counter, F_2, that is also associated to each one of the final nodes that already exist (Figure 3), is increased. When new words (i. e. words specific for the second file) are found, additional final nodes are created, with a value of F_1 that is equal to zero. At the end of the process, the index of pointer P reflects the number of different words found in both files together. For every word that has been found, there is a final node in which the value of F_1 gives the number of times that such word appeared in the first chromosome and the value of F_2 gives the number of times that appeared in the second chromosome.

When we want to retrieve information about words that have unequal representations in each of the two chromosomes, we use the values of the counters F_1 and F_2. By comparing them, we are able to select the significant words and to show their frequencies in each chromosome without having to search the whole tree. Moreover, once the two files containing the sequences to compare are read, we can release the memory used to store the tree, because then only the final nodes are relevant.

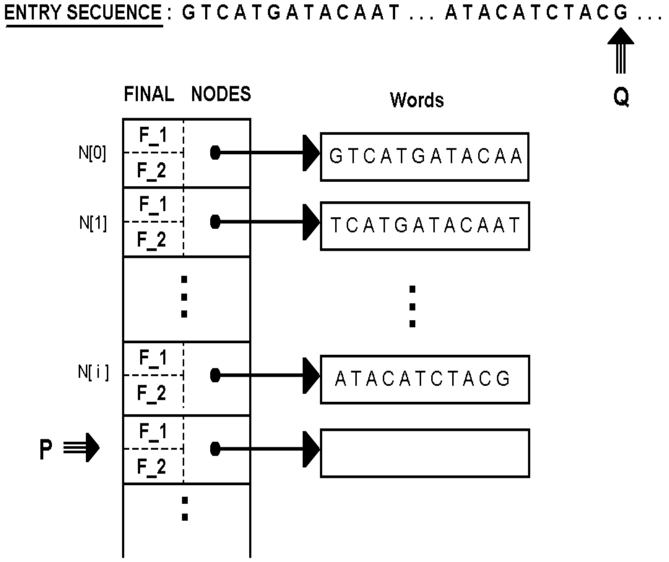

Figure 3. Structure of the final nodes.

The program stores the words according to when they appear. This has two significant consequences. First, it allows to compact the information in the computer memory. Second, it helps to detect long singular words. For example, a word of 20 nucleotides that appears often in a chromosome but only very rarely in another one will be easily detected. The first time it is found, that long word is fragmented by the program into nine words of length 12 that will be consecutively stored. Because of its size, those words are expected to be found by chance only rarely, even in very large sequences (e.g. a word of size 12 is expected to be found, in a chromosome of random sequence and with a length of 33 Mb, only about two times in average). Thus, each one of those nine words has a high probability of being also much more frequent in one chromosome that in the other. They will show in the final files as a consecutive list of nine words that are obviously related in sequence and are strongly biased in their representation, allowing the easy reconstruction of the whole 20-nucleotide-long sequence that is differentially represented in both chromosomes. We have taken advantage of this fact to perform the analysis that are detailed in the next section.

The program that implements this algorithm has been written in C. The use of multiple pointers, one for each level of the tree, required arduous programming, but they offer a very fast speed of generation and reading of the trees.

0-7695-1926-1/03/$17.00 (C) 2003 IEEE

3. Search for Unique Sequences in Human Chromosomes 21 and 22

As a demonstration of the features of our program, we will show here some final data from the exhaustive comparison of human chromosomes 21 and 22. The sequences of these chromosomes were obtained from the National Center for Biotechnology Information (NCBI) web pages (http://www.ncbi.nlm.nih.gov/). We then determined the 12-nucleotide-long words that were chromosome specific by making three independent analyses, needed to consider all possible orientations of the double helices, namely 1) chromosome 21 vs. chromosome 22; 2) chromosome 21 vs. chromosome 22 inverted/complemented; and, 3) chromosome 22 vs. chromosome 21 inverted/complemented. As an example of the results found, the comparison between chromosome 21 and chromosome 22 generated a total of 11457580 words present in at least one of those chromosomes. These are 68.3% of all possible words of length 12. Reading and quantification of the files in this case were achieved at a rate of about 45 Mb/minute. Thus, the final files are generated in a few minutes.

