Technical University of Crete

Electronic & Computer

Engineering Department

Microprocessors & Hardware Laboratory

Diploma Thesis

Rate-based Prefiltering Approach for BLAST Algorithm Acceleration

Panagiotis Afratis

Supervising Professor: Associate Professor Dionisios Pnevmatikatos

Committee: Associate Professor Dionisios Pnevmatikatos
Professor Apostolos Dollas
Assistant Professor Ioannis Papaefstathiou

August 2008
Chania

The last step of a trip,
is the first of the next adventure.

Abstract

DNA sequence comparison and database search has evolved in the last years as a field of strong competition between several, reconfigurable hardware computing groups, attempting to provide performance boosting at this fundamental algorithm of computational biology. In this thesis we present a BLAST preprocessor that efficiently marks the parts of the database that may produce matches. The actual matches over these "high probability" database regions can be determined by running the full BLAST algorithm. Our prefiltering approach offers significant reduction in the size of the database that needs to be fully processed by BLAST, with a corresponding reduction in the run-time of the algorithm. We have implemented our architecture; we evaluate its effectiveness for a variety of databases and queries, and compare its accuracy against the original NCBI BLAST (software) implementation. We find that prefiltering offers at least a factor of 3 and up to 5 orders of magnitude reduction in the database space that needs to be fully searched. Due to its prefiltering nature, our approach can be combined with all major reconfigurable acceleration architectures that have been presented up to date.

Περίληψη

Το σημαντικότερο πρόβλημα της υπολογιστικής μοριακής βιολογίας είναι η σύγκριση γενετικών αλληλουχιών. Γονίδια ή πρωτεΐνες συγκρίνονται με το γονιδίομα ενός ή περισσοτέρων οργανισμών με σκοπό να ελεγχθεί η ύπαρξη τους στους συγκεκριμένους οργανισμούς. Το συγκεκριμένο πρόβλημα είναι υπολογιστικά πολύ ακριβό ενώ οι βάσεις δεδομένων στις οποίες γίνετε η αναζήτηση αυξάνουν με ραγδαίο ρυθμό. Έχουν αναπτυχθεί διάφοροι αλγόριθμοι δυναμικού προγραμματισμού από την δεκαετία του 1970 μέχρι σήμερα για την επίλυση του με γρήγορο τρόπο. Από τις αρχές της δεκαετίας του 1990 επικράτησε ως μη βέλτιστος αλλά σημαντικά αποδοτικότερος ο αλγόριθμος BLAST. Ο αλγόριθμος αυτός συνεχίζει να βελτιστοποιείται από το National Center for Biotechnology Information (NCBI) με κοινή προσπάθεια της ερευνητικής κοινότητας. Το μεγάλο ενδιαφέρον για την εφαρμογή αυτή και ο τεράστιος και συνεχώς αυξανόμενος όγκος δεδομένων οδήγησε την κοινότητα σχεδιασμού συστημάτων βασισμένα σε αναδιατασσόμενη λογική (FPGAs) στην ανάπτυξη αρχιτεκτονικών και υλικού ειδικού σκοπού για τον αλγόριθμο BLAST με έντονο ανταγωνισμό. Οι προτεινόμενες μέχρι σήμερα αρχιτεκτονικές έχουν ως σκοπό την αποτύπωση του αλγόριθμου BLAST με τρόπο αποδοτικό ώστε να γίνετε πιο γρήγορα η επεξεργασία των δεδομένων. Στην εργασία αυτή, παρουσιάζουμε μια νέα προσέγγιση στο πρόβλημα, με μια νέα αρχιτεκτονική ενός προ-επεξεργαστή του αλγορίθμου BLAST, που φιλτράρει τα προς επεξεργασία δεδομένα και επιλέγει ένα μικρό ποσοστό τους για την εφαρμογή του αλγόριθμου. Πιο συγκεκριμένα αυτή η προσέγγιση φιλτραρίσματος προσφέρει μια μείωση των τριών ή μερικές φορές των πέντε τάξεων μεγέθους του διαστήματος αναζήτησης που πρέπει να εξεταστεί και συνεπώς μια επιτάχυνση τριών ή πέντε τάξεων μεγέθους της χρονικής εκτέλεσης του αλγορίθμου. Αυτή η μέθοδος δοκιμάστηκε αναλυτικά και τα αποτελέσματα της ταυτοποιήθηκαν με αυτά που παράγει το λογισμικό του NCBI. Επίσης σχεδιάστηκε, προσομοιώθηκε αναλυτικά και υλοποιήθηκε η αντίστοιχη αρχιτεκτονική. Η συγκεκριμένη προτεινόμενη προσέγγιση – με μικρές αλλαγές υλοποίησης – μπορεί να συνδυαστεί με όλες τις σημαντικές αρχιτεκτονικές αναδιατασσόμενου υλικού που έχουν παρουσιαστεί έως σήμερα ή ακόμα και με συστήματα λογισμικού.

Acknowledgements

Το κείμενο που κρατάτε στα χέρια σας, δεν τεκμηριώνει απλώς την διπλωματική μου εργασία, αλλά αποτελεί το τελευταίο μου βήμα στην πορεία των προπτυχιακών μου σπουδών, γι' αυτό και θα ήθελα να μου επιτραπεί το κομμάτι των ευχαριστιών να γραφτεί στην Ελληνική γλώσσα και σε ένα πιο προσωπικό ύφος.

Αρχικά, θέλω να ευχαριστήσω τον κύριο Πνευματικάτο, επιβλέποντα καθηγητή της εργασίας αυτής, γιατί με τις γνώσεις του, την εμπειρία του και την επίβλεψη του με καθοδήγησε σε όλη τη διάρκεια της εργασίας μου.

Οφείλω βέβαια, να ευχαριστήσω και τους κύριους Δόλλα και Παπαευσταθίου, που ως μέλη της εξεταστικής επιτροπής, συμμετέχουν στην επιτυχή ολοκλήρωση της εργασίας αυτής. Ας μου επιτραπεί να εκφράσω άλλο ένα ευχαριστώ στον κύριο Δολλά, καθώς ως καθηγητής μου σε όλα τα μαθήματα του εργαστηρίου Μικροεπεξεργαστών και Υλικού, δεν μας μετέδωσε μόνο στοιχεία από τις γνώσεις και την εμπειρία του, αλλά μας μεταλαμπάδευσε την όρεξη, την αγάπη και το μεράκι για την γνώση και την δουλειά, μέσα από τις πολύτιμες συμβουλές του.

Θα επιθυμούσα επίσης, να ευχαριστήσω και τον κύριο Σωτηριάδη, διδακτορικό φοιτητή, ο οποίος όχι απλώς συνεργάστηκε μαζί μου για να με καθοδηγήσει στα πλαίσια της εργασίας, αλλά ήταν πάντα διαθέσιμος να με ακούσει και να με στηρίξει στις σκέψεις μου, στις ιδέες μου και στους προβληματισμούς μου.

Η ομάδα του εργαστηρίου Μικροεπεξεργαστών και Υλικού είναι αυτή που δίνει ζωή στο εργαστήριο και αποτελεί το σύνολο εκείνο που θα σε στηρίξει στην καθημερινότητα της δουλειάς σου, τόσο με την βοήθεια και τις συμβουλές της, πολύ περισσότερο όμως με το χαμόγελο της, το καλαμπούρι της, τα αστεία της θα δώσει ένα άλλο τόνο στην προσπάθεια της εργασίας κάνοντας την πιο δημιουργική και ευχάριστη. Για αυτό και θέλω να ευχαριστήσω όλα τα μέλη της ομάδας του εργαστηρίου, όχι μόνο τωρινά, αλλά και παλαιότερα, καθώς ο καθένας τους έβαλε και από ένα λιθαράκι στο ολοκληρωμένο οικοδόμημα της εργασίας αυτής.

Δεν θα μπορούσα να μην εκφράσω ένα μεγάλο ευχαριστώ στους φίλους μου. Πρώτα στους συμφοιτητές μου, εννοώντας όλους εκείνους που ξεκινήσαμε μαζί την πορεία προς το πτυχίο, παρακολουθήσαμε με ζήλο τα μαθήματα, δώσαμε όλες

τις δυνάμεις μας στις εξετάσεις, και ζήσαμε όλη αυτή την πορεία των σπουδών μας. Στους φίλους μου από όλο το Πολυτεχνείο, που απολαύσαμε αυτήν την μοναδική εμπειρίας ζωής των σπουδών μας, στηρίζοντας ο ένας τον άλλον σε δύσκολες και όμορφες στιγμές. Στους φίλους μου από το BEST, Έλληνες και ξένους, για τον πολύτιμο χρόνο και κόπο που δώσαμε στις εκδηλώσεις που διοργανώσαμε και στα ταξίδια που κάναμε μέσα από τα οποία, ο καθένας με το δικό του τρόπο με έκανε να νοιώθω ακόμη πιο ολοκληρωμένος σαν μέλος ενός ευρύτερου κοινωνικού συνόλου.

Τέλος, και πάνω από όλα, ευχαριστώ την οικογένεια μου, τους γονείς μου και την αδερφή μου, που όχι απλά με στήριξαν και μου συμπαραστάθηκαν όλα αυτά τα χρόνια, αλλά γιατί αποτελούν την σταθερή βάση στην πορεία της ζωής μου και σε κάθε μου σκέψη, ιδέα και προσπάθειά μου, εκτός από τα εφόδια που μου δίνουν, είναι δίπλα μου παρέχοντας μου την καλύτερη ασφάλεια και σιγουριά.

Contents

List of Figures

List of Tables

Chapter 1 - Introduction

1.1 Preamble

As we move into the 21st century, we stand at a grand inflection point in biology, how we view and practice biology has forever changed. This inflection point has been catalyzed be number of events, perhaps the most important of which is the human genome project. It provided a genetics parts list and catalyzed the development of high throughput measurement tools and high throughput measurements strategies, as well as stimulating the development of powerful new computational tools for acquiring, storing and analyzing biological information.

The human genome project also changed how we view and practice biology in several other ways. First, it has catalyzed the view that biology is an informational science. Second, biology has become increasingly cross-disciplinary as biologists, chemists, computer scientists, engineers, mathematicians, and physicists work together to develop the high throughput technologies and computational/ mathematical tools required for this new biology – all driven by the contemporary needs of biology. Finally, all of those changes have enabled the emergence of systems biology. Systems approaches have been practiced for many years, but what is unique about today's systems biology is that it can make global measurements and can integrate them from different levels of biological information.

The world of biology is, accordingly, very different from what it was even ten years ago. One of the biggest challenges is to bring an awareness and understanding of the central role that mathematics, computer science and statistics play in deciphering the complexities of this new world of biology. This is done through bioinformatics and computational biology, where bioinformatics refers to the creation and advancement of algorithms, computational and statistical techniques, and theory to solve formal and practical problems arising from the management and analysis of biological data, and on the other hand, computational biology refers to hypothesis-driven investigation of a specific biological problem using computers, carried out with experimental or simulated data, with the primary goal of discovery and the advancement of biological knowledge. The combination and

integration of bioinformatics, computational biology and computer architecture consists the area where this thesis belongs to.

1.2 Contribution of Current Thesis

The main contribution of this thesis is the new approach in the problem of the BLAST algorithm acceleration. This new approach offers a significant reduction in the size of the database that needs to be fully processed by BLAST, with a corresponding reduction in the run-time of the algorithm. According to the results of our prefiltering analysis and after studying carefully the potential of it, we proposed our architectural approach, which was tested analytically with real datasets.

In a brief list the contribution of this thesis is the following:

- New approach on performance boosting of BLAST algorithm execution with search space reduction.
- Software implementation of a BLAST machine for understanding the algorithm in depth.
- Development of various software tools for BLAST algorithm analysis.
- Software implementation of our prefiltering TUC PreBLAST preprocessor based on the prefiltering potential and analysis. This implementation also serves as the verification and profiling tool of the hardware implementation.
- VHDL coding and synthesis, post place and route simulation of TUC PreBLAST architecture.
- Fully automated verification of TUC PreBLAST against NCBI BLAST.
- Evaluation and performance measurements of TUC PreBLAST.

1.3 Thesis Overview

The contents of this thesis are structured as follows:

In chapter 2, the BLAST algorithm is introduced through a brief description of the background of the algorithm, a small example that shows its execution, and the related work and approaches of accelerating him.

In chapter 3, our prefiltering theory potential is analyzed, as well described with all the methods of our approach.

In chapter 4, the hardware design is presented, showing the issues that led us to our architecture choices, and the performance measurements that were taken.
In chapter 5, the conclusion of the thesis presents also the possible ideas for future work.
At the end of this thesis, two appendixes can be found, with the first one having all the results of our experiments and the second giving a small description of the software tools that were implemented.

Chapter 2 – BLAST Algorithm & Acceleration Approaches

2.1 Background and BLAST Algorithm

One of the cornerstones of bioinformatics is the process of comparing sequences to deduce whether the sequences are actually related to one another. Through this type of comparative analysis, one can draw inferences regarding whether two proteins have similar function, contain similar structural motifs, or have a discernible evolutionary relationship. There are pair-wise alignments, where two sequences are directly compared, position by position, to deduce these relationships. Another technique, multiple sequence alignment, is used to identify important features common to three or more sequences.

The generation of all possible alignments between two sequences and the choice of the ones giving the greatest score, consists the most obvious approach of computing the optimal score between these sequences. However, such an approach could produce a too slow algorithm as the number of the alignments grows exponentially with the length of the involved sequences. With the involvement of dynamic programming, Needleman and Wunsch [1] solved that problem in 1970. Instead of determining the similarity of two sequences as a whole, the solution is built up by the similarities between arbitrary prefixes of the two sequences, starting with the ones with the smaller length and continuing with the larger prefixes. A variation of the Needleman-Wunsch algorithm, with the use of dynamic programming, was presented by Smith and Waterman [2] in 1981 for performing faster local alignment.

Although the aforementioned dynamic programming algorithms for computing the similarity and the optimal alignment between two sequences produce acceptable results, their quadratic complexities makes them too slow for searching large databases hence their use is forbidden. To overcome this problem, algorithms that work much faster than the original dynamic programming, had been implemented with the use of heuristic methods.

FASTA was the first program in use, based on heuristic methods, designed for database similarity searching and was developed by Lipman and Pearson [3] [4] .

FASTA enables the user to compare a query sequence against large databases, and various versions of the program are available, shown on Table 1. The FASTA algorithm can be divided into four major steps. In the first step, FASTA determines all overlapping words of certain length in both query sequence and in each of the sequences in the target database, creating two lists in the process. In step two, only the ten best regions for a given pair-wise alignment are considered for further analysis. In step three, FASTA ranks all the concatenated sequences , and then considers further only the best of them in the list. In the fourth and final step, FASTA assesses the significance of the alignments by estimating what the anticipated distribution of scores would be for randomly generated sequences having the same overall composition.

Program	**Query**	**Database**
FASTA	Nucleotide Protein	Nucleotide Protein
FASTX/FASTY	DNA	Protein
TFASTYX/TFASTY	Protein	Translated DNA

Table 1 FASTA Algorithms

By far, the most widely used technique for detecting similarity between sequences of interest is BLAST (short for Basic Local Alignment Search Tool). The ideas in BLAST were developed by Altschul, Gish, Miller, Myers and Lipman in 1990 [5] with purpose to increase the speed of the FASTA program. The widespread adoption of BLAST as a fundamental technique in sequence analysis lies in its ability to detect similarities accurately between nucleotide or protein sequences quickly, without sacrificing sensitivity – the original BLAST paper was the most widely cited paper of the 1990's, with over 10000 citations. The acronym BLAST refers not to a single program but to a family of programs. Each of them is suitable for a different problem domain but they all use the BLAST algorithm. Table 2 presents the name and the description of all the programs that are members of BLAST family.

Program	Query	Database
BLASTN	Nucleotide	Nucleotide
BLASTP	Protein	Protein
BLASTX	Nucleotide, six-frame translation	Protein
TBLASTN	Protein	Nucleotide, six-frame translation
TBLASTX	Nucleotide, six-frame translation	Nucleotide, six-frame translation

Table 2 BLAST Algorithms

Since the initial development of BLAST in 1990, the size of genetic databases has continued to grow at an exponential rate, meaning that improving BLAST performance has remained an important goal. Algorithmic modifications such as MegaBLAST [6] that make further speed/sensitivity tradeoffs have been proposed and are used in certain situations where performance is critical and sensitivity of the search is secondary, however accelerated versions of BLAST that do not sacrifice sensitivity compared to the original BLAST algorithm are more desirable.

2.2 BLAST Overview

BLAST is local alignment method that is capable of detecting not only the best region of local alignment between a query sequence and its target, but also whether there are other plausible alignments between the query and the target. BLAST algorithm consists of 3 steps whose implementation depends on the form of the data processed, nucleotide sequences or amino acid sequences. The nucleotide variant of BLAST, called BLASTn, will be the focus of this thesis, though many of the ideas presented should be applicable to the other variants as well. As we already mentioned, the inputs of BLAST algorithm are a query and a genetic database. The outputs of the algorithm are pairs of the position of match in the database and the query and the associated score, named High Score Pairs (HSP). Every match is a possible HSP depending on its score. The lower bound for HSP is defined by biologists and is dependent on research carried out each time. Although the scoring scheme of the algorithm is based on PAM matrices (Point Accepted

Mutations), we used a simpler scheme where every match corresponds to +5 and every mismatch corresponds to a penalty of -4. BLAST algorithm consists of three steps which depend on the form of the data processed.

During the first step, the query is processed. The method begins by "seeding" the search with small subset of letters from the query sequence, known as w-mer. The result of this process is a list of w-mers, which are contiguous substrings of the query. An example will illustrate better how the query is processed resulting to w-mers. We define the size of w-mer to be 12 letters-long. Consider that the following sequence is a part of the query: ATGCAATATGGCCCGTAT. The corresponding list of w-mers is presented in Figure 1.

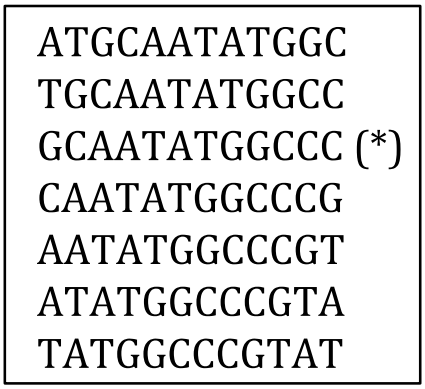

Fig. 1 W-mers list produced during the first step of BLASTn algorithm

In the second step the genetic database is searched for hits. A hit is an exact match between a w-mer and a sequence of letters of the database. Every hit is possible to be part of a High Scoring Pair (HSP). This procedure is illustrated in Figure 2.

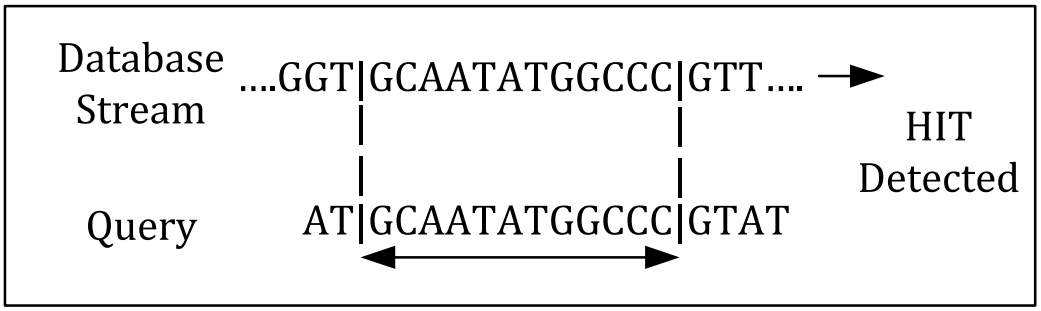

Fig. 2 Second step of BLASTn algorithm; database is searched for hits

In the third step, the list of hits is processed, so that its hit is extended in both directions until its score no longer gets improved under the scoring rules. The process followed during the third step is illustrated in detail in Figure 3.