We used two cutoff values, that we called F_SUP and F_INF, to look for sequences that were over represented in one of the chromosomes. The words that are present at a frequency that exceeds F_SUP in one chromosome and is lower than F_INF in the other one can be easily retrieved and listed. As an example, Tables 1, 2 and 3 show all chains in the comparison chromosome 21 vs. chromosome 22 where F_INF(chromosome 21) = 0 and F_SUP(chromosome 22) = 50 (Table 2 and Table 3) or, alternatively, F_INF(chromosome 22) = 0 and F_SUP(chromosome 21) = 50 (Table 1). As we said above, a feature of these tables is that it is often found that consecutive or almost consecutive nodes are detected.

In order to validate whether the program was actually correctly detecting the words that are chromosome-specific, we performed the three searches described above with cutoff values F_INF = 0 and F_SUP = 40 and then we used BLAST searches available at the NCBI web pages (http://www.ncbi.nlm.nih.gov/BLAST/; we used the ”search for nearly exact matches” page and the BLASTN program) in order to find all words detected in the human genome. We found that our results were fully validated, that is, sequences that were found by our method only in one of the two chromosomes were also detected as present in that same chromosome and absent from the other using BLAST searches. This result not only shows that our analysis is correct, but also demonstrates another advantage of using words of such a substantial length. They can be easily checked and interpreted biologically using conventional BLAST searches. Table 4 summarizes the results found, including accession numbers in the NCBI databases, chromosomal positions and biological meaning of the chromosome-specific words. In this table, if possible, multiple consecutive words have been merged. When it is found that but they cannot be merged together, but they still belong to the same gene or repeat, they are written consecutively.

Table 1. Words that are found only in the sequence of chromosome 21 when the direct comparison chromosome 21 vs. chromosome 22 is performed.

No. of Node	Word	Chrom. 21 frequency	Chrom. 22 frequency
2811439	AAGCGCATTCAC	58	0
7124108	GCAGGCGTTTCC	57	0
8389206	AGGCGTTTCCCC	57	0
8389207	GGCGTTTCCCCT	56	0
8824281	GGAAGCGCATTC	51	0
8824282	GAAGCGCATTCA	62	0
9033764	GCGTTTCCCCTT	57	0
9033766	TTACCTGCACCG	56	0
9033767	TACCTGCACCGA	54	0
9033768	CTGCACCGAGCC	54	0
9033787	TCCACGCAGGCG	55	0
9033791	CGCAGGCGTTTC	54	0

Sequences that appear at least 40 times in one chromosome and are absent from the other must be very rare and also very characteristic. In fact, it can be seen in Table 4 that all of them can be classified into two different classes. On one hand, there are several characteristic tandem repeats, that, for some reason are absent in one of the chromosomes (although there are all found in other places in the human genome besides chromosomes 21 or 22, as detected by BLASTN). On the other hand, we detect sequences that belong to several genes with highly repeated structures (e. g. related to mucins, see [17]).

4 Conclusions

The method described in this study allows the exhaustive determination of words of 12 nucleotides in very large sequences and in a very short time. Comparisons between two very large sequences can be performed in a few minutes. Thus, its use may be generalized at the genomic level. These words are large enough as to be easily found using standard searches with the BLASTN program in any of the publicly available databases. This allows further refinement of the results, because it gives information about precise chromosomal location and also, in many cases, functions of the sequences where the particular words are found. All re-

Table 2. Words specific of chromosome 22 (same comparison as in Table 1).