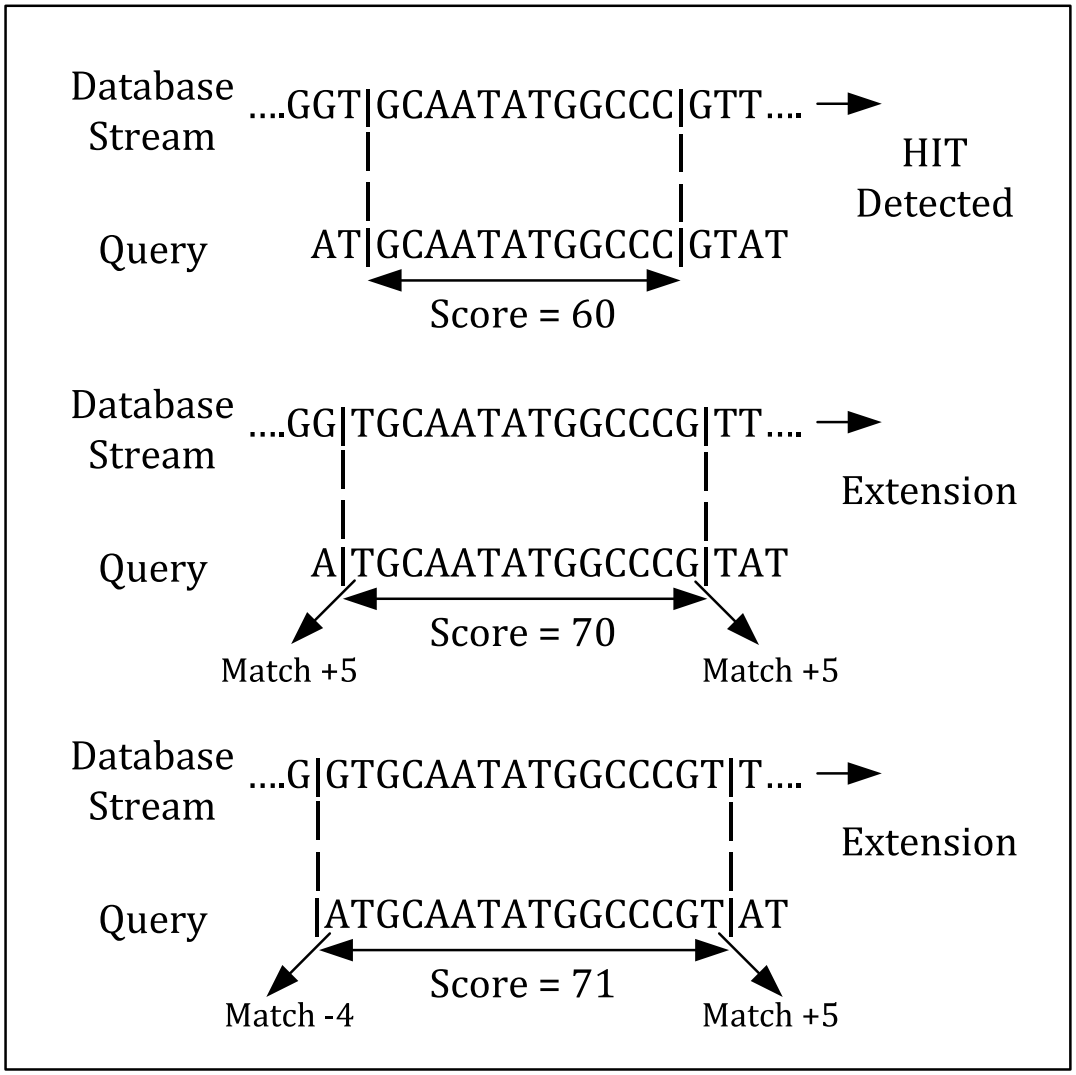

Fig. 3 The hit extension process executed during the third step of BLASTn algorithm

2.3 BLAST Profiling

In [7], the authors profile BLAST running with three different query string lengths (10K, 100K, and 1M bases), finding that step 1 of the BLAST computation takes up an average of about 85% of the total pipeline processing time, while step 2 takes up an average of about 15% of the running time, and the amount of time spent in stage 3 is negligible. The authors of [8], doing profiling on a single query on a small

database with an older version of NCBI BLAST, also concluded that step 1 of the BLAST algorithm accounts for about 80% of the program running time.

While the profiling results in [7] are useful for understanding the performance of BLAST with larger queries, no benchmarks for query sizes under 10000 base-pairs are reported (recall that according to NCBI statistics, over 90% of nucleotide BLAST searches performed through their website are with queries of 2000 base-pairs or less). The results reported in [8] seem to be in agreement with those in [7], however, they are by no means exhaustive, covering only a single query on a single database and using an old version of NCBI BLAST. In any case, all sets of profiling results suggest that step 1 (word matching) should be the first target for hardware acceleration, because of its dominance of the overall running time.

2.4 BLAST Accelerators

The National Center for Biotechnology Information (NCBI) [9] maintains the most widely-used BLAST software implementation (hereafter referred to as NCBI BLAST) and GenBank, the largest collection of all publicly available DNA sequences. While the fundamental BLAST algorithm has undergone little change since the late 1990's, advancements in general-purpose microprocessor technology have provided necessary speed enhancements. However, the exponential growth of sequence data has exposed serious limitations to this strategy. For example, the BLAST server on the NCBI website makes use of a Linux cluster consisting of around 200 CPUs [10]. NCBI reported processing 140,000 queries on a typical weekday in 2004 and planned to double their computing capabilities to keep up with demand. The majority of BLAST accelerators run on a cluster of workstations. A few have been designed to run on FPGA devices.

Faster Search Algorithm BLAST: FSA-BLAST [11] [12] employs software optimization and modifications to the BLAST algorithm. The lookup table in step 1 is replaced by a deterministic finite automaton that is engineered for fast, cache-conscious operation. A semi-gapped extension stage is added between step 2 and 3 to further filter data. Here, a dynamic programming recurrence similar to the ungapped extension stage is used, but with gaps allowed only at every n^{th} residue in the two sequences. Finally, the recurrence of the gapped extension phase is

modified to disallow adjacent gaps in the two sequences, leading to reduced computation per cell. An overall speedup of 20-30% over NCBI BLASTp is reported.

Apple/Genentech BLAST: Apple Computer and Genentech (AG-BLAST) [13] provide an open-source version of NCBI BLAST customized to use Altivec instructions on PowerMac G4 and G5 processors. The modifications are in the seed generation stage. AG-BLAST used with word lengths 20 - 40 provides a two-fold speed increase over MegaBLAST. However, the use of large word lengths makes it unsuitable for searching divergent sequences.

BLAST clusters: The embarrassingly parallel nature of BLAST can be exploited to run on a cluster of nodes. Query segmentation splits the set of query sequences and runs each on individual nodes of a cluster. BLAST searches a subset of the queries against the entire database on each node. This approach provides a near linear scalability if the database can fit in main memory. Alternatively, the database can be segmented, with the same query being processed against different subsets of the database. NCBI-BLAST implements a native multi-threaded search that can take advantage of SMP systems. **M**essage **P**assing **I**nterface BLAST (mpiBLAST) [14] is capable of running on a diverse set of architectures including Beowulf clusters, exhibiting near linear scalability on small numbers of nodes. SGI **H**igh **T**hroughput **C**omputational BLAST (HTC-BLAST) [15] is a distributed cluster implementation of BLAST on SGI Origin 300 servers that enables high-throughput homology searching. A BLASTX comparison of a large number of query sequences against the NR protein database on a 32-processor cluster yields a 30x speedup over a single machine. A commercial offering, TurboBLAST [16], runs on many parallel computing environments including heterogeneous workstations, parallel supercomputers, and grids. Paracel BLAST [17] is designed to run on high-end Sun clusters.

Cluster implementations can significantly decrease turn-around time on high-throughput BLAST searches. Distributed resources can be harnessed to search large queries or databases which would be infeasible on a single node. However,

they scale poorly as more nodes are added to the cluster, since more time is spent formatting databases and collating results. Equal-size database segments on each node need not mean equal workload on the nodes. A large number of homologous sequences in a database segment can cause an imbalance in the load. Clusters typically also have high operational costs when compared to single-node solutions.

FPGA Accelerators:

Rdisk [18] is an FPGA based system to accelerate stage 1 of BLASTN. Reconfigurable logic is attached close to a hard disk, providing on-the-fly filtering capabilities. Rather than using lookup tables, the pattern matching computation is performed between a database word and all query words. This computation can proceed in parallel for all query words and requires processing elements proportional to the size of the query. Rdisk reports a throughput of 60 MCharacters/sec for nucleotide searching.

DeCypherBLAST [19] is a commercial product running on FPGA based engines attached to high-end servers. Scarcity of information on this offering makes a side-by-side comparison impossible.

RC-BLAST [8] is a recent implementation of the BLAST word matching phase on FPGAs. The work illustrates the difficulties faced in accelerating heuristic algorithms on FPGAs. The final FPGA implementation was slower than software version, although this was attributed to the limitations of the technology used by the authors.

BEE2 BLAST [20] is an FPGA reconfigurable platform consisting of three primary components: processing elements, memory elements, and interconnects. On the system level, processing elements are the FPGA chips; memory elements are the external DRAM modules locally attached to each of the FPGA; interconnects consists of local connections, which links local FPGAs on the same PCB board, as well as global connections that link multiple boards into a unified system. The main difference of the BEE2 design from traditional parallel computer system

design is that the processing elements are FPGA chips rather than microprocessors. In addition to the primary components, BEE2 also incorporate a range of secondary system components, including bootstrap, clock distribution, power regulation, and thermal regulation. They support and monitor the primary components to ensure proper operation of the overall system. By combining a cycle-based simulator with an analytical model, they projected the performance capability of the BEE2 platform to be 1 to 2 magnitude order higher than any of the computing systems when run the BLAST algorithm then.

Mercury BLAST [7] [21] [22] involves the use of the Mercury system [23] provides the infrastructure to support high-throughput disk-based computation on reconfigurable hardware attached to general-purpose workstations. Data from disks is streamed directly through pipelined logic blocks, typically being filtered by progressively more complex computations before being sent for post-processing on the attached workstation. Hardware/software code sign is necessary to ensure efficient implementation of an application. The work on the Mercury system targets a two-order-of-magnitude acceleration of the NCBI BLAST algorithm,

FPGA/FLASH [24] [25] combines the use of FPGA components and FLASH memories, allowing a large amount of data to be rapidly accessed and quickly processed. A PCI based system including a 64 GBytes FLASH memory connected to a Xilinx Virtex-2 Pro board was developed, achieving a speed-up of 75 on TBLASTn algorithm.

TreeBLASTP [26] is an FPGA-based accelerator for BLASTP, which accelerates seed generation and un-gapped extension. The seed generation phase is similar to the one-hit approach. High-scoring word matches are detected using dynamic programming (thus eliminating lookup tables), and then passed to un-gapped extension servers. Since two-hit filtering is not performed, larger word lengths and threshold values must be used so as to not overwhelm un-gapped extension. The authors claim a database processing rate of 170 million amino acids per second on query sizes of 1024 residues on the latest FPGA. The effect of decreased sensitivity due to the higher neighborhood threshold must be factored into these results.

TUC-BLAST [27] [28] [29] [30] [31] is an FPGA solution to accelerate DNA searches of small query sequences (1000 bases). The basic computation unit is a hit finder and an extension unit. The former stores a hash table of the query in on-chip block RAMs to detect hits. The extension unit performs ungapped extension to detect significant alignments. High-throughput searching is achieved by replication of the basic computation units.

An extensive literature survey of accelerated sequence analysis applications has established the need for faster solutions, specifically for BLAST. The limited number of BLAST accelerators highlights the difficulties faced in designing a hardware amenable architecture for seed generation.

Chapter 3 – Prefiltering for the BLAST Algorithm

3.1 BLAST Prefiltering Potential

Our main observation that leads to our prefiltering approach is that the BLAST algorithm finds and reports matches in the areas of high similarity between data base and query, i.e. in areas where the third step of the algorithm is active and successfully processes a large number of extensions. These areas with high activity in the third step are also areas where the second step of the algorithm produces multiple hits between different w-mers of the query and different offsets of the database.

Our prefiltering formulates this observation: if within a particular portion (*window*) of the database the high hit rate between the database and the set of w-mers exceeds a *Threshold*, then there is high probability that this area will result to a high similarity (extensions) between data base and query and we need to run the full BLAST algorithm. Portions of the database for which the hit rate does not reach the threshold are not processed further. Notice however that when the hit-rate does exceed the threshold, there is no guarantee that we will actually find a match in this window: multiple hits may be produced from different w-mers in an incorrect order or distance, so they may not correspond to actual extensions. Our approach is depicted in Figures 4 & 5 that plot the hit rate distribution for a window of 100 characters that slides over a streaming database input. Figure 4 shows the hit density for window of 100 characters over the entire chimpanzee's chromosome Y database, and a part of human's chromosome Y as the query. It is clear that there is significant variation in the hit distribution over time, the basis for our pre-filtering technique to work. Figure 5 zooms-in the Figure 1 data at a smaller portion of the database at character positions 6×10^6 up to 6.2×10^6. Figure 5 shows more clearly the spikes in the hit distribution that form at a small subset of database locations. Figure 5 also shows the way our technique will work. Using a threshold, we will select to investigate further only windows with hit value exceeding the threshold. Portions of the database with low hit values are not investigated, saving computations compared to the traditional BLAST approaches.

According to the threshold, different portions of the database are considered interesting. The higher the threshold, the more selective the filtering, but if we exceed a certain threshold value we will miss (some of) the correct BLAST results.

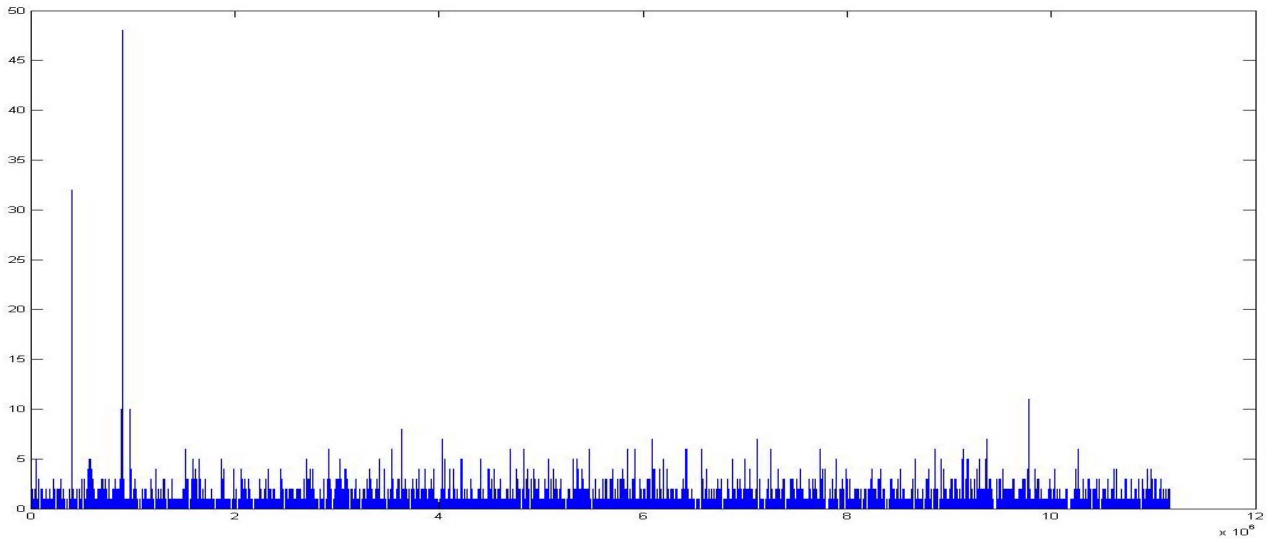

Fig. 4 Hit rate distribution for chimpanzee chromosome Y as database, and a part of human chromosome Y as query, over a window of 100 characters.

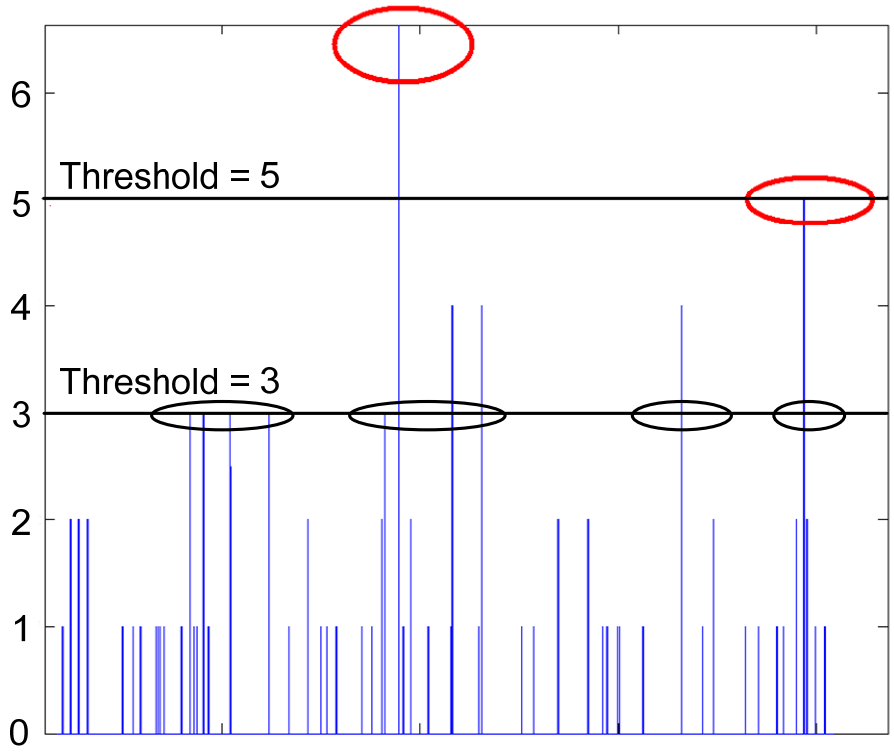

Fig. 5 Hit rate distribution for a window of 100 characters over the streaming database input. The two top circled areas are "of interest" i.e. they result in BLAST matches. The top horizontal line represents the optimal threshold (=5) to identify all these areas. Thresholds less than 5 will produce more candidate regions without identifying more hits (drawn for Threshold=3), while thresholds greater than 5 will miss some of the hits reported by BLAST.

To analyze the potential of BLAST prefiltering, a set of software tools (detailed in Appendix B) was built that implement BLAST searching. We run these tools using several data sets that were provided from NCBI site, and we compare the results against those of the original NCBI BLAST software. In our main experiments parts of Human's chromosomes (Homo Sapiens) (queries) were compared against Chimpanzee's (Pan Troglodytes) genome (database). The data exhibit a high degree of similarity which leads to high hit rate at the second step of the algorithm. A second type of experiments was the comparison of the same parts of Human's chromosomes (Homo Sapiens) (queries) against Mouse's (Mus Musculus) genome (database). The results of those experiments, due to table size, are shown on Appendix A.