No. of Node	Word	Chrom. 21 frequency	Chrom. 22 frecuency
9139033	CATCATCGAATG	0	81
9139034	ATCATCGAATGG	0	126
9139045	CGAATGGAATCA	0	160
9139053	TCGAATGGAATC	0	196
9139054	GAATCATCATCG	0	73
9139055	AATCATCATCGA	0	80
9139063	AATCGAATGGAA	0	105
9139076	GGAATCATCGAA	0	54
9139103	GAATCATCGAAT	0	55
9139104	AATCATCGAATG	0	62
9139105	CATCGAATGGAA	0	99
9139106	ATCGAATGGAAT	0	197
9139108	GAATGGAATCGA	0	71
9139109	ATGGAATCGAAT	0	94
9139110	TGGAATCGAATG	0	92
9139111	GGAATCGAATGG	0	85
9153783	CAAGCCAGCCAA	0	172
9167410	CAGATACATTGT	0	60
9314281	CTAACGAGGACG	0	71
9314282	TAACGAGGACGC	0	73
9314295	GGCATCGCTAAC	0	56
9314296	GCATCGCTAACG	0	56
9314297	CATCGCTAACGA	0	66
9314298	ATCGCTAACGAG	0	65
9314299	TCGCTAACGAGG	0	139
9314308	CGCCCAGGGCAT	0	59
9314309	CCCAGGGCATCG	0	66
9314310	CCAGGGCATCGC	0	97
9314322	AACGAGGACGCC	0	109
9314323	ACGAGGACGCCG	0	121
9314324	CGAGGACGCCGC	0	82
9314325	AGGACGCCGCCC	0	99
9314326	GGACGCCGCCCA	0	103
9314327	GACGCCGCCCAG	0	66
9314328	ACGCCGCCCAGG	0	64
9314356	GAGGACGCCGTC	0	54
9314357	AGGACGCCGTCC	0	55
9314358	GGACGCCGTCCA	0	54
9314434	CGCTAACGAGGA	0	91
9314557	GCTAACGAGGAC	0	79
9314566	TGAGGACGCTGT	0	90
9415879	GAGGACGCTGTG	0	65
9415900	CGGTGAGGACGC	0	54
9494059	GGCGTCGCTAAC	0	70
9494060	GCGTCGCTAACG	0	69
9494061	CGTCGCTAACGA	0	73
9494062	GTCGCTAACGAG	0	73

Table 3. Continuation of Table 2.

No. of Node	Word	Chro. 21 frequency	Chro. 22 frecuency
9836715	CCTCCATCTGAC	0	68
10137604	CCAACACAGATA	0	72
10245373	CAAAGGATTCCA	0	72
10513310	CAGTCATACTGA	0	56
10783478	AGTCATACTGAC	0	53
11205601	GTAGGTTCCCCT	0	59
11351944	GTCATACTGACT	0	52
11440776	GAACACTGCTAC	0	85

sults presented in this work, and that constitute an exhaustive search for words that are specific for human chromosomes 21 and 22 (over 66 Mb) and the interpretation of their biological meaning, can be obtained in just a few hours.

Among the most general applications of this program are the finding of singular or differentially represented sequences in chromosomes or genomes (as shown here), precise analysis of the number of times some characteristic sequences are present in two molecules or the characterization of the number and types of repeated sequences (e. g. we have performed analyses to detect Alu sequences in these human chromosomes, using characteristic signatures of 12 nucleotides, finding results that are comparable to those described in [11] and [12]). The program is easily adaptable to words of any size below 12, and in fact we already have developed versions for words of six and nine nucleotides. A related program that allows the use of ambiguities (i. e. more than one nucleotide in particular positions) is also available.

Acknowledgements: This paper is supported by CICYT (Grant No. TIC2000-1151-C07-04), by Fundació La Caixa (Grant No. 01/080-00) and Generalitat Valenciana (Grant No. BM-011/2002). It is also part of the NEUROGENOMICA project, supported by an MCYT grant (GEN2001-4851-C06-02).

References

[1] S. Karlin, A. M. Campbell, J. Mrázek. Comparative DNA analysis across diverse genomes. *Annu. Rev. Genet.*, 32:185-225 (1998).

[2] A. J. Gentles, S. Karlin. Genome-scale compositional comparisons in eukaryotes. *Genome Res.*, 11:540-546 (2001).

[3] E. Uberbacher, R. Mural. Locating protein coding regions in human DNA sequences by a multiple sensor-neural network approach. *Proc Natl. Acad. Sci. USA*, 388:11261-11265 (1991).

[4] E. Zinder, G. Stormo. Identification of coding regions in genomic DNA sequences: an application of dynamic programming and neural netwoks. *Nucleic Acids Res.*, 21:607-613 (1993).

[5] V. Solovyev, A. Salamov, C. Lawrence. Predicting internal exons by oligonucleotide composition and discriminant analysis of sliceable open reading frames. *Nucleic Acids Res.*, 22:5156-5163. (1994).