Also, the BioPerf [32] benchmark datasets for BLASTn were used in order to achieve an even more acceptable verification; the results of them are shown on Table 3.

BioPerf Input Datasets		**Window % of query**	**Hit Rate %**	**Database Space %**
Database	**Query**			
Escherichia coli (E.coli)	test (Size 573 characters)	10%	0,026%	2,01006%
		20%		2,24051%
		30%		2,43499%
		40%		2,65555%
		50%		2,84138%
		60%		3,08064%
		70%		3,27476%
		80%		3,46452%
		90%		3,65406%
		100%		3,84691%

Table 3 Percentage of database space for BLASTn Bioperf benchmark datasets.

Several metrics were tracked in our simulations, such as the number of the hits from the second step of the algorithm, the distances between the hits, and their distribution. We also collected measurements from the third step of the algorithm, i.e. the final BLAST reported matches, the number of extensions, and their width and distribution, etc.

All the database spaces, which were taken as results of our experiments, were compared with the results of NCBI BLASTn in order to validate that the complete set of solutions were included.

3.2 Prefiltering Window Size

First we investigate the effect of the window size, i.e. the width of the database region in which we measure the hit rate. Figure 6 plots "Space" (i.e. the resulting percentage of the database that we need to process after prefiltering) versus window size: small values are better since they correspond to less input to the full BLAST processing. Since the query size may vary greatly, we express the window size as a percentage of the query length, ranging from 10% up to 100%. Intuitively, larger window sizes will produce more hits shifting the hit rate upwards. The results in Figure 6 lead to two conclusions. First, regarding window size, space is either unaffected or increases as the window size increases; hence a small window is both more effective and sufficient to capture the necessary information. Second, the effectiveness of pre-filtering varies greatly: we find cases where the results are excellent (space in the range of 3% or less of the database), while totally ineffective in other cases (chromosomes 12 and 13) with space = 100% i.e. the entire database is candidate for match. We will address this limitation in section 3.5.

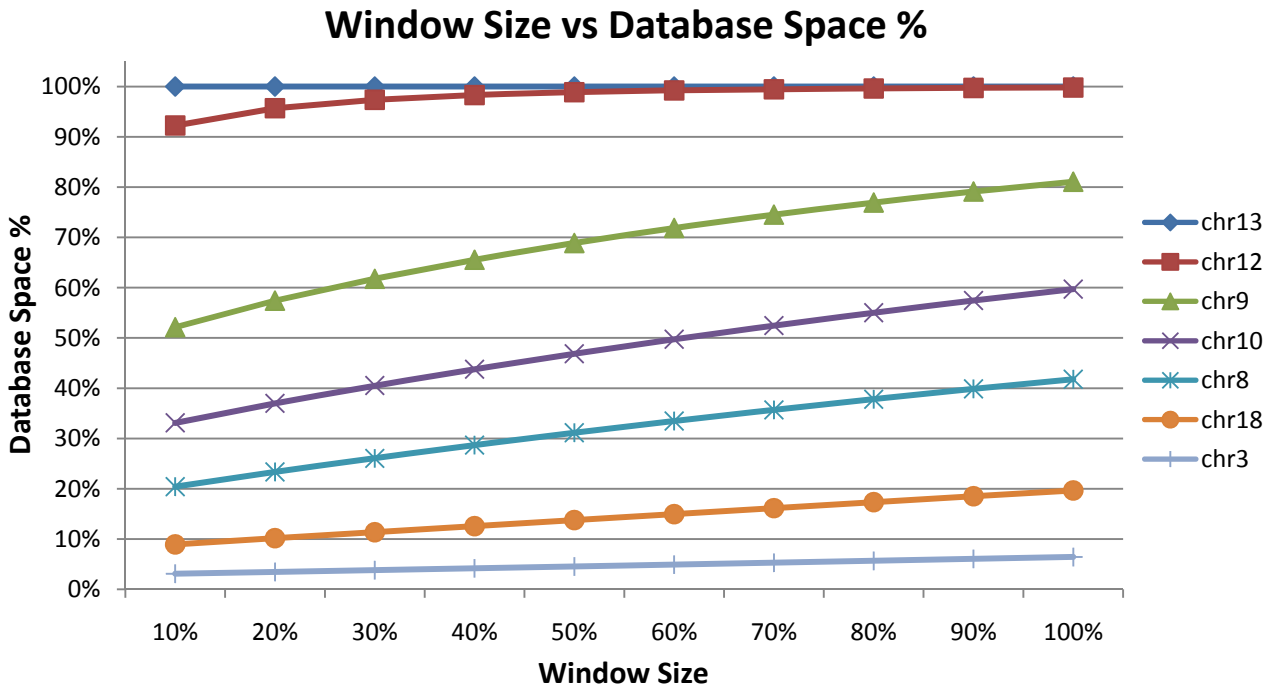

Fig. 6 Database Space % vs. window Size.

3.3 Prefiltering Threshold

The other main prefiltering parameter is the threshold. Figure 7 plots the database space versus a threshold that ranges between two and five. We see that as threshold increases there is a decrease in space, even for some of the "difficult" cases (chromosome 12) identified in the previous paragraph. However, the results for other queries such as chromosome 13 are insensitive to increasing the threshold. Choosing the threshold value is not straightforward. Setting the threshold too low results in larger database space that needs to be processed. Setting the threshold too high we risk ignoring portions of the database that will produce actual hits. In the rest of the thesis we use a threshold value of 2 based on the following observation: for the BLAST algorithm to begin the extension process we need at least one match. Since there will be at least one extension (otherwise the BLAST extension process stops), we will find another hit for a w-mer overlapping with the first. We tested all our results for all our runs and verified that indeed this threshold identifies all the reported NCBI BLAST results. To safely use larger threshold values we need to further investigate and understand the biological significance on the reported results. We believe that setting larger threshold values may omit only the least significant BLAST results while still report the high ranked ones.

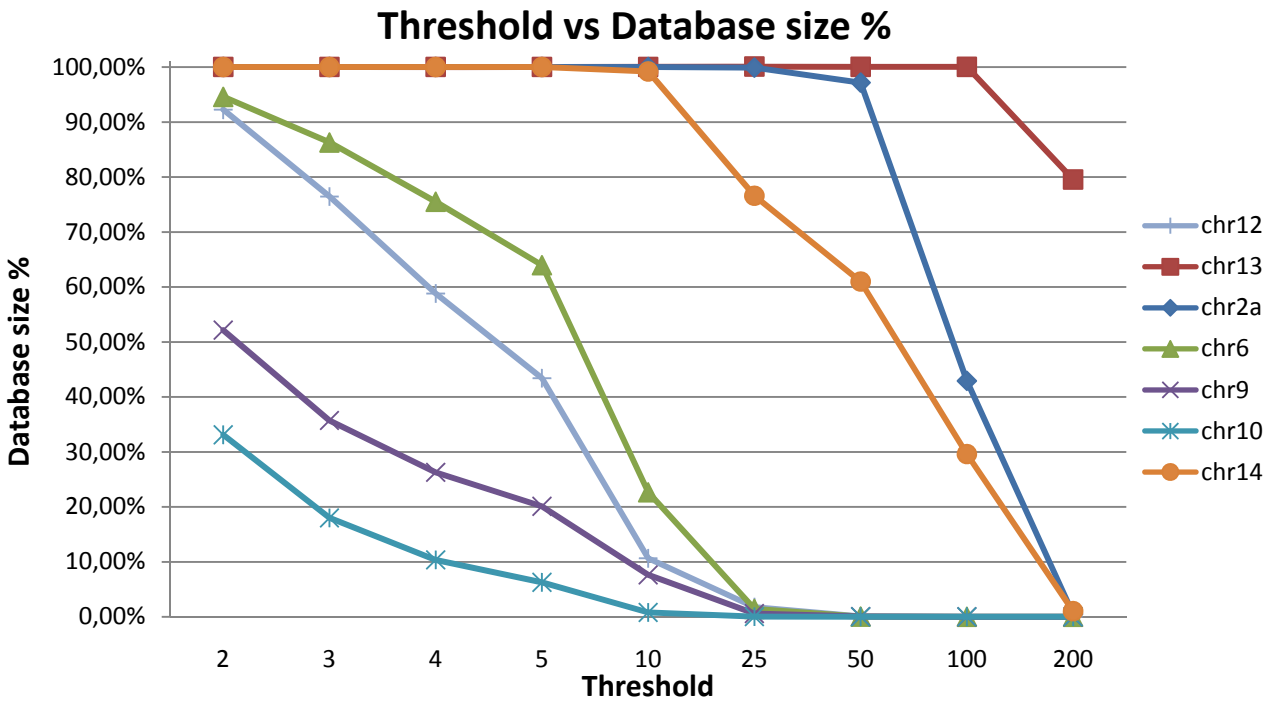

Fig. 7 Database Space % vs. Threshold.

3.4 Sensitivity to Query Size

To understand the behavior of the "difficult" cases such as of the chromosome 12 and 13 queries, we analyzed our results and observed that they all corresponded to very long queries in the order to many thousand characters. In Figure 8 we plot the effect of the query size on the resulting database space that must be searched for the queries that are not amenable to prefiltering. To produce small queries we use a prefix of the original query at a particular size. The trend in Figure 8 is very clear: large queries are not amenable to prefiltering, while small queries show great potential. A possible explanation for this behavior is that a large query contains more distinct w-mers than a smaller one, so the probability of finding multiple hits between the database and any two (or threshold many) of them is larger. Prefiltering for these queries works very well for queries a few hundred character long, and offers no improvement for queries longer than 5 thousand characters.

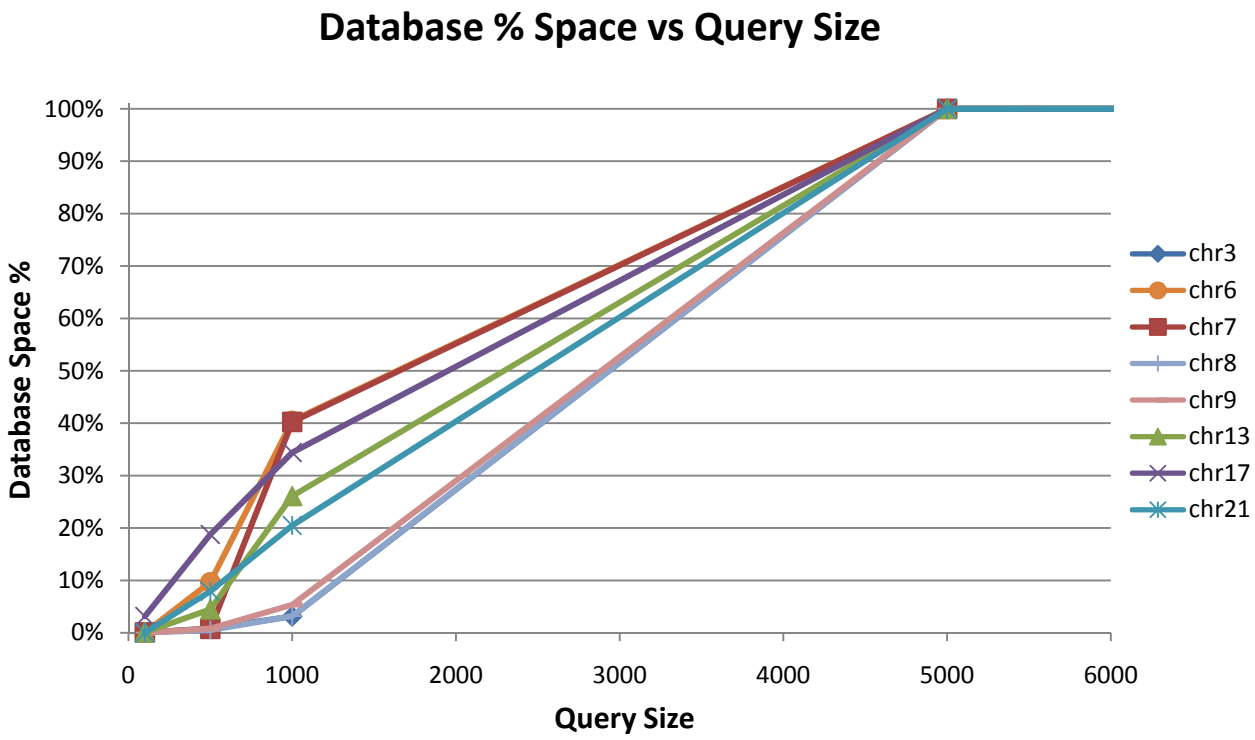

Fig. 8 Database Space % vs. Query size.

3.5 Partitioned Queries

The results from Figure 8 made clear that long queries, while very useful in Biology, cannot be handled effectively by prefiltering. However, the same results offer the solution to the problem: if we partition the query in smaller pieces and processed in parallel, we may achieve operation in the effective prefiltering region. Figure 9 evaluates the partitioning potential. Starting with the original query size, we subdivide it to pieces of one thousand, 500, 250 characters and so on, evaluating the resulting database space that we need to search. As indicated from Figure 8, as the query size becomes smaller, the effectiveness of prefiltering increases. The best results are achieved for small sub-queries less that 250 characters, and for all the difficult queries pre-filtering achieves a 5-fold decrease in the space that needs to be explored (space = 20% of the database). More important is the correlation of query and prefiltering potential: given the database and the query, we can determine the effectiveness of prefiltering, and the need for and extend of partitioning the query.

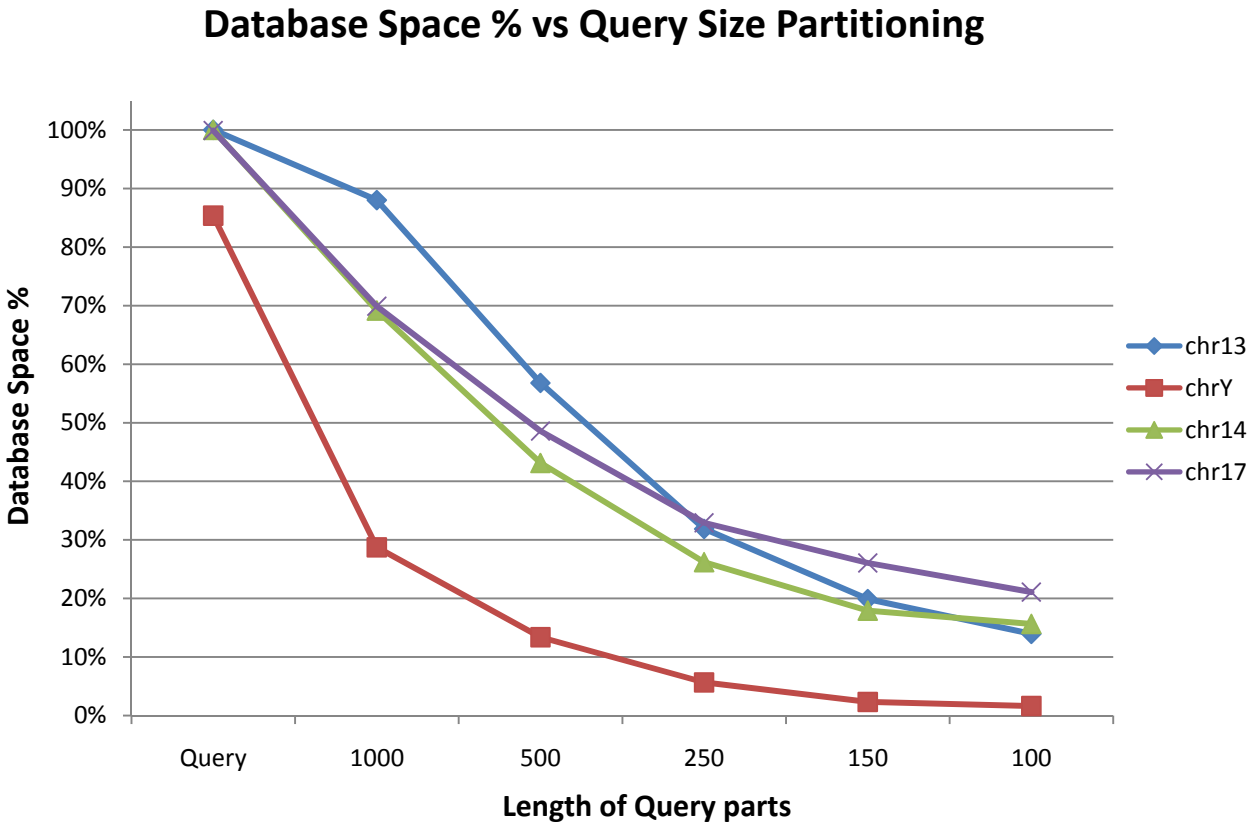

Fig. 9 Query partitioning effect to Database Space.

Chapter 4 – Hardware Design, Implementation and Performance

4.1 Implementation Issues

All the prefiltering analysis properties are based on the number of the hits that are produced on the second step of the algorithm. In order to find hits, comparisons should be performed between every w-mer and the complete database.

Comparisons are 24 bit-wide (12 characters x 2 bits/character) and their number is almost equal to the size of query. For a 1000 characters query 989 w-mers are produced and consequently 989 concurrent comparisons are needed.

There are several implementations proposed for this problem.

(i) The comparisons can be multiplexed in time using one 24 bit comparator or multiplexed in space using for example 989 parallel comparators. This method either takes a lot of time, or consumes a significant number of reconfigurable resources respectively, which is not appropriate for reconfigurable logic based systems [31].

(ii) Another approach is to use a Content Access Memory (CAM) which will have to be too deep (24 bits address) due to w-mer size and hence very expensive in terms of area.

(iii) A memory cache-like scheme could be also used. A single memory cannot be implemented due to its size (24 bits address) that can not fit to any reconfigurable device. Using of memories has the advantage that the size of the designed hardware is proportional to w-mer size which is constant and not to query size which varies.

Due to hardware implementation problems, an alternative method is proposed. In this method we count *probable* hits instead of actual hits. A probable hit is defined as the exact match between the bits 0 to 14 and 3 to 17 and 6 to 20 and 9 to 23, of the examined part of database and the corresponding part of any of the w-mers. That match does not produce necessarily a hit in contrast to the second step of the algorithm. This alternative approach gives slightly worse results but is better suited for FPGA implementation. The 15-bit wide ranges are not chosen arbitrarily: each one of them can occupy an embedded BRAM block to be efficiently

implemented. Xilinx BRAM blocks are available with 32kx1 bit size and need 15 bits for addressing. The BRAM is initialized with 0 in all locations except those that appear in the query w-mer list that are initialized to 1. For example, if the w-mer is 110011001100110011001100, an '1' will be preloaded at positions 110011001100110 of the first memory, 011001100110011 of the second, 001100110011001 of the third, and finally at position 100110011001100 of the fourth memory. This example is illustrated at Figure 10. Hence a simple lookup in the memory identifies if this w-mer portion is a sub-match with some w-mer of the query. Using multiple (4) overlapping BRAMs reduced the probability of reporting false matches. In Table 4 we can see the comparison between the actual and the *probable* hits for several of our experimental datasets.

Database	**Actual Hits**	***Probable* Hits**	**Percentage Actual/*Probable***
chr2A	360435	6364378	5,66%
chr3	19132	33982	56,30%
chr5	656694	751234	87,42%
chr6	109198	1029581	10,61%
chr7	6355	16919	37,56%
chr8	22258	85900	25,91%
chr9	39865	246654	16,16%
chr10	30224	180618	16,73%
chr11	1331477	2708931	49,15%
chr12	97850	437342	22,37%
chr13	656217	11044906	5,94%
chr14	1514674	2839267	53,35%
chr15	720788	1315354	54,80%
chr16	25230	199093	12,67%
chr17	1532071	2460778	62,26%
chr18	8762	16092	54,45%
chr19	464064	522410	88,83%
chr20	669683	1030465	64,99%
chr21	64610	82123	78,67%
chr22	198204	803426	24,67%
chrX	23605	161691	14,60%
chrY	4636	23270	19,92%
ecoli	185	1234	14,99%

Table 4 Actual VS Probable hits

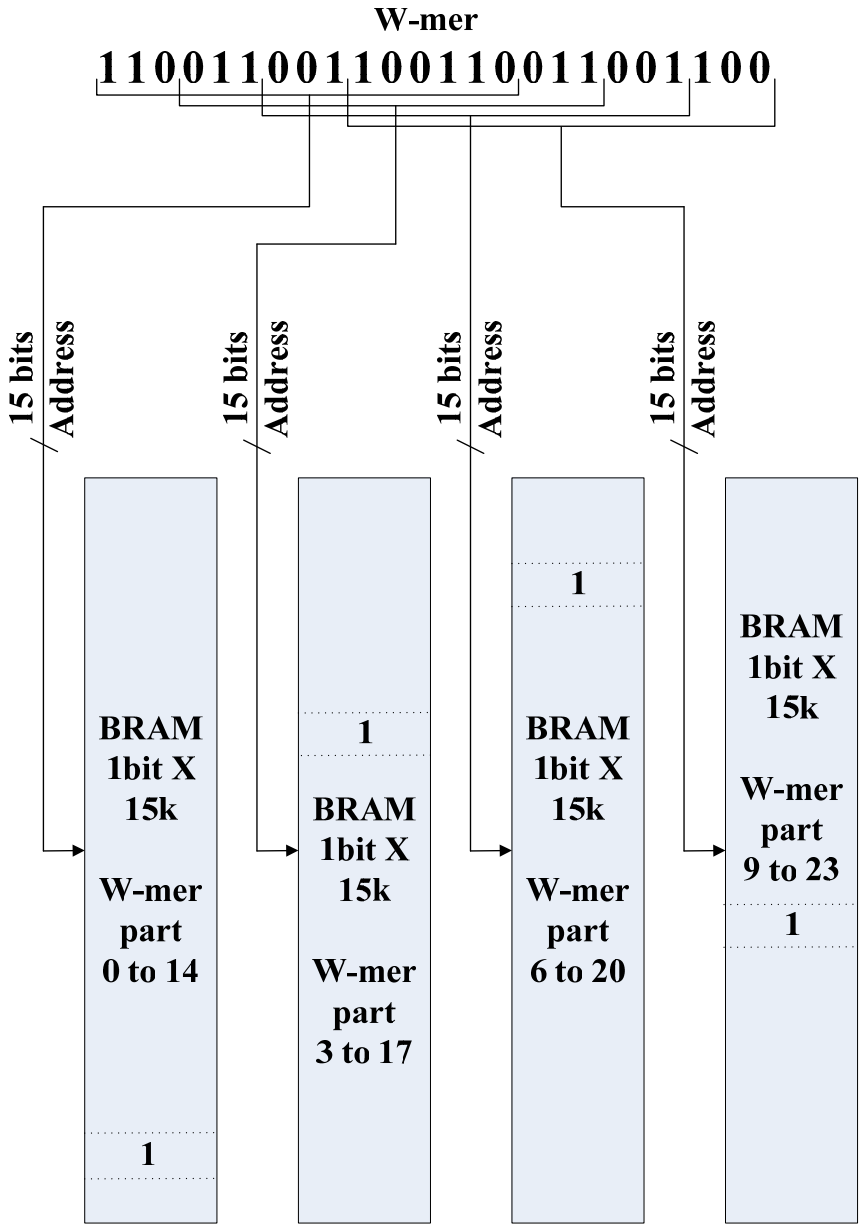

Fig. 10 Illustration of the example of BRAMs preloading

4.2 PreBLAST Architecture

The input of the designed architecture is the database stream. A new character, 2 bits, is processed at every clock cycle. The data path consists of a 12 characters (24 bits) shift register and four blocks of RAM 1 bit x 15K. The shift register gets a new character (2 bits) and addresses the four blocks of RAM at every clock cycle. If the value '1' is stored in all the corresponding 15 bit addresses then a probable hit is detected at the output. Shift register has 24 bit width due to w-mer size which is 12 characters or 24 bits. At every clock cycle when a new character (2 bits) is

inserted shift register perform a shift of two positions. Shift register addresses the four preloaded RAMs with 15 bits each. RAM Data outputs are 1 bit wide and drive an AND gate. If there is '1' at all RAM output at the same clock cycle then the output of AND gate will be '1' and a probable hit will be produced. Two data path designs had been made based on the type of RAMs. The use of dual port RAMs utilizes more efficiently the available number of BRAMs. Figure 11 and Figure 12 show the two data paths that had been designed.

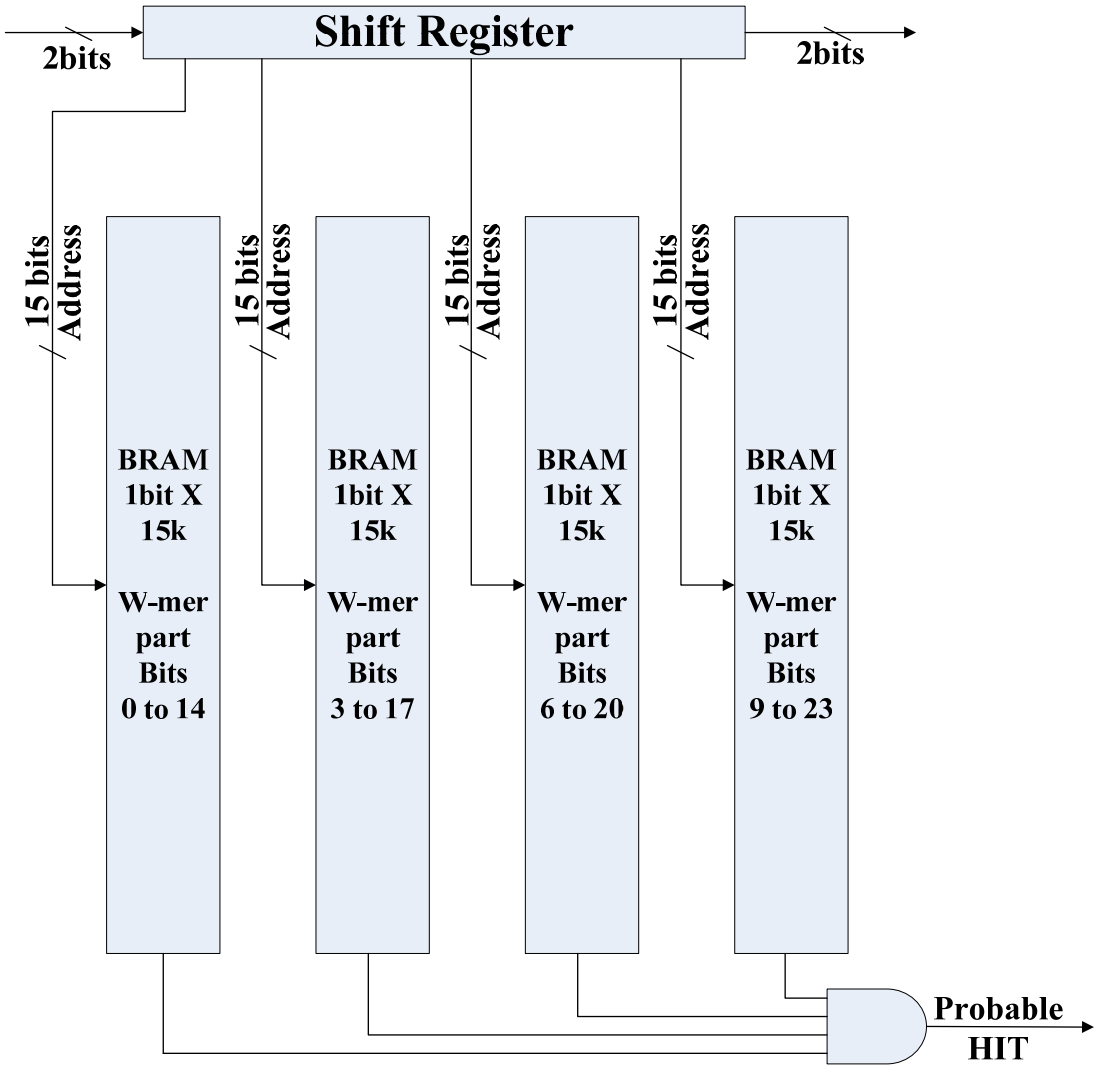

Fig. 11 Data path of the designed system with single port RAMs

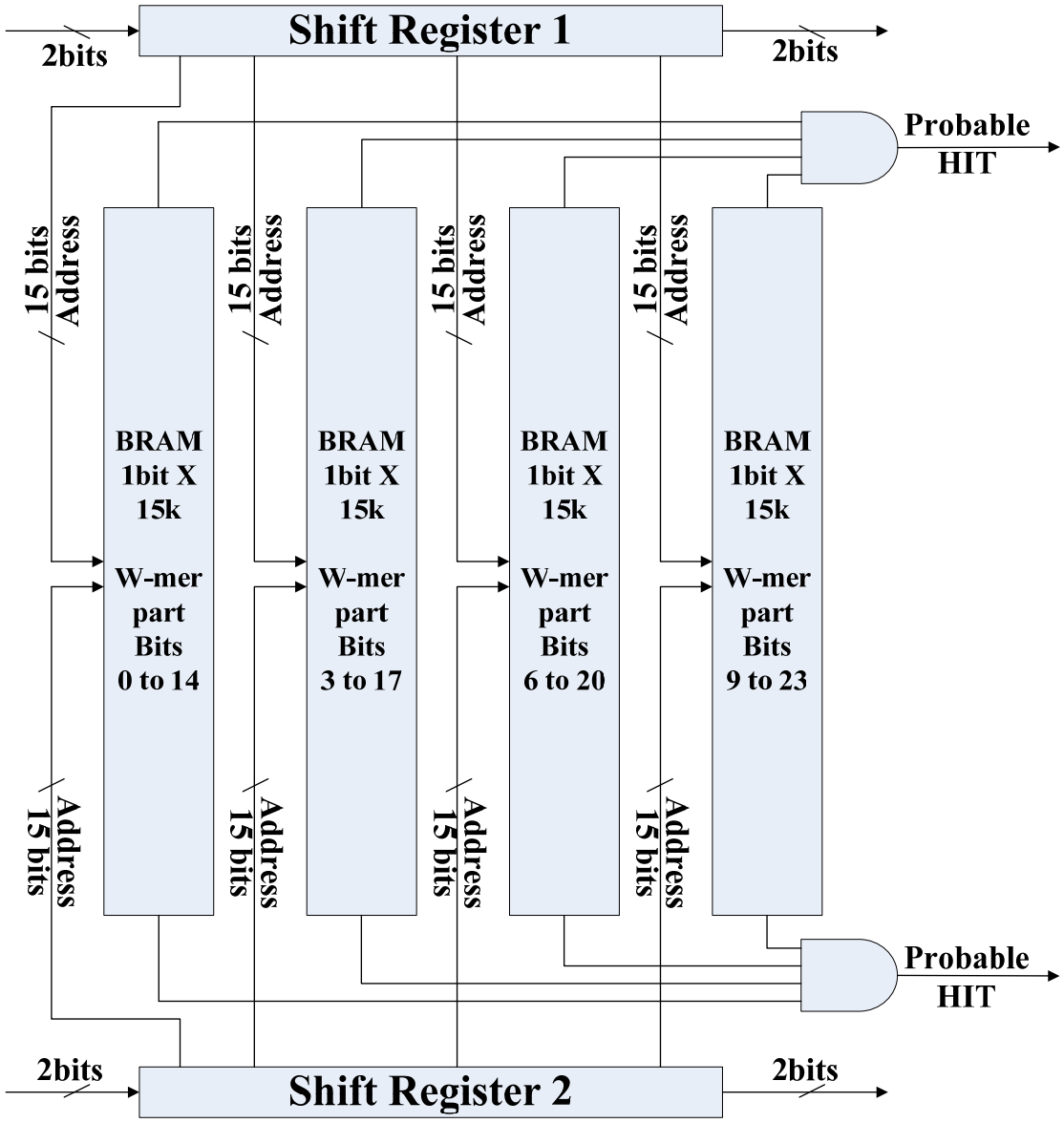

Fig. 12 Data path of the designed system with dual port RAMs

Figure 13 shows the control path of the design which consists of a shift register at window size, an up/down counter, a 32 bit position counter, a control unit and a space memory. At every clock cycle system checks for a new probable hit. The input data is inserted into the shift register and if its value is '1' the up down counter increases. If an '1' shifts out from the shift register, the up/down counter decreases. This design counts the number of the '1' that the shift register holds. The value of the up/down counter is the number of the probable hits that have been detected at the character window examined for the certain database position. Position counter increases every time that a new character from data base stream

is inserted in the design. When the up/down counter value exceeds the predefined threshold, the control unit writes the value of the position counter at the space memory, tagging the start of the area of high similarity. When the up/down counter value falls below threshold value, the control unit writes the value of the position counter at the space memory tagging the end of the area of high similarity. This architecture tags all the areas of high similarity of the data base stream.

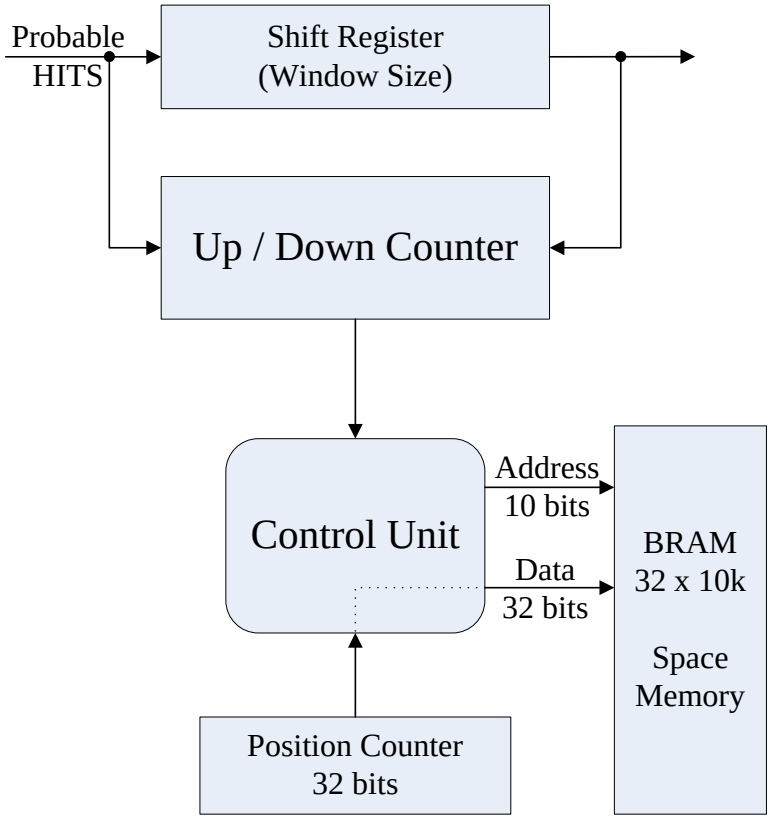

Fig. 13 Control path of the designed system

4.3 Performance Measurements

The design was implemented, full post placed and routed simulated. A Xilinx Virtex 5 family FPGA XC5VLX330T was used for the implementation. Table 5 shows the allocation of the resources for two different implementations with single port RAMs in the data path, for one preprocessor and for 64 parallel preprocessors working in a single chip. Table 6 shows the allocation of the resources for two different implementations with dual port RAMs in the data path, for two preprocessors and for 108 parallel preprocessors working in a single chip. The designs are bounded up to 64 and 108 parallel preprocessors because of the total available BRAMs. On the other hand very few LUTs are used. Table 7 and Table 8

show the corresponding system clock speed and throughput of the two designs. The speed data of version ADVANCED 1.53 was used to measure the clock speed.

Number of Preprocessors	*LUTs/Unit*	*BRAM/Unit*
1	105	5
64	3780	320
Total FPGA Resources	207360	324
Coverage Percentage	1,82%	98,76%

Table 5 Resources allocated to TUC PreBLAST with single port RAMs

Number of Preprocessors	*LUTs/Unit*	*BRAM/Unit*
2	177	6
108	6397	324
Total FPGA Resources	207360	324
Coverage Percentage	3,08%	100%

Table 6 Resources allocated to TUC PreBLAST with dual port RAMs

Number of Preprocessors	*Clock Speed MHz*	*Throughput Characters 10^6*
1	232,13	232,32
64	140,21	8973,64

Table 7 System clock speed and throughput of TUC PreBLAST with single port RAMs

Number of Preprocessors	*Clock Speed MHz*	*Throughput Characters 10^6*
2	204,50	409
108	120,48	13011,84

Table 8 System clock speed and throughput of TUC PreBLAST with dual port RAMs

Performances essentially come from the point that today, FPGA components house a huge potential computational power which can really be exploited if they can be fed at a consequent data rate. The use of TUC PreBLAST attains such filtering rates, as to provide a BLAST processor with the appropriate data in a rate that will effect in an equivalent speed up. In Table 9, is depicted in brief, the search space reduction that we achieved based on our proposed methods with the use of TUC PreBLAST. Chromosome 3 is a typical instance of our experiments, in contrast with the rest 4 chromosomes that consisted the 4 most "hard" cases. Accordingly, the real execution time of a BLAST processor can achieve correspondent speed ups, which are shown in Table 10.

Database chimpanzee chromosome	**Query part of human chromosome**	**Database Space % for total query**	**Database Space % for separated queries**				
			Query, parts of 250 characters				
chr3	(total query) chr3q (Size 965 characters)	3,11632%	0,52070%				
			Query, parts of 1000 characters	Query, parts of 500 characters	Query, parts of 250 characters	Query, parts of 150 characters	Query, parts of 100 characters
chr13	(total query) chr13q (Size 12775 characters)	100,01275 %	88,01238%	56,79287%	31,86639%	19,89180%	13,94842%
chrY	(total query) chrYq (Size 3175 characters)	85,37016%	28,71856%	13,37890%	5,64921%	2,31991%	1,61528%
chr14	(total query) chr14q (Size 5511 characters)	100,01674 %	69,15938%	43,12856%	26,19166%	17,92562%	15,64721%
chr17	(total query) chr17q (Size 3959 characters)	99,90502%	69,89792%	48,57718%	32,92334%	26,06596%	21,07837%

Table 9 Database Search Space reduction for some experiments

Database chimpanzee chromosome	**Query part of human chromosome**	**Speed up for total query**	**Speed up for separated queries**				
			Query, parts of 250 characters				
chr3	(total query) chr3q (Size 965 characters)	32,09	192,05				
			Query, parts of 1000 characters	Query, parts of 500 characters	Query, parts of 250 characters	Query, parts of 150 characters	Query, parts of 100 characters
chr13	(total query) chr13q (Size 12775 characters)	1	1,13	1,76	3,13	5,02	7,16
chrY	(total query) chrYq (Size 3175 characters)	1,17	3,48	7,47	17,70	43,10	61,90
chr14	(total query) chr14q (Size 5511 characters)	1	1,44	2,31	3,81	5,57	6,39
chr17	(total query) chr17q (Size 3959 characters)	1	1,43	2,05	3,03	3,83	4,74

Table 10 Equivalent speed up for some experiments

Chapter 5 – Conclusion & Future Work

In this thesis we exploit a property of the BLAST algorithm to collect simple measurements and filter the database that needs to be considered for a query. We show that BLAST prefiltering offers significant search space reduction that ranges from a factor of 3 for long queries up to 5 orders of magnitude for short queries, and a proportional acceleration to the entire query execution time. Prefiltering is very compact in terms of logic, and requires memory blocks in proportion with the required bandwidth. The filtered database can be subsequently processed with any existing software or hardware BLAST processing system in a streaming fashion.