[6] D. Prestridge. Predicting pol II promoter sequences using transcription factor binding sites. *J. Mol. Biol.*, 249:923-932. (1995).

[7] Q. Chen, G. Hertz, G. Stormo. PromFD 1.0: a computer program that predicts eukaryotic pol II promoters using strings and IMD matrices. *Comput. Applic. Biosci.*, 13:29-35 (1997).

[8] A. Brazma, I. Jonassen, J. Vilo, E. Ukkonen. Predicting gene regulatory elemetns in silico on a genomic scale. *Genome Res.*, 8:1202-1215 (1998).

[9] J. van Helden, B. André, J. Collado-Vives. Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. *J. Mol. Biol.*, 281:827-842 (1998).

[10] S. Karlin, J. Mrázek. Compositional differences within and between eukaryotic genomes. *Proc. Natl. Acad. Sci. USA*, 94:10227-10232 (1997).

[11] I. Dunham, N. Shimizu, B. A. Roe, S. Chissoe, et al. The DNA sequence of human chromosome 22. *Nature* , 402:489-495 (1999).

[12] M. Hattori, A. Fujiyama, T. D. Taylor, H. Watanabe et al. The DNA sequence of human chromosome 21. *Nature*, 405:311-319 (2000).

[13] P. Kapranov, S. E. Cawley, J. Drenkow, S. Bekiranov, R. L. Strausberg, S. P. A. Fodor, T. R. Gingeras. Large-scale transcriptional activity in chromosomes 21 and 22. *Science*, 296:916-919 (2002)

[14] Ch. Chen, A. J. Gentles, J. Jurka, S. Karlin. Genes, pseudogenes, and Alu sequence organization across human chromosomes 21 and 22. *Proc. Natl. Acad. Sci. USA*, 99:2930-2935 (2002).

[15] P. Baldi, S. Brunak. *Bioinformatics. The Machine Learning Approach.*, Second Edition. A Bradford Book. The MIT Press. (2001).

[16] R. Durbin, S. R. Eddy, A. Krogh, G. Mitchison. *Biological sequence analysis. Probabilistic models of proteins and nucleic acids*, Cambridge University Press, Cambridge (1998).

[17] J. L. Desseyn, J. P. Aubert, N. Porchet, A. Laine. Evolution of the large secreted gel-forming mucins. *Mol. Biol. Evol.*, 17:1175-1184 (2000).

Table 4. Chromosome-specific sequences, with F_SUP = 40.

SUMMARY SEQUENCE(s)	**Accession numbers**	**Chromosomal location**	**Description of the sequences**	**Additional notes**
GCGGAAGCGCATTC	AP000335	21q22	TANDEM REPEAT	CHROM. 21-specific
TCCACGCAGGCGTTTCCCCTT TTACCTGCACCGA	XM_066238	21q22	LOC128934 Gene similar to zinc finger 347	CHROM. 21-specific
CGAATGGAATCGATGG ATGGAATCGAATGGAA GAATCATCATCGAATGGAAT GAATCATCGAAT	AP000543	22q11	Several related satellite sequences, similar to $(CGAAT)_n$ $(AATAG)_n$	CHROM. 22-specific
CATCGCTAACGAGGACGCCGCCCAGGGCATCGCTAACGAGGACGCCGTCCA GAGGTCGCCGCC CCCACGGCGTCGCTAACGAGGTCGC CAGGGCATCGCTA CCAGGGCGTCGCTAA	XM_092883	22q12	Several sequences that belong to LOC164854 Mucin-like gene	CHROM. 22-specific
TGGGCGGCGTCCT	XM_092877 XM_092883	22q11 22q12	LOC164573 Mucin-like gene LOC164854 Mucin-like gene	CHROM. 22-specific Found in two related genes (LOC164854, LOC164573) that are close in the chromosome but in opposite orientations
TTCCCCTG TGCGT	AL021392	22q13	TANDEM REPEAT	CHROM. 22-specific
GGTTGAAGT CTC	AL078613	Unknown	TANDEM REPEAT	CHROM. 22-specific
GCGGTGAGGACGCTGTG	XN_066267	22q11	LOC128983 Mucin - like gene	CHROM. 22-specific