The new database has different statistical characteristics concerning the hit rate than the initial one. This may affects the execution performance of the system that will process the BLAST algorithm. TUC PreBLAST can be connected as it is shown in Figure 14 with any BLAST processing system (software or hardware). In that case, the throughput of TUC PreBLAST has to be greater or equal than the throughput of BLAST processing system. The BLAST processing system has less computational load due to this architectural arrangement. TUC PreBLAST's output rate is the same with the rate of the input database stream, not at a continuous base but at small time windows. So the BLAST processing system can manipulate data at lower rates, provided that it has an input storage system.

Another arrangement of the system is the existence of multiple TUC PreBLAST systems connected with one BLAST processing element, which will also, needs an input storage system.

Fig. 14 TUC PreBLAST and BLAST processing system arrangement

The investigation of prefiltering can be investigated in several directions. First, alternatives can be evaluated to determine the optimal number and position of bits that address the BRAMs in our implementation. One such alternative is to use hashing in a Bloom [33] filter-like fashion. Also it can be investigated what is the necessary number of BRAMs needed to reduce the false positive results and work closer to the actual BLAST hits.

In this thesis we have considered mainly a small window size of 10% of the query, and a threshold of 2. Ways to dynamically determine these parameters can be investigated, using sampling methods to achieve better filtering without losing accuracy in the results.

Finally as with all BLAST acceleration works there are significant system-level IO issues. Our proposed TUC PreBLAST system demands an aggregate of 13 Giga characters per second or 26 Gbps at maximum performance. While the pin I/O bandwidth is supported in current FPGA devices, sustaining such demands at the system level is still an open issue for the reconfigurable hardware community.

References

[1] Saul Needleman, and Christian Wunsch., "A general method applicable to the search for similarities in the amino acid sequence of two proteins." Journal of Molecular Biology, 1970, Issue 3, Vol. 48.

[2] Temple Smith, and Michael Waterman., "Identification of Common Molecular Subsequences." Journal of Molecular Biology, 1981, Vol. 147, pp. 195-197.

[3] David Lipman, and William Pearson., "Rapid and Sensitive Protein Similarity Searches." Science, 1985, Issue 4693, Vol. 227, pp. 1435-1441.

[4] David Lipman, and William Pearson., "Improved Tools for Biological Sequence Comparison." National Academy of Sciences of the United States of America, 1988, Issue 8, Vol. 85, pp. 2444-2448.

[5] Stephen Altschul, Warren Gish, Webb Miller, Eugene Myers, and David Lipman., "Basic Local Alignment Search Tool." Journal of Molecular Biology, 1990, Issue 3, Vol. 215, pp. 403-410.

[6] Zheng Zhang, Scott Schwartz, Lukas wagner, and Webb Miller., "A greedy algorithm for aligning DNA sequences." Journal of Computational Biology, 2000, Issue 1-2, Vol. 7, pp. 203-214.

[7] Praveen Krishnamurthy, Jeremy Buhler, Roger Chamberlain, Mark Franklin, Kwame Gyang, and Joseph Lancaster., "Biosequence Similarity Search on the Mercury System." 2004. International Conference on Application-specific Systems, Architectures and Processors. pp. 365-375.

[8] Krishna Muriki, Keith D. Underwood, and Ron Sass., "RC-BLAST: Towards a Portable, Cost-Effective Open Source Hardware Implementation." 2005. International Parallel and Distributed Processing Symposium. p. 196b.

[9] National Center for Biotechnology Information., NCBI. [Online] www.ncbi.nlm.nih.gov.

[10] Scott McGinnis, and Thomas Madden., "BLAST: at the core of a powerful and diverse set of sequence analysis tools." Nucleid Acids Reaserch, Web Server issue, 2004, Vol. 32, pp. 20-25.

[11] Michael Cameron, Hugh E. Williams, and Adam Cannane., "Improved gapped alignment in BLAST." IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2004, Issue 3, Vol. 1, pp. 116-129.

[12] Michael Cameron, Hugh E. Williams, and Adam Cannane., "A deterministic finite automaton for faster protein hit detection in BLAST." Journal of Computation Biology, 2006, Issue 4, Vol. 13, pp. 965-978.

[13] Apple/Genentech BLAST., [Online] www.apple.com/acg.

[14] Aaron E. Darling, Lucas Carey, and Wu chun Feng., "The design, implementation, and evaluation of mpiBLAST." 2003. In Proceedings of ClusterWorld 2003.

[15] SGI High Throughput Computational BLAST., [Online] www.sgi.com/industries/sciences/chembio/papers.html#bio.

[16] R.D. Bjornson, A.H. Sherman, S.B. Weston, N. Willard, J. Wing, and Turbogenomics Inc., "TurboBLAST: A parallel implementation of BLAST built on the turbohub." 2002. International Parallel and Distributed Processing Symposium. pp. 183-190.

[17] Paracel BLAST., [Online] www.paracel.com.

[18] Dominique Lavenier, Stephane Guyetant, Steven Derrien, and Stephane Rubini., "A reconfigurable parallel disk system for filtering genomic banks." 2003. International Conference on Engineering of Reconfigurable Systems and Algorithms. pp. 154-166.

[19] Timelogic DeCypher BLAST., [Online] www.timelogic.com.

[20] Chen Chang., BLAST Implementation on BEE2. [Online] www.cs.berkeley.edu/~Eejr/GSI/cs267-s04/final-projects/chenzh/BLAST_implementation_on_BEE2.pdf.

[21] Joseph Lancaster, Jeremy Buhler, and Roger Chamberlain,., "Acceleration of Ungapped Extension in Mercury BLAST." 2005. Workshop on Media and Streaming Processors. pp. 50-57.

[22] Arpith Jacob, Joseph Lancaster, Jeremy Buhler, and Roger Chamberlain., "FPGA-accelerated seed generation in Mercury BLASTP." 2007. International Symposium on Field-Programmable Custom Computing Machines. pp. 95-106.

[23] Roger Chamberlain, Ron Cytron, Mark Franklin, and Ronald Indeck., "The Mercury System: Exploiting Truly Fast Hardware for Data Search." 2003. International Workshop on Storage Network Architecture and Parallel I/Os. pp. 65-72.

[24] Dominique Lavenier, Gilles Georges, and Xinchun Liu., "Seed-based genomic sequence comparison using a FPGA/FLASH Accelerator." 2006. International Conference on Field Programmable Technology. pp. 41-48.

[25] Dominique Lavenier, Gilles Georges, and Xinchun Liu., " A Reconfigurable Index FLASH Memory tailored to Seed-Based Genomic Sequence Comparison Algorithms." The Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology, Special Issue on Computing Architectures and Acceleration for Bioinformatics Algorithms, 2007, Issue 3, Vol. 48, pp. 255-269.

[26] Martin Herbordt, Tom VanCourt, Yongfeng Gu, Josh Model, and Bharat Sukhwani., "Single pass approximate string matching on FPGAs." 2006. International Field-Programmable Custom Computing Machines. pp. 217 - 226.

[27] Euripides Sotiriades, Christos Kozanitis, and Apostolos Dollas., "Some initial results on hardware BLAST acceleration with a reconfigurable architecture." 2006. International Parallel and Distributed Processing Symposium. p. 251.

[28] Euripides Sotiriades, Christos Kozanitis, and Apostolos Dollas., "FPGA based architecture for DNA sequence comparison and database search." 2006. International Parallel and Distributed Processing Symposium. p. 186.

[29] Euripides Sotiriades, Christos Kozanitis, Grigorios Chrysos, and Apostolos Dollas., "Rapid Phototyping of a System-on-a-Chip for the BLAST Algorithm Implementation." 2006. International Workshop on Rapid System Prototyping. pp. 223-229.

[30] Euripides Sotiriades, and Apostolos Dollas., "Design Space Exploration for the BLAST Algorithm Implementation." 2007. International Symposium on Field-Programmable Custom Computing Machines. pp. 323-326.

[31] Euripides Sotiriades, and Apostolos Dollas., "A General Reconfigurable Architecture for the BLAST algorithm." The Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology, Special Issue on Computing Architectures and Acceleration for Bioinformatics Algorithms, 2007, Issue 3, Vol. 48, pp. 189 – 208.

[32] D.A. Bader, Yue Li, Tao Li, and V. Sachdeva., "BioPerf: a benchmark suite to evaluate high-performance computer architecture on bioinformatics applications." 2005. IEEE Workload Characterization Symposium. pp. 163-173.

[33] Burton H. Bloom., "Space/Time trade-offs in hash coding with allowable errors." Communications of the ACM, 1970, Issue 7, Vol. 13, pp. 422-426.

Appendix A – Experimental Results

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2	Database Space % for threshold 3	Database Space % for threshold 4	Database Space % for threshold 5
Chr2 (Size 111497100 characters)	Chr2q (Size 8079 characters)	10%	5,708%	99,79117%	100,00709%	100,00708%	100,00707%
		20%		99,99551%	100,00709%	100,00708%	100,00707%
		30%		100,00202%	100,00709%	100,00708%	100,00707%
		40%		100,00275%	100,00709%	100,00708%	100,00707%
		50%		100,00347%	100,00709%	100,00708%	100,00707%
		60%		100,00420%	100,00709%	100,00708%	100,00707%
		70%		100,00492%	100,00709%	100,00708%	100,00707%
		80%		100,00565%	100,00709%	100,00708%	100,00707%
		90%		100,00637%	100,00709%	100,00708%	100,00707%
		100%		100,00710%	100,00709%	100,00708%	100,00707%

Table 11 Chimpanzee VS Human Chromosome 2, thresholds 2,3,4,5

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 10	Database Space % for threshold 25	Database Space % for threshold 50	Database Space % for threshold 100
Chr2 (Size 111497100 characters)	Chr2q (Size 8079 characters)	10%	5,708%	100,00706%	99,89465%	97,16463%	42,89804%
		20%		100,00706%	100,00039%	99,57486%	90,69474%
		30%		100,00706%	100,00642%	99,86911%	97,07084%
		40%		100,00706%	100,00673%	99,96645%	98,87351%
		50%		100,00706%	100,00673%	99,99421%	99,53920%
		60%		100,00706%	100,00673%	100,00483%	99,77784%
		70%		100,00706%	100,00673%	100,00634%	99,89909%
		80%		100,00706%	100,00673%	100,00634%	99,95259%
		90%		100,00706%	100,00673%	100,00634%	99,97861%
		100%		100,00706%	100,00673%	100,00634%	99,99306%

Table 12 Chimpanzee VS Human Chromosome 2, thresholds 10,25,50,100

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2	Database Space % for threshold 3	Database Space % for threshold 4	Database Space % for threshold 5
Chr3 (Size 202464459 characters)	Chr3q (Size 965 characters)	10%	0,017%	0,56886%	0,93604%	0,28160%	0,09276%
		20%		1,16451%	1,05831%	0,32442%	0,10911%
		30%		1,77051%	1,18502%	0,36696%	0,13055%
		40%		2,39901%	1,32396%	0,41836%	0,14981%
		50%		3,03719%	1,45569%	0,46693%	0,16724%
		60%		3,69490%	1,59651%	0,51922%	0,18511%
		70%		4,36180%	1,75020%	0,57929%	0,20399%
		80%		5,04421%	1,90403%	0,63337%	0,22821%
		90%		5,74385%	2,05289%	0,69059%	0,25151%
		100%		6,44482%	2,20486%	0,75341%	0,27734%

Table 13 Chimpanzee VS Human Chromosome 3, thresholds 2,3,4,5

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 10	Database Space % for threshold 25	Database Space % for threshold 50	Database Space % for threshold 100
Chr3 (Size 202464459 characters)	Chr3q (Size 965 characters)	10%	0,017%	0,00464%	0,00452%	0,00373%	0,00000%
		20%		0,00537%	0,00476%	0,00403%	0,00091%
		30%		0,00589%	0,00499%	0,00422%	0,00096%
		40%		0,00618%	0,00523%	0,00442%	0,00100%
		50%		0,00647%	0,00547%	0,00460%	0,00105%
		60%		0,00675%	0,00571%	0,00480%	0,00110%
		70%		0,00704%	0,00647%	0,00499%	0,00194%
		80%		0,00733%	0,00675%	0,00518%	0,00231%
		90%		0,00761%	0,00704%	0,00537%	0,00240%
		100%		0,00790%	0,00733%	0,00556%	0,00250%

Table 14 Chimpanzee VS Human Chromosome 3, thresholds 10,25,50,100

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2	Database Space % for threshold 3	Database Space % for threshold 4	Database Space % for threshold 5
Chr5 (Size 182067534 characters)	Chr5q (Size 1776 characters)	10%	0,413%	12,08713%	33,27838%	28,30759%	25,71133%
		20%		20,92470%	36,49611%	30,96001%	27,99713%
		30%		29,08665%	39,44959%	33,42132%	30,11649%
		40%		36,60987%	42,27456%	35,80807%	32,23415%
		50%		43,51047%	45,07359%	38,20479%	34,25408%
		60%		49,69416%	47,72435%	40,48027%	36,25870%
		70%		55,32630%	50,30633%	42,83528%	38,22168%
		80%		60,35552%	52,70858%	45,01849%	40,16186%
		90%		64,89509%	55,03252%	47,14292%	42,06409%
		100%		68,96098%	57,28893%	49,21493%	43,90987%

Table 15 Chimpanzee VS Human Chromosome 5, thresholds 2,3,4,5

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 10	Database Space % for threshold 25	Database Space % for threshold 50	Database Space % for threshold 100
Chr5 (Size 182067534 characters)	Chr5q (Size 1776 characters)	10%	0,413%	19,85300%	7,60146%	0,14444%	0,00100%
		20%		21,94841%	11,60463%	1,59910%	0,00115%
		30%		23,54969%	12,88965%	2,33779%	0,01275%
		40%		25,11762%	14,11504%	2,99677%	0,04415%
		50%		26,63341%	15,30237%	3,58871%	0,08750%
		60%		28,10802%	16,42697%	4,17986%	0,15846%
		70%		29,57054%	17,56749%	4,76626%	0,23978%
		80%		30,97229%	18,65645%	5,42425%	0,34520%
		90%		32,37313%	19,74981%	6,05600%	0,45107%
		100%		33,74760%	20,83996%	6,67312%	0,56570%

Table 16 Chimpanzee VS Human Chromosome 5, thresholds 10,25,50,100

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2	Database Space % for threshold 3	Database Space % for threshold 4	Database Space % for threshold 5
Chr6 (Size 177555873 characters)	Chr6q (Size 2434 characters)	10%	0,580%	48,73887%	86,28147%	75,49835%	63,95943%
		20%		74,08333%	91,37351%	84,14880%	75,68035%
		30%		86,55467%	94,12591%	89,04922%	82,66914%
		40%		92,74900%	95,80311%	92,04285%	87,17840%
		50%		95,94914%	96,90631%	94,07035%	90,25854%
		60%		97,66489%	97,65796%	95,48940%	92,51560%
		70%		98,60918%	98,19698%	96,52904%	94,13659%
		80%		99,15208%	98,59148%	97,30346%	95,38782%
		90%		99,46949%	98,89132%	97,85350%	96,35722%
		100%		99,65802%	99,11798%	98,28950%	97,06925%

Table 17 Chimpanzee VS Human Chromosome 6, thresholds 2,3,4,5

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 10	Database Space % for threshold 25	Database Space % for threshold 50	Database Space % for threshold 100
Chr6 (Size 177555873 characters)	Chr6q (Size 2434 characters)	10%	0,580%	22,64196%	1,42127%	0,02934%	0,00342%
		20%		35,65681%	2,73032%	0,07139%	0,00356%
		30%		46,57498%	4,75275%	0,14805%	0,00370%
		40%		55,63832%	7,45859%	0,28872%	0,00384%
		50%		63,03328%	10,71703%	0,49006%	0,00399%
		60%		69,22076%	14,46121%	0,74515%	0,00413%
		70%		74,24727%	18,37403%	1,10984%	0,00427%
		80%		78,48050%	22,73721%	1,59180%	0,00867%
		90%		81,94488%	27,01496%	2,25449%	0,01257%
		100%		84,80291%	31,43938%	3,01522%	0,01991%

Table 18 Chimpanzee VS Human Chromosome 6, thresholds 10,25,50,100

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2	Database Space % for threshold 3	Database Space % for threshold 4	Database Space % for threshold 5
Chr7 (Size 162359053 characters)	Chr7q (Size 700 characters)	10%	0,010%	1,46995%	0,52218%	0,23878%	0,10572%
		20%		1,63786%	0,59140%	0,27791%	0,12283%
		30%		1,79685%	0,66366%	0,31098%	0,13854%
		40%		1,95355%	0,73377%	0,34614%	0,16016%
		50%		2,10369%	0,80261%	0,37804%	0,17563%
		60%		2,26206%	0,86790%	0,40775%	0,18947%
		70%		2,41918%	0,93529%	0,43996%	0,20463%
		80%		2,57780%	1,00588%	0,47626%	0,22206%
		90%		2,74586%	1,07591%	0,51011%	0,24126%
		100%		2,90275%	1,14495%	0,54440%	0,25791%

Table 19 Chimpanzee VS Human Chromosome 7, thresholds 2,3,4,5

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 10	Database Space % for threshold 25	Database Space % for threshold 50	Database Space % for threshold 100
Chr7 (Size 162359053 characters)	Chr7q (Size 700 characters)	10%	0,010%	0,00562%	0,00139%	0,00087%	0,00000%
		20%		0,00709%	0,00193%	0,00092%	0,00080%
		30%		0,00896%	0,00206%	0,00096%	0,00087%
		40%		0,00981%	0,00219%	0,00101%	0,00091%
		50%		0,01005%	0,00277%	0,00105%	0,00095%
		60%		0,01126%	0,00204%	0,00155%	0,00100%
		70%		0,01194%	0,00212%	0,00165%	0,00104%
		80%		0,01380%	0,00221%	0,00173%	0,00108%
		90%		0,01597%	0,00229%	0,00182%	0,00113%
		100%		0,01684%	0,00238%	0,00191%	0,00117%

Table 20 Chimpanzee VS Human Chromosome 7, thresholds 10,25,50,100

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2	Database Space % for threshold 3	Database Space % for threshold 4	Database Space % for threshold 5
Chr8 (Size 148638763 characters)	Chr8q (Size 1780 characters)	10%	0,058%	0,01401%	8,88965%	3,73705%	1,73178%
		20%		0,01531%	11,06658%	5,09439%	2,51629%
		30%		0,01628%	12,96321%	6,36724%	3,27681%
		40%		0,01849%	14,86652%	7,63375%	4,14873%
		50%		0,01957%	16,77179%	8,92049%	4,96908%
		60%		0,02065%	18,55964%	10,20215%	5,80738%
		70%		0,02173%	20,31498%	11,48185%	6,64326%
		80%		0,02281%	22,13715%	12,76331%	7,55417%
		90%		0,02516%	23,87049%	14,07677%	8,48957%
		100%		0,02636%	25,58314%	15,38733%	9,38510%

Table 21 Chimpanzee VS Human Chromosome 8, thresholds 2,3,4,5

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 10	Database Space % for threshold 25	Database Space % for threshold 50	Database Space % for threshold 100
Chr8 (Size 148638763 characters)	Chr8q (Size 1780 characters)	10%	0,058%	0,09134%	0,02622%	0,01914%	0,01401%
		20%		0,14524%	0,02935%	0,02047%	0,01531%
		30%		0,19584%	0,03257%	0,02179%	0,01628%
		40%		0,25860%	0,03593%	0,02313%	0,01849%
		50%		0,32316%	0,04064%	0,02445%	0,01957%
		60%		0,42267%	0,04500%	0,02577%	0,02065%
		70%		0,52176%	0,05308%	0,02709%	0,02173%
		80%		0,61095%	0,05852%	0,02843%	0,02281%
		90%		0,75053%	0,06190%	0,02975%	0,02516%
		100%		0,90001%	0,06506%	0,03107%	0,02636%

Table 22 Chimpanzee VS Human Chromosome 8, thresholds 10,25,50,100

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2	Database Space % for threshold 3	Database Space % for threshold 4	Database Space % for threshold 5
Chr9 (Size 120061799 characters)	Chr9q (Size 1865 characters)	10%	0,205%	52,14174%	35,70315%	26,27202%	20,09645%
		20%		57,42680%	41,11884%	30,91144%	23,89371%
		30%		61,77050%	45,82078%	35,11153%	27,55357%
		40%		65,55173%	50,01162%	39,00396%	30,98112%
		50%		68,86724%	53,84642%	42,63471%	34,33123%
		60%		71,86312%	57,30924%	46,09915%	37,51657%
		70%		74,52124%	60,50824%	49,37307%	40,53543%
		80%		76,92847%	63,45718%	52,39454%	43,49823%
		90%		79,10944%	66,23881%	55,38659%	46,32537%
		100%		81,06245%	68,77914%	58,11400%	49,05627%

Table 23 Chimpanzee VS Human Chromosome 9, thresholds 2,3,4,5

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 10	Database Space % for threshold 25	Database Space % for threshold 50	Database Space % for threshold 100
Chr9 (Size 120061799 characters)	Chr9q (Size 1865 characters)	10%	0,205%	7,61398%	0,58820%	0,04430%	0,00332%
		20%		9,14115%	0,78465%	0,06048%	0,01041%
		30%		10,68664%	0,96372%	0,07056%	0,01303%
		40%		12,32053%	1,20165%	0,08808%	0,01582%
		50%		13,98859%	1,43010%	0,10389%	0,01924%
		60%		15,72245%	1,68350%	0,11823%	0,02065%
		70%		17,52834%	1,94172%	0,13965%	0,02205%
		80%		19,37556%	2,24474%	0,16118%	0,02345%
		90%		21,16511%	2,54717%	0,17993%	0,02649%
		100%		23,02853%	2,92572%	0,20209%	0,02812%

Table 24 Chimpanzee VS Human Chromosome 9, thresholds 10,25,50,100

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2	Database Space % for threshold 3	Database Space % for threshold 4	Database Space % for threshold 5
Chr10 (Size 137441083 characters)	Chr10q (Size 1270 characters)	10%	0,131%	33,11310%	17,98532%	10,35217%	6,26894%
		20%		36,99100%	21,42048%	12,79046%	7,98756%
		30%		40,53500%	24,62980%	15,27329%	9,72599%
		40%		43,76999%	27,58035%	17,62341%	11,54304%
		50%		46,85358%	30,39318%	19,91120%	13,26481%
		60%		49,72797%	33,13655%	22,17431%	15,01556%
		70%		52,46424%	35,81592%	24,38500%	16,79329%
		80%		55,02326%	38,36060%	26,56434%	18,57778%
		90%		57,43763%	40,81492%	28,73264%	20,38455%
		100%		59,70814%	43,17155%	30,85577%	22,16591%

Table 25 Chimpanzee VS Human Chromosome 10, thresholds 2,3,4,5

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 10	Database Space % for threshold 25	Database Space % for threshold 50	Database Space % for threshold 100
Chr10 (Size 137441083 characters)	Chr10q (Size 1270 characters)	10%	0,131%	0,78837%	0,02204%	0,00183%	0,00175%
		20%		1,04613%	0,02891%	0,00297%	0,00184%
		30%		1,40171%	0,03439%	0,00315%	0,00193%
		40%		1,72067%	0,04491%	0,00334%	0,00202%
		50%		2,10432%	0,05045%	0,00352%	0,00212%
		60%		2,52762%	0,05512%	0,00473%	0,00221%
		70%		2,93305%	0,06390%	0,00501%	0,00230%
		80%		3,40080%	0,07020%	0,00529%	0,00239%
		90%		3,93509%	0,08089%	0,00558%	0,00249%
		100%		4,44654%	0,09343%	0,00679%	0,00258%

Table 26 Chimpanzee VS Human Chromosome 10, thresholds 10,25,50,100

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2	Database Space % for threshold 3	Database Space % for threshold 4	Database Space % for threshold 5
Chr11 (Size 135429951 characters)	Chr11q (Size 4772 characters)	10%	2,000%	99,89969%	99,72064%	99,27667%	98,32503%
		20%		99,94135%	99,87177%	99,72273%	99,46184%
		30%		99,95473%	99,90879%	99,84229%	99,71368%
		40%		99,96212%	99,93078%	99,88834%	99,81526%
		50%		99,96688%	99,94362%	99,91043%	99,86711%
		60%		99,97065%	99,95679%	99,92417%	99,89568%
		70%		99,97417%	99,96288%	99,93577%	99,91358%
		80%		99,97706%	99,96917%	99,94787%	99,93091%
		90%		99,97955%	99,97332%	99,95748%	99,93940%
		100%		99,98167%	99,97675%	99,96620%	99,94928%

Table 27 Chimpanzee VS Human Chromosome 11, thresholds 2,3,4,5

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 10	Database Space % for threshold 25	Database Space % for threshold 50	Database Space % for threshold 100
Chr11 (Size 135429951 characters)	Chr11q (Size 4772 characters)	10%	2,000%	88,40167%	64,37156%	64,37156%	2,69741%
		20%		94,59926%	73,33299%	73,33299%	10,48407%
		30%		97,31555%	79,56146%	79,56146%	18,74556%
		40%		98,54749%	84,21386%	84,21386%	26,06771%
		50%		99,20707%	88,03289%	88,03289%	32,59917%
		60%		99,51169%	91,08030%	91,08030%	38,36511%
		70%		99,65589%	93,40593%	93,40593%	43,61202%
		80%		99,75756%	95,14963%	95,14963%	48,33756%
		90%		99,80734%	96,49118%	96,49118%	52,58424%
		100%		99,84457%	97,47937%	97,47937%	56,68336%

Table 28 Chimpanzee VS Human Chromosome 11, thresholds 10,25,50,100

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2	Database Space % for threshold 3	Database Space % for threshold 4	Database Space % for threshold 5
Chr12 (Size 135675203 characters)	Chr12q (Size 3162 characters)	10%	0,322%	92,25198%	76,45637%	58,81778%	43,40459%
		20%		95,69126%	85,98683%	72,39902%	58,37106%
		30%		97,36197%	90,93100%	80,77802%	68,90745%
		40%		98,31786%	93,94807%	86,53004%	76,73967%
		50%		98,87507%	95,90379%	90,35958%	82,63476%
		60%		99,23449%	97,12689%	93,07238%	86,99107%
		70%		99,46598%	97,96996%	94,93220%	90,22009%
		80%		99,61403%	98,54809%	96,28588%	92,61384%
		90%		99,72749%	98,94715%	97,26367%	94,38800%
		100%		99,80232%	99,21605%	97,95864%	95,73729%

Table 29 Chimpanzee VS Human Chromosome 12, thresholds 2,3,4,5

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 10	Database Space % for threshold 25	Database Space % for threshold 50	Database Space % for threshold 100
Chr12 (Size 135675203 characters)	Chr12q (Size 3162 characters)	10%	0,322%	10,63824%	1,76147%	0,06211%	0,00297%
		20%		17,69201%	2,35818%	0,11761%	0,00563%
		30%		25,10284%	3,04437%	0,20006%	0,00610%
		40%		32,62599%	3,91268%	0,26852%	0,00660%
		50%		39,74949%	5,02782%	0,37591%	0,00981%
		60%		46,86688%	6,31075%	0,47629%	0,01061%
		70%		53,50624%	7,86151%	0,60614%	0,01141%
		80%		59,64344%	9,62828%	0,76867%	0,01284%
		90%		65,04520%	11,60009%	0,95564%	0,01833%
		100%		70,14120%	13,67762%	1,19092%	0,02667%

Table 30 Chimpanzee VS Human Chromosome 12, thresholds 10,25,50,100

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2	Database Space % for threshold 3	Database Space % for threshold 4	Database Space % for threshold 5
Chr13 (Size 98704794 characters)	Chr13q (Size 12775 characters)	10%	11,190 %	100,01275%	100,01272%	100,01272%	100,01271%
		20%		100,01275%	100,01272%	100,01272%	100,01271%
		30%		100,01275%	100,01272%	100,01272%	100,01271%
		40%		100,01275%	100,01272%	100,01272%	100,01271%
		50%		100,01275%	100,01272%	100,01272%	100,01271%
		60%		100,01275%	100,01272%	100,01272%	100,01271%
		70%		100,01275%	100,01272%	100,01272%	100,01271%
		80%		100,01275%	100,01272%	100,01272%	100,01271%
		90%		100,01275%	100,01272%	100,01272%	100,01271%
		100%		100,01275%	100,01272%	100,01272%	100,01271%

Table 31 Chimpanzee VS Human Chromosome 13, thresholds 2,3,4,5

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 10	Database Space % for threshold 25	Database Space % for threshold 50	Database Space % for threshold 100
Chr13 (Size 98704794 characters)	Chr13q (Size 12775 characters)	10%	11,190%	100,01264%	100,08978%	100,02399%	100,03153%
		20%		100,01264%	100,01245%	100,01225%	100,04959%
		30%		100,01264%	100,01246%	100,01226%	100,01170%
		40%		100,01264%	100,01246%	100,01226%	100,01170%
		50%		100,01264%	100,01246%	100,01226%	100,01170%
		60%		100,01264%	100,01246%	100,01226%	100,01170%
		70%		100,01264%	100,01245%	100,01225%	100,01170%
		80%		100,01264%	100,01246%	100,01226%	100,01170%
		90%		100,01264%	100,01246%	100,01226%	100,01170%
		100%		100,01264%	100,01246%	100,01226%	100,01170%

Table 32 Chimpanzee VS Human Chromosome 13, thresholds 10,25,50,100

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2	Database Space % for threshold 3	Database Space % for threshold 4	Database Space % for threshold 5
Chr14 (Size 90582208 characters)	Chr14q (Size 5511 characters)	10%	3,134%	100,01674%	100,02127%	100,02037%	100,01067%
		20%		100,00556%	100,00489%	100,00471%	100,00438%
		30%		100,00587%	100,00550%	100,00532%	100,00512%
		40%		100,00587%	100,00587%	100,00584%	100,00573%
		50%		100,00587%	100,00587%	100,00584%	100,00584%
		60%		100,00587%	100,00587%	100,00584%	100,00584%
		70%		100,00587%	100,00587%	100,00584%	100,00584%
		80%		100,00587%	100,00587%	100,00584%	100,00584%
		90%		100,00587%	100,00587%	100,00584%	100,00584%
		100%		100,00587%	100,00587%	100,00584%	100,00584%

Table 33 Chimpanzee VS Human Chromosome 14, thresholds 2,3,4,5

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 10	Database Space % for threshold 25	Database Space % for threshold 50	Database Space % for threshold 100
Chr14 (Size 90582208 characters)	Chr14q (Size 5511 characters)	10%	3,134%	99,20939%	76,58866%	60,98448%	29,58968%
		20%		99,97177%	88,35075%	68,76847%	42,22873%
		30%		99,99255%	96,07625%	75,18778%	50,70975%
		40%		99,99646%	99,04029%	81,20936%	57,33208%
		50%		100,00429%	99,75066%	87,10653%	62,87852%
		60%		100,00535%	99,94408%	92,16482%	68,13112%
		70%		100,00570%	99,98484%	95,86815%	72,85467%
		80%		100,00570%	99,99701%	98,03053%	77,27981%
		90%		100,00570%	100,00103%	99,10465%	81,21912%
		100%		100,00570%	100,00276%	99,62128%	84,66596%

Table 34 Chimpanzee VS Human Chromosome 14, thresholds 10,25,50,100

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2	Database Space % for threshold 3	Database Space % for threshold 4	Database Space % for threshold 5
Chr15 (Size 82071288 characters)	Chr15q (Size 3179 characters)	10%	1,603%	99,60550%	98,07147%	94,81573%	90,27161%
		20%		99,82635%	99,29924%	97,90313%	95,56102%
		30%		99,90867%	99,68677%	99,03781%	97,83201%
		40%		99,94220%	99,83286%	99,50142%	98,86066%
		50%		99,96375%	99,90058%	99,73948%	99,40170%
		60%		99,97551%	99,93968%	99,84955%	99,65491%
		70%		99,98396%	99,96825%	99,91067%	99,78738%
		80%		99,98862%	99,97983%	99,94553%	99,87662%
		90%		99,99252%	99,98628%	99,96650%	99,92498%
		100%		99,99516%	99,99279%	99,98307%	99,95425%

Table 35 Chimpanzee VS Human Chromosome 15, thresholds 2,3,4,5

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 10	Database Space % for threshold 25	Database Space % for threshold 50	Database Space % for threshold 100
Chr15 (Size 82071288 characters)	Chr15q (Size 3179 characters)	10%	1,603%	69,32392%	42,68612%	19,09732%	2,28634%
		20%		78,35360%	49,68117%	25,64255%	4,95528%
		30%		85,00927%	55,50405%	31,43767%	7,73283%
		40%		89,79801%	60,63714%	36,39553%	10,55928%
		50%		93,10135%	65,32694%	40,76020%	13,64020%
		60%		95,41285%	69,78300%	44,88812%	16,50829%
		70%		96,92890%	73,71839%	48,79154%	19,54672%
		80%		97,97292%	77,45162%	52,37787%	22,50427%
		90%		98,64872%	80,89259%	55,71112%	25,41964%
		100%		99,13848%	83,93436%	58,95344%	28,33352%

Table 36 Chimpanzee VS Human Chromosome 15, thresholds 10,25,50,100

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2	Database Space % for threshold 3	Database Space % for threshold 4	Database Space % for threshold 5
Chr16 (Size 83696349 characters)	Chr16q (Size 3113 characters)	10%	0,238%	75,05567%	55,94701%	41,23058%	30,66650%
		20%		80,22641%	64,03604%	49,88208%	39,22708%
		30%		83,81862%	69,89466%	56,63570%	45,91827%
		40%		86,55481%	74,41572%	62,15374%	51,69872%
		50%		88,68524%	78,14949%	66,82448%	56,72068%
		60%		90,45492%	81,19580%	70,98990%	61,06850%
		70%		91,92241%	83,77107%	74,48263%	64,95820%
		80%		93,07519%	85,89730%	77,57457%	68,54085%
		90%		94,07531%	87,64048%	80,07964%	71,81475%
		100%		94,92660%	89,18615%	82,30692%	74,64930%

Table 37 Chimpanzee VS Human Chromosome 16, thresholds 2,3,4,5

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 10	Database Space % for threshold 25	Database Space % for threshold 50	Database Space % for threshold 100
Chr16 (Size 83696349 characters)	Chr16q (Size 3113 characters)	10%	0,238%	8,01946%	0,94156%	0,08521%	0,01205%
		20%		12,88258%	1,83610%	0,26765%	0,02046%
		30%		16,97894%	2,63759%	0,41019%	0,03371%
		40%		20,88884%	3,51784%	0,58724%	0,05723%
		50%		24,68702%	4,52520%	0,77003%	0,06677%
		60%		28,23994%	5,48885%	0,97164%	0,09755%
		70%		31,77278%	6,56632%	1,11827%	0,12737%
		80%		35,11558%	7,62094%	1,34923%	0,15318%
		90%		38,42928%	8,71572%	1,61541%	0,17519%
		100%		41,62968%	9,84672%	1,92804%	0,20582%

Table 38 Chimpanzee VS Human Chromosome 16, thresholds 10,25,50,100

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2	Database Space % for threshold 3	Database Space % for threshold 4	Database Space % for threshold 5
Chr17 (Size 81665014 characters)	Chr17q (Size 3959 characters)	10%	3,013%	99,90502%	99,61152%	98,89835%	97,55295%
		20%		99,94462%	99,86193%	99,58611%	99,15419%
		30%		99,95720%	99,91966%	99,82722%	99,58853%
		40%		99,96291%	99,93874%	99,90653%	99,80478%
		50%		99,96897%	99,95590%	99,93400%	99,87901%
		60%		99,97427%	99,96670%	99,94506%	99,91702%
		70%		99,97815%	99,97140%	99,95462%	99,93737%
		80%		99,98641%	99,97601%	99,96246%	99,94982%
		90%		99,99041%	99,97854%	99,96771%	99,95781%
		100%		99,99238%	99,98097%	99,97451%	99,96638%

Table 39 Chimpanzee VS Human Chromosome 17, thresholds 2,3,4,5

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 10	Database Space % for threshold 25	Database Space % for threshold 50	Database Space % for threshold 100
Chr17 (Size 81665014 characters)	Chr17q (Size 3959 characters)	10%	3,013%	87,76555%	77,51344%	54,60578%	5,37139%
		20%		93,08693%	81,21808%	64,81239%	22,27246%
		30%		96,25514%	84,19845%	70,73300%	34,92096%
		40%		97,93112%	86,73487%	74,95942%	43,85599%
		50%		98,80345%	89,07984%	78,22613%	50,75199%
		60%		99,24372%	91,15241%	80,93637%	56,19707%
		70%		99,52838%	92,92491%	83,24138%	60,81683%
		80%		99,68628%	94,47012%	85,27887%	64,64345%
		90%		99,76978%	95,77254%	87,10630%	68,08451%
		100%		99,81332%	96,79230%	88,63662%	71,19724%

Table 40 Chimpanzee VS Human Chromosome 17, thresholds 10,25,50,100

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2	Database Space % for threshold 3	Database Space % for threshold 4	Database Space % for threshold 5
Chr18 (Size 77548041 characters)	Chr18q (Size 2167 characters)	10%	0,021%	8,94617%	3,17624%	1,23872%	0,51319%
		20%		10,18067%	3,76013%	1,51615%	0,67149%
		30%		11,36236%	4,35377%	1,76132%	0,77564%
		40%		12,58287%	4,99119%	2,03282%	0,88787%
		50%		13,77734%	5,60016%	2,30521%	1,03323%
		60%		14,96043%	6,21027%	2,65582%	1,21655%
		70%		16,14681%	6,79381%	2,99134%	1,38169%
		80%		17,32706%	7,42132%	3,33926%	1,57651%
		90%		18,52157%	8,07781%	3,67813%	1,76158%
		100%		19,65884%	8,72865%	4,05268%	1,95660%

Table 41 Chimpanzee VS Human Chromosome 18, thresholds 2,3,4,5

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 10	Database Space % for threshold 25	Database Space % for threshold 50	Database Space % for threshold 100
Chr18 (Size 77548041 characters)	Chr18q (Size 2167 characters)	10%	0,021%	0,05912%	0,02248%	0,00704%	0,00392%
		20%		0,06448%	0,02444%	0,00760%	0,00420%
		30%		0,06980%	0,02640%	0,00816%	0,00448%
		40%		0,07510%	0,02835%	0,00872%	0,00475%
		50%		0,08345%	0,03031%	0,00928%	0,00503%
		60%		0,08904%	0,03227%	0,00984%	0,00531%
		70%		0,09462%	0,03422%	0,01039%	0,00559%
		80%		0,10026%	0,03618%	0,01099%	0,00587%
		90%		0,10590%	0,03814%	0,01155%	0,00615%
		100%		0,11149%	0,04010%	0,01210%	0,00643%

Table 42 Chimpanzee VS Human Chromosome 18, thresholds 10,25,50,100

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2	Database Space % for threshold 3	Database Space % for threshold 4	Database Space % for threshold 5
Chr19 (Size 58176543 characters)	Chr19q (Size 1438 characters)	10%	0,898%	50,52817%	45,79913%	43,25422%	41,51494%
		20%		53,39265%	48,57161%	45,97346%	44,34091%
		30%		55,99948%	51,02554%	48,36606%	46,63198%
		40%		58,41500%	53,34010%	50,57408%	48,79465%
		50%		60,69059%	55,53151%	52,66745%	50,79630%
		60%		62,81737%	57,52321%	54,61675%	52,71256%
		70%		64,82570%	59,44365%	56,49107%	54,51286%
		80%		66,70372%	61,25022%	58,29792%	56,21834%
		90%		68,45551%	62,98366%	59,96321%	57,86764%
		100%		70,09377%	64,60676%	61,56533%	59,43275%

Table 43 Chimpanzee VS Human Chromosome 19, thresholds 2,3,4,5

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 10	Database Space % for threshold 25	Database Space % for threshold 50	Database Space % for threshold 100
Chr19 (Size 58176543 characters)	Chr19q (Size 1438 characters)	10%	0,898%	35,09451%	13,58127%	0,19739%	0,00000%
		20%		38,58750%	21,87424%	1,74784%	0,00000%
		30%		40,89679%	24,81156%	4,02454%	0,00517%
		40%		43,01896%	27,17355%	6,27972%	0,05067%
		50%		45,02905%	29,22324%	8,16142%	0,12024%
		60%		46,88660%	31,20970%	9,84788%	0,27577%
		70%		48,69856%	33,06492%	11,41306%	0,49319%
		80%		50,42867%	34,83238%	12,86679%	0,75716%
		90%		52,05698%	36,56544%	14,41330%	1,16781%
		100%		53,61335%	38,26990%	15,82005%	1,57814%

Table 44 Chimpanzee VS Human Chromosome 19, thresholds 10,25,50,100

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2	Database Space % for threshold 3	Database Space % for threshold 4	Database Space % for threshold 5
Chr20 (Size 61944263 characters)	Chr20q (Size 3482 characters)	10%	1,664%	99,24431%	96,35832%	90,72847%	83,48308%
		20%		99,65191%	98,52804%	95,88725%	91,82204%
		30%		99,78678%	99,32063%	97,96266%	95,47757%
		40%		99,86097%	99,57534%	98,84724%	97,31797%
		50%		99,91186%	99,72042%	99,31605%	98,34419%
		60%		99,94150%	99,80368%	99,55511%	98,96623%
		70%		99,96114%	99,86395%	99,69174%	99,32875%
		80%		99,97176%	99,90457%	99,76992%	99,52756%
		90%		99,98036%	99,94052%	99,83352%	99,66186%
		100%		99,98523%	99,95411%	99,87926%	99,74535%

Table 45 Chimpanzee VS Human Chromosome 20, thresholds 2,3,4,5

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 10	Database Space % for threshold 25	Database Space % for threshold 50	Database Space % for threshold 100
Chr20 (Size 61944263 characters)	Chr20q (Size 3482 characters)	10%	1,664%	56,75542%	43,68808%	32,75301%	7,67493%
		20%		66,46641%	47,19827%	36,47983%	13,03435%
		30%		75,31526%	50,67713%	39,76887%	16,65025%
		40%		82,43932%	53,95031%	42,78993%	20,11140%
		50%		87,51412%	57,39245%	45,55401%	23,12341%
		60%		91,09386%	60,87687%	48,20399%	25,96649%
		70%		93,59518%	64,64938%	50,69280%	28,86166%
		80%		95,30162%	68,13549%	53,11246%	31,46540%
		90%		96,55112%	71,69106%	55,30480%	33,93814%
		100%		97,44507%	75,04563%	57,47056%	36,43924%

Table 46 Chimpanzee VS Human Chromosome 20, thresholds 10,25,50,100

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2	Database Space % for threshold 3	Database Space % for threshold 4	Database Space % for threshold 5
Chr21 (Size 32724799 characters)	Chr21q (Size 1227 characters)	10%	0,251%	40,08949%	29,38879%	23,53711%	19,54954%
		20%		43,88507%	32,51746%	26,21607%	21,96319%
		30%		47,30834%	35,62359%	28,81889%	24,19390%
		40%		50,48933%	38,58809%	31,25734%	26,35913%
		50%		53,48847%	41,46022%	33,72057%	28,51041%
		60%		56,44865%	44,22841%	36,14676%	30,63168%
		70%		59,11959%	46,82447%	38,47617%	32,70594%
		80%		61,70248%	49,38473%	40,77647%	34,81510%
		90%		64,11385%	51,80925%	42,93177%	36,82214%
		100%		66,30757%	54,04895%	45,10337%	38,75791%

Table 47 Chimpanzee VS Human Chromosome 21, thresholds 2,3,4,5

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 10	Database Space % for threshold 25	Database Space % for threshold 50	Database Space % for threshold 100
Chr21 (Size 32724799 characters)	Chr21q (Size 1227 characters)	10%	0,251%	8,57729%	0,34779%	0,00000%	0,00000%
		20%		9,98194%	0,53831%	0,00000%	0,00000%
		30%		11,27377%	0,70079%	0,00000%	0,00000%
		40%		12,61903%	0,87822%	0,00394%	0,00000%
		50%		13,90003%	1,07943%	0,00431%	0,00000%
		60%		15,20248%	1,29479%	0,00873%	0,00000%
		70%		16,52645%	1,55128%	0,00948%	0,00000%
		80%		17,88774%	1,78550%	0,01023%	0,00000%
		90%		19,24865%	2,09038%	0,01944%	0,00000%
		100%		20,62824%	2,37891%	0,02545%	0,00000%

Table 48 Chimpanzee VS Human Chromosome 21, thresholds 10,25,50,100

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2	Database Space % for threshold 3	Database Space % for threshold 4	Database Space % for threshold 5
Chr22 (Size 35163897 characters)	Chr22q (Size 4583 characters)	10%	2,285%	99,97316%	99,95051%	99,91323%	99,83514%
		20%		99,97072%	99,96033%	99,94066%	99,92615%
		30%		99,96252%	99,96657%	99,95596%	99,95028%
		40%		99,96552%	99,95760%	99,94999%	99,94590%
		50%		99,96682%	99,96151%	99,95445%	99,95272%
		60%		99,96812%	99,96542%	99,95836%	99,95667%
		70%		99,96943%	99,96921%	99,96227%	99,96059%
		80%		99,97073%	99,97071%	99,96593%	99,96426%
		90%		99,97203%	99,97202%	99,96854%	99,96687%
		100%		99,97303%	99,97302%	99,97085%	99,96917%

Table 49 Chimpanzee VS Human Chromosome 22, thresholds 2,3,4,5

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 10	Database Space % for threshold 25	Database Space % for threshold 50	Database Space % for threshold 100
Chr22 (Size 35163897 characters)	Chr22q (Size 4583 characters)	10%	2,285%	97,08669%	71,29801%	20,44742%	0,08011%
		20%		99,19989%	86,12109%	44,65673%	3,29838%
		30%		99,70497%	92,25090%	63,27983%	9,70366%
		40%		99,86124%	95,45992%	74,61622%	19,14591%
		50%		99,90898%	97,36417%	82,16152%	30,67986%
		60%		99,94390%	98,46414%	87,04508%	42,53968%
		70%		99,95336%	99,17791%	90,49428%	52,41529%
		80%		99,95767%	99,53908%	92,93012%	60,73340%
		90%		99,96158%	99,68307%	94,69129%	67,67643%
		100%		99,96508%	99,80814%	96,14135%	73,09763%

Table 50 Chimpanzee VS Human Chromosome 22, thresholds 10,25,50,100

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2	Database Space % for threshold 3	Database Space % for threshold 4	Database Space % for threshold 5
ChrX (Size 150212081 characters)	ChrXq (Size 1658 characters)	10%	0,108%	33,55978%	19,46973%	12,34355%	7,65527%
		20%		37,15220%	23,18893%	15,21837%	9,91249%
		30%		40,52605%	26,24985%	17,78410%	11,84692%
		40%		43,51536%	29,06583%	20,07127%	13,73210%
		50%		46,32315%	31,73492%	22,30471%	15,51644%
		60%		48,96896%	34,24429%	24,45723%	17,35475%
		70%		51,45258%	36,68559%	26,57641%	19,16258%
		80%		53,83319%	39,04943%	28,67284%	20,99111%
		90%		56,06682%	41,27280%	30,72401%	22,75623%
		100%		58,23453%	43,49595%	32,76069%	24,50828%

Table 51 Chimpanzee VS Human Chromosome X, thresholds 2,3,4,5

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 10	Database Space % for threshold 25	Database Space % for threshold 50	Database Space % for threshold 100
ChrX (Size 150212081 characters)	ChrXq (Size 1658 characters)	10%	0,108%	1,40004%	0,01660%	0,00000%	0,00000%
		20%		1,86112%	0,03767%	0,00120%	0,00000%
		30%		2,34148%	0,05840%	0,00393%	0,00000%
		40%		2,83536%	0,08088%	0,00553%	0,00000%
		50%		3,36201%	0,10657%	0,00610%	0,00000%
		60%		3,89377%	0,12605%	0,00656%	0,00000%
		70%		4,45566%	0,15150%	0,00811%	0,00000%
		80%		5,07416%	0,17907%	0,00985%	0,00121%
		90%		5,69801%	0,20935%	0,01295%	0,00132%
		100%		6,32393%	0,24690%	0,01639%	0,00146%

Table 52 Chimpanzee VS Human Chromosome X, thresholds 10,25,50,100

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2	Database Space % for threshold 3	Database Space % for threshold 4	Database Space % for threshold 5
ChrY (Size 11163273 characters)	ChrYq (Size 3175 characters)	10%	0,208%	85,37016%	65,01173%	46,25236%	30,27861%
		20%		89,50116%	74,44551%	58,11092%	42,86122%
		30%		92,10471%	80,56263%	67,29021%	53,41478%
		40%		94,03492%	85,04324%	73,73703%	60,76076%
		50%		95,22364%	88,01914%	78,64068%	66,94981%
		60%		96,20835%	90,46402%	82,55172%	72,36895%
		70%		96,92775%	92,40725%	85,64545%	76,82966%
		80%		97,47011%	93,88837%	88,21238%	80,39675%
		90%		97,89473%	95,00236%	90,08088%	83,37480%
		100%		98,32645%	95,99099%	91,93404%	86,00310%

Table 53 Chimpanzee VS Human Chromosome Y, thresholds 2,3,4,5

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 10	Database Space % for threshold 25	Database Space % for threshold 50	Database Space % for threshold 100
ChrY (Size 11163273 characters)	ChrYq (Size 3175 characters)	10%	0,208%	2,46926%	0,11004%	0,10616%	0,04358%
		20%		5,99838%	0,14909%	0,11508%	0,04644%
		30%		10,24330%	0,19179%	0,12361%	0,04948%
		40%		15,01277%	0,28892%	0,13273%	0,05234%
		50%		20,39106%	0,34182%	0,14130%	0,05518%
		60%		26,33332%	0,40148%	0,15005%	0,05805%
		70%		31,40135%	0,52472%	0,15865%	0,06089%
		80%		36,78767%	0,73833%	0,16750%	0,06376%
		90%		41,71927%	0,93576%	0,17608%	0,06660%
		100%		46,93111%	1,59710%	0,21828%	0,06946%

Table 54 Chimpanzee VS Human Chromosome Y, thresholds 10,25,50,100

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2
Chr3 (Size 202464459 characters)	Chr3q(Size 100 characters)	10%	0,001%	0,01729%
		20%		0,01887%
		30%		0,02045%
		40%		0,02202%
		50%		0,02360%
		60%		0,02517%
		70%		0,02675%
		80%		0,02832%
		90%		0,02990%
		100%		0,03147%
	Chr3q(Size 500 characters)	10%	0,008%	0,79670%
		20%		0,87351%
		30%		0,95016%
		40%		1,02851%
		50%		1,10437%
		60%		1,18071%
		70%		1,25862%
		80%		1,33731%
		90%		1,41301%
		100%		1,48997%
	Chr3q(Size 1000 characters)	10%	0,017%	3,11632%
		20%		3,47686%
		30%		3,83110%
		40%		4,19408%
		50%		4,55829%
		60%		4,92814%
		70%		5,30221%
		80%		5,67721%
		90%		6,06688%
		100%		6,44482%
	Chr3q(Size 5000 characters)	10%	4,622%	100,00274%
		20%		100,00185%
		30%		100,00138%
		40%		100,00197%
		50%		100,00222%
		60%		100,00230%
		70%		100,00230%
		80%		100,00230%
		90%		100,00230%
		100%		100,00230%
	Chr3q(Size 10000 characters)	10%	10,083%	100,00489%
		20%		100,00489%
		30%		100,00489%
		40%		100,00489%
		50%		100,00489%
		60%		100,00489%
		70%		100,00489%
		80%		100,00489%
		90%		100,00489%
		100%		100,00489%

Table 55 Chimpanzee VS Human Chromosome 3, sample of query sizes

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2
Chr6 (Size 177555873 characters)	Chr6q(Size 100 characters)	10%	0,002%	0,04073%
		20%		0,04451%
		30%		0,04835%
		40%		0,05215%
		50%		0,05589%
		60%		0,05963%
		70%		0,06337%
		80%		0,06711%
		90%		0,07085%
		100%		0,07459%
	Chr6q(Size 500 characters)	10%	0,099%	9,64847%
		20%		10,68120%
		30%		11,67053%
		40%		12,66332%
		50%		13,63866%
		60%		14,59264%
		70%		15,52534%
		80%		16,45459%
		90%		17,38603%
		100%		18,31909%
	Chr6q(Size 1000 characters)	10%	0,236%	40,46808%
		20%		44,64828%
		30%		48,31180%
		40%		51,61407%
		50%		54,63318%
		60%		57,42039%
		70%		60,00257%
		80%		62,38816%
		90%		64,61751%
		100%		66,69790%
	Chr6q(Size 5000 characters)	10%	2,810%	100,00389%
		20%		100,00183%
		30%		100,00227%
		40%		100,00256%
		50%		100,00260%
		60%		100,00260%
		70%		100,00260%
		80%		100,00260%
		90%		100,00260%
		100%		100,00260%
	Chr6q(Size 10000 characters)	10%	7,246%	100,00553%
		20%		100,00553%
		30%		100,00553%
		40%		100,00553%
		50%		100,00553%
		60%		100,00553%
		70%		100,00553%
		80%		100,00553%
		90%		100,00553%
		100%		100,00553%

Table 56 Chimpanzee VS Human Chromosome 6, sample of query sizes

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2
Chr7 (Size 162359053 characters)	Chr7q(Size 100 characters)	10%	0,003%	0,05655%
		20%		0,06164%
		30%		0,06699%
		40%		0,07245%
		50%		0,07759%
		60%		0,08266%
		70%		0,08803%
		80%		0,09335%
		90%		0,09842%
		100%		0,10357%
	Chr7q(Size 500 characters)	10%	0,007%	0,73294%
		20%		0,80748%
		30%		0,88017%
		40%		0,95209%
		50%		1,02418%
		60%		1,09549%
		70%		1,16561%
		80%		1,23774%
		90%		1,31259%
		100%		1,38700%
	Chr7q(Size 1000 characters)	10%	0,915%	40,22183%
		20%		43,13810%
		30%		45,81004%
		40%		48,39850%
		50%		50,82363%
		60%		53,10113%
		70%		55,27278%
		80%		57,36327%
		90%		59,34553%
		100%		61,23563%
	Chr7q(Size 5000 characters)	10%	3,803%	99,98067%
		20%		99,98305%
		30%		99,98625%
		40%		99,98844%
		50%		99,99041%
		60%		99,99225%
		70%		99,99380%
		80%		99,99534%
		90%		99,99720%
		100%		99,99873%
	Chr7q(Size 10000 characters)	10%	7,320%	100,00611%
		20%		100,00611%
		30%		100,00611%
		40%		100,00611%
		50%		100,00611%
		60%		100,00611%
		70%		100,00611%
		80%		100,00611%
		90%		100,00611%
		100%		100,00611%

Table 57 Chimpanzee VS Human Chromosome 7, sample of query sizes

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2
Chr8 (Size 148638763 characters)	Chr8q(Size 100 characters)	10%	0,001%	0,01185%
		20%		0,01292%
		30%		0,01398%
		40%		0,01504%
		50%		0,01611%
		60%		0,01717%
		70%		0,01823%
		80%		0,01930%
		90%		0,02036%
		100%		0,02142%
	Chr8q(Size 500 characters)	10%	0,005%	0,49938%
		20%		0,54744%
		30%		0,59497%
		40%		0,64194%
		50%		0,69111%
		60%		0,73795%
		70%		0,78682%
		80%		0,83513%
		90%		0,88369%
		100%		0,93326%
	Chr8q(Size 1000 characters)	10%	0,016%	3,17670%
		20%		3,55469%
		30%		3,92693%
		40%		4,31028%
		50%		4,69081%
		60%		5,07828%
		70%		5,45789%
		80%		5,83749%
		90%		6,22314%
		100%		6,60664%
	Chr8q(Size 5000 characters)	10%	1,715%	99,99139%
		20%		99,99715%
		30%		99,99967%
		40%		100,00161%
		50%		100,00270%
		60%		100,00303%
		70%		100,00303%
		80%		100,00303%
		90%		100,00303%
		100%		100,00303%
	Chr8q(Size 10000 characters)	10%	6,395%	100,00661%
		20%		100,00661%
		30%		100,00661%
		40%		100,00661%
		50%		100,00661%
		60%		100,00661%
		70%		100,00661%
		80%		100,00661%
		90%		100,00661%
		100%		100,00661%

Table 58 Chimpanzee VS Human Chromosome 8, sample of query sizes

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2
Chr9 (Size 120061799 characters)	Chr9q(Size 100 characters)	10%	0,001%	0,01484%
		20%		0,01620%
		30%		0,01765%
		40%		0,01921%
		50%		0,02059%
		60%		0,02196%
		70%		0,02343%
		80%		0,02484%
		90%		0,02622%
		100%		0,02760%
	Chr9q(Size 500 characters)	10%	0,008%	0,80243%
		20%		0,88056%
		30%		0,95851%
		40%		1,03707%
		50%		1,11236%
		60%		1,18926%
		70%		1,26874%
		80%		1,34567%
		90%		1,42409%
		100%		1,50226%
	Chr9q(Size 1000 characters)	10%	0,026%	5,35506%
		20%		6,01849%
		30%		6,63870%
		40%		7,26293%
		50%		7,89173%
		60%		8,52433%
		70%		9,14423%
		80%		9,74554%
		90%		10,35877%
		100%		10,97574%
	Chr9q(Size 5000 characters)	10%	1,718%	99,98122%
		20%		100,00082%
		30%		99,99261%
		40%		99,99472%
		50%		99,99797%
		60%		99,99935%
		70%		100,00018%
		80%		100,00101%
		90%		100,00185%
		100%		100,00287%
	Chr9q(Size 10000 characters)	10%	2,823%	100,00818%
		20%		100,00818%
		30%		100,00818%
		40%		100,00818%
		50%		100,00818%
		60%		100,00818%
		70%		100,00818%
		80%		100,00818%
		90%		100,00818%
		100%		100,00818%

Table 59 Chimpanzee VS Human Chromosome 9, sample of query sizes

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2
Chr13 (Size 98704794 characters)	Chr13q(Size 100 characters)	10%	0,003%	0,06280%
		20%		0,06863%
		30%		0,07437%
		40%		0,08021%
		50%		0,08595%
		60%		0,09170%
		70%		0,09744%
		80%		0,10319%
		90%		0,10893%
		100%		0,11467%
	Chr13q(Size 500 characters)	10%	0,070%	4,45073%
		20%		4,87248%
		30%		5,28352%
		40%		5,68569%
		50%		6,08693%
		60%		6,48881%
		70%		6,89201%
		80%		7,29818%
		90%		7,69215%
		100%		8,09314%
	Chr13q(Size 1000 characters)	10%	0,167%	26,08391%
		20%		29,16503%
		30%		31,96758%
		40%		34,65496%
		50%		37,20974%
		60%		39,61983%
		70%		41,96174%
		80%		44,20348%
		90%		46,39564%
		100%		48,44972%
	Chr13q(Size 5000 characters)	10%	2,076%	99,99819%
		20%		99,99937%
		30%		100,00039%
		40%		100,00161%
		50%		100,00212%
		60%		100,00263%
		70%		100,00313%
		80%		100,00373%
		90%		100,00424%
		100%		100,00461%
	Chr13q(Size 10000 characters)	10%	7,273%	100,00994%
		20%		100,00994%
		30%		100,00994%
		40%		100,00994%
		50%		100,00994%
		60%		100,00994%
		70%		100,00994%
		80%		100,00994%
		90%		100,00994%
		100%		100,00994%

Table 60 Chimpanzee VS Human Chromosome 13, sample of query sizes

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2
Chr17 (Size 81665014 characters)	Chr17q(Size 100 characters)	10%	0,181%	3,11416%
		20%		3,37625%
		30%		3,63447%
		40%		3,88956%
		50%		4,14086%
		60%		4,39101%
		70%		4,63784%
		80%		4,88117%
		90%		5,12172%
		100%		5,35860%
	Chr17q(Size 500 characters)	10%	0,263%	18,75500%
		20%		20,05536%
		30%		21,32886%
		40%		22,55922%
		50%		23,77716%
		60%		24,95837%
		70%		26,11051%
		80%		27,24080%
		90%		28,35097%
		100%		29,43217%
	Chr17q(Size 1000 characters)	10%	0,292%	34,39168%
		20%		36,88812%
		30%		39,21623%
		40%		41,42825%
		50%		43,52947%
		60%		45,54940%
		70%		47,48160%
		80%		49,34243%
		90%		51,12970%
		100%		52,84733%
	Chr17q(Size 5000 characters)	10%	2,447%	100,01353%
		20%		100,00947%
		30%		100,01129%
		40%		100,00588%
		50%		100,00588%
		60%		100,00588%
		70%		100,00588%
		80%		100,00588%
		90%		100,00588%
		100%		100,00588%
	Chr17q(Size 10000 characters)	10%	7,634%	100,02408%
		20%		100,01215%
		30%		100,01215%
		40%		100,01215%
		50%		100,01215%
		60%		100,01215%
		70%		100,01215%
		80%		100,01215%
		90%		100,01215%
		100%		100,01215%

Table 61 Chimpanzee VS Human Chromosome 17, sample of query sizes

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2
Chr20 (Size 61944263 characters)	Chr20q(Size 100 characters)	10%	0,002%	0,02642%
		20%		0,02882%
		30%		0,03123%
		40%		0,03363%
		50%		0,03604%
		60%		0,03844%
		70%		0,04085%
		80%		0,04326%
		90%		0,04566%
		100%		0,04807%
	Chr20q(Size 500 characters)	10%	0,284%	1,36073%
		20%		1,49779%
		30%		1,63864%
		40%		1,78397%
		50%		1,92537%
		60%		2,06960%
		70%		2,23883%
		80%		2,38248%
		90%		2,53070%
		100%		2,67858%
	Chr20q(Size 1000 characters)	10%	0,677%	7,74490%
		20%		8,76071%
		30%		9,77452%
		40%		10,83406%
		50%		11,87529%
		60%		12,88276%
		70%		13,96062%
		80%		15,02230%
		90%		16,07882%
		100%		17,13547%
	Chr20q(Size 5000 characters)	10%	8,054%	100,00203%
		20%		99,99682%
		30%		99,99983%
		40%		100,00172%
		50%		100,00253%
		60%		100,00333%
		70%		100,00414%
		80%		100,00495%
		90%		100,00576%
		100%		100,00656%
	Chr20q(Size 10000 characters)	10%	20,769%	100,01601%
		20%		100,01601%
		30%		100,01601%
		40%		100,01601%
		50%		100,01601%
		60%		100,01601%
		70%		100,01601%
		80%		100,01601%
		90%		100,01601%
		100%		100,01601%

Table 62 Chimpanzee VS Human Chromosome 20, sample of query sizes

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2
Chr21 (Size 32724799 characters)	Chr21q(Size 100 characters)	10%	0,0003%	0,00568%
		20%		0,00620%
		30%		0,00672%
		40%		0,00724%
		50%		0,00776%
		60%		0,00828%
		70%		0,00879%
		80%		0,00931%
		90%		0,00983%
		100%		0,01035%
	Chr21q(Size 500 characters)	10%	0,104%	7,92979%
		20%		8,61542%
		30%		9,30939%
		40%		9,99223%
		50%		10,66799%
		60%		11,34378%
		70%		12,01907%
		80%		12,68003%
		90%		13,33473%
		100%		13,98347%
	Chr21q(Size 1000 characters)	10%	0,135%	20,47533%
		20%		22,59326%
		30%		24,66217%
		40%		26,60677%
		50%		28,55396%
		60%		30,47313%
		70%		32,36602%
		80%		34,20779%
		90%		36,05433%
		100%		37,82167%
	Chr21q(Size 5000 characters)	10%	3,813%	100,00949%
		20%		100,01370%
		30%		100,01462%
		40%		100,01462%
		50%		100,01462%
		60%		100,01462%
		70%		100,01462%
		80%		100,01462%
		90%		100,01462%
		100%		100,01462%
	Chr21q(Size 10000 characters)	10%	9,228%	100,03037%
		20%		100,03037%
		30%		100,03037%
		40%		100,03037%
		50%		100,03037%
		60%		100,03037%
		70%		100,03037%
		80%		100,03037%
		90%		100,03037%
		100%		100,03037%

Table 63 Chimpanzee VS Human Chromosome 21, sample of query sizes

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Database Space % for total query	Database Space % for separated queries
				Query, parts of 250 characters
chr3	(total query) chr3q (Size 965 characters)	10%	3,11632%	0,52070%
		20%	3,47686%	0,56867%
		30%	3,83110%	0,61614%
		40%	4,19408%	0,66427%
		50%	4,55829%	0,71237%
		60%	4,92814%	0,76059%
		70%	5,30221%	0,80866%
		80%	5,67721%	0,85704%
		90%	6,06688%	0,90536%
		100%	6,44482%	0,95454%

Table 64 Chimpanzee VS Human Chromosome 3, partitioned queries

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Database Space % for total query	Database Space % for separated queries				
				Query, parts of 1000 characters	Query, parts of 500 characters	Query, parts of 250 characters	Query, parts of 150 characters	Query, parts of 100 characters
chr13	(total query) chr13q (Size 12775 characters)	10%	100,01275%	88,01238%	56,79287%	31,86639%	19,89180%	13,94842%
		20%	100,01275%	90,67320%	60,12930%	34,10059%	21,40844%	15,08292%
		30%	100,01275%	92,60044%	63,19930%	36,23447%	22,86692%	16,09720%
		40%	100,01275%	94,09599%	66,01189%	38,29229%	24,26311%	17,18371%
		50%	100,01275%	95,25021%	68,58937%	40,28316%	25,62632%	18,24584%
		60%	100,01275%	96,13737%	70,97263%	42,20140%	26,95467%	19,20843%
		70%	100,01275%	96,84698%	73,17878%	44,06180%	28,25127%	20,23413%
		80%	100,01275%	97,40666%	75,21913%	45,85386%	29,52300%	21,15410%
		90%	100,01275%	97,85075%	77,08886%	47,59061%	30,77262%	22,12870%
		100%	100,01275%	98,21509%	78,83651%	49,32940%	32,06973%	23,08460%

Table 65 Chimpanzee VS Human Chromosome 13, partitioned queries

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Database Space % for total query	Database Space % for separated queries				
				Query, parts of 1000 characters	Query, parts of 500 characters	Query, parts of 250 characters	Query, parts of 150 characters	Query, parts of 100 characters
chrY	(total query) chrYq (Size 3175 characters)	10%	85,37016%	28,71856%	13,37890%	5,64921%	2,31991%	1,61528%
		20%	89,50116%	31,22707%	14,55194%	6,16694%	2,53105%	1,76568%
		30%	92,10471%	33,60840%	15,69361%	6,66244%	2,73901%	1,90569%
		40%	94,03492%	36,03200%	16,83530%	7,16010%	2,94596%	2,05462%
		50%	95,22364%	38,34534%	17,96173%	7,65594%	3,15315%	2,20667%
		60%	96,20835%	40,53956%	19,06005%	8,14972%	3,35590%	2,34350%
		70%	96,92775%	42,69016%	20,15520%	8,64082%	3,56221%	2,49282%
		80%	97,47011%	44,79639%	21,23398%	9,12619%	3,76416%	2,62907%
		90%	97,89473%	46,85125%	22,29112%	9,60871%	3,96835%	2,77846%
		100%	98,32645%	48,77194%	23,35790%	10,10898%	4,18313%	2,92668%

Table 66 Chimpanzee VS Human Chromosome Y, partitioned queries

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Database Space % for total query	Database Space % for separated queries				
				Query, parts of 1000 characters	Query, parts of 500 characters	Query, parts of 250 characters	Query, parts of 150 characters	Query, parts of 100 characters
chr14	(total query) chr14q (Size 5511 characters)	10%	100,01674%	69,15938%	43,12856%	26,19166%	17,92562%	15,64721%
		20%	100,00556%	72,65680%	45,61789%	27,68458%	18,80748%	16,44722%
		30%	100,00587%	75,72566%	47,97386%	29,12305%	19,66644%	17,14927%
		40%	100,00587%	78,50986%	50,21297%	30,52113%	20,50034%	17,90154%
		50%	100,00587%	80,87882%	52,35994%	31,88578%	21,31537%	18,63114%
		60%	100,00587%	82,96427%	54,39879%	33,21766%	22,11451%	19,28584%
		70%	100,00587%	84,81686%	56,34392%	34,51588%	22,90178%	19,98893%
		80%	100,00587%	86,49343%	58,21141%	35,78447%	23,67827%	20,62464%
		90%	100,00587%	87,98409%	59,97977%	37,02412%	24,44489%	21,30725%
		100%	100,00587%	89,31199%	61,71904%	38,28187%	25,24824%	21,98113%

Table 67 Chimpanzee VS Human Chromosome 14, partitioned queries

Database chimpanzee chromosome	Query part of human chromosome	Window % of query	Database Space % for total query	Database Space % for separated queries				
				Query, parts of 1000 characters	Query, parts of 500 characters	Query, parts of 250 characters	Query, parts of 150 characters	Query, parts of 100 characters
chr17	(total query) chr17q (Size 3959 characters)	10%	99,90502%	69,89792%	48,57718%	32,92334%	26,06596%	21,07837%
		20%	99,94462%	72,76146%	50,88465%	34,52019%	27,35643%	22,13933%
		30%	99,95720%	75,26236%	53,03681%	36,01898%	28,56185%	23,09437%
		40%	99,96291%	77,55061%	55,06353%	37,45038%	29,65408%	24,09981%
		50%	99,96897%	79,60203%	56,96315%	38,83275%	30,70280%	25,03218%
		60%	99,97427%	81,45742%	58,76281%	40,17371%	31,70883%	25,87087%
		70%	99,97815%	83,12965%	60,47796%	41,48152%	32,68530%	26,75642%
		80%	99,98641%	84,64677%	62,08856%	42,74030%	33,64453%	27,53505%
		90%	99,99041%	85,99897%	63,62694%	43,96280%	34,57922%	28,33422%
		100%	99,99238%	87,25219%	65,10413%	45,19452%	35,54353%	29,11285%

Table 68 Chimpanzee VS Human Chromosome 17, partitioned queries

Database mouse chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2
Chr1 (Size 192572129 characters)	Chr1q (Size 7154 characters)	10%	2,488%	99,98306%
		20%		99,98018%
		30%		99,98093%
		40%		99,98167%
		50%		99,98241%
		60%		99,98316%
		70%		99,98379%
		80%		99,98416%
		90%		99,98453%
		100%		99,98490%

Table 69 Mouse VS Human Chromosome 1

Database mouse chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2
Chr2 (Size 178498087 characters)	Chr2q (Size 8079 characters)	10%	4,980%	99,95824%
		20%		99,95915%
		30%		99,96006%
		40%		99,96096%
		50%		99,96187%
		60%		99,96277%
		70%		99,96368%
		80%		99,96458%
		90%		99,96549%
		100%		99,96639%

Table 70 Mouse VS Human Chromosome 2

Database mouse chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2
Chr3 (Size 156486682 characters)	Chr3q (Size 965 characters)	10%	0,019%	3,82329%
		20%		4,26198%
		30%		4,69041%
		40%		5,13222%
		50%		5,56361%
		60%		6,02517%
		70%		6,47945%
		80%		6,93592%
		90%		7,41301%
		100%		7,87954%

Table 71 Mouse VS Human Chromosome 3

Database mouse chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2
Chr4 (Size 152250714 characters)	Chr4q (Size 5170 characters)	10%	1,036%	99,85915%
		20%		99,85152%
		30%		99,85355%
		40%		99,85559%
		50%		99,85763%
		60%		99,85967%
		70%		99,86187%
		80%		99,86391%
		90%		99,86595%
		100%		99,86799%

Table 72 Mouse VS Human Chromosome 4

Database mouse chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2
Chr5 (Size 148148609 characters)	Chr5q (Size 1776 characters)	10%	0,100%	29,61796%
		20%		33,41637%
		30%		36,88392%
		40%		40,36044%
		50%		43,44496%
		60%		46,44846%
		70%		49,28706%
		80%		52,03128%
		90%		54,64723%
		100%		57,13288%

Table 73 Mouse VS Human Chromosome 5

Database mouse chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2
Chr6 (Size 146317037 characters)	Chr6q (Size 2434 characters)	10%	0,508%	92,06880%
		20%		94,67190%
		30%		96,19884%
		40%		97,24030%
		50%		97,95163%
		60%		98,44507%
		70%		98,81499%
		80%		99,07253%
		90%		99,26276%
		100%		99,40975%

Table 74 Mouse VS Human Chromosome 6

Database mouse chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2
Chr7 (Size 142713043 characters)	Chr7q (Size 700 characters)	10%	0,020%	2,91641%
		20%		3,21043%
		30%		3,51286%
		40%		3,81125%
		50%		4,11005%
		60%		4,41583%
		70%		4,71612%
		80%		5,02466%
		90%		5,33138%
		100%		5,63053%

Table 75 Mouse VS Human Chromosome 7

Database mouse chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2
Chr8 (Size 125596464 characters)	Chr8q (Size 1780 characters)	10%	0,078%	26,20453%
		20%		29,99362%
		30%		33,44835%
		40%		36,71574%
		50%		39,69132%
		60%		42,37054%
		70%		44,93152%
		80%		47,40462%
		90%		49,80466%
		100%		51,99646%

Table 76 Mouse VS Human Chromosome 8

Database mouse chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2
Chr9 (Size 121198575 characters)	Chr9q (Size 1865 characters)	10%	0,333%	54,09982%
		20%		59,69588%
		30%		64,56008%
		40%		68,52633%
		50%		72,00284%
		60%		75,01927%
		70%		77,65243%
		80%		80,00777%
		90%		82,10859%
		100%		83,94394%

Table 77 Mouse VS Human Chromosome 9

Database mouse chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2
Chr10 (Size 126857255 characters)	Chr10q (Size 1270 characters)	10%	0,145%	32,89505%
		20%		37,08353%
		30%		40,94344%
		40%		44,54076%
		50%		47,93626%
		60%		51,02908%
		70%		53,91866%
		80%		56,60324%
		90%		59,15723%
		100%		61,57005%

Table 78 Mouse VS Human Chromosome 10

Database mouse chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2
Chr11 (Size 118743856 characters)	Chr11q (Size 4772 characters)	10%	1,211%	100,00562%
		20%		100,00121%
		30%		100,00287%
		40%		100,00338%
		50%		100,00338%
		60%		100,00338%
		70%		100,00338%
		80%		100,00338%
		90%		100,00338%
		100%		100,00338%

Table 79 Mouse VS Human Chromosome 11

Database mouse chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2
Chr12 (Size 117826530 characters)	Chr12q (Size 3162 characters)	10%	0,511%	94,75894%
		20%		97,19886%
		30%		98,30664%
		40%		98,87153%
		50%		99,19999%
		60%		99,40289%
		70%		99,52222%
		80%		99,60005%
		90%		99,64316%
		100%		99,67021%

Table 80 Mouse VS Human Chromosome 12

Database mouse chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2
Chr13 (Size 116800623 characters)	Chr13q (Size 12775 characters)	10%	9,566%	99,97485%
		20%		99,97704%
		30%		99,97922%
		40%		99,98141%
		50%		99,98360%
		60%		99,98556%
		70%		99,98665%
		80%		99,98775%
		90%		99,98884%
		100%		99,98993%

Table 81 Mouse VS Human Chromosome 13

Database mouse chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2
Chr14 (Size 122093164 characters)	Chr14q (Size 5511 characters)	10%	1,246%	99,67909%
		20%		99,67930%
		30%		99,68365%
		40%		99,68771%
		50%		99,69177%
		60%		99,69583%
		70%		99,69990%
		80%		99,70396%
		90%		99,70802%
		100%		99,71209%

Table 82 Mouse VS Human Chromosome 14

Database mouse chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2
Chr15 (Size 100439974 characters)	Chr15q (Size 3179 characters)	10%	0,645%	99,35845%
		20%		99,79576%
		30%		99,91337%
		40%		99,97319%
		50%		99,99412%
		60%		99,99443%
		70%		99,99748%
		80%		99,99878%
		90%		99,99973%
		100%		100,00068%

Table 83 Mouse VS Human Chromosome 15

Database mouse chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2
Chr16 (Size 95283144 characters)	Chr16q (Size 3113 characters)	10%	0,149%	61,70553%
		20%		68,76250%
		30%		73,73586%
		40%		77,69442%
		50%		80,93794%
		60%		83,61692%
		70%		85,89748%
		80%		87,80272%
		90%		89,46215%
		100%		90,81614%

Table 84 Mouse VS Human Chromosome 16

Database mouse chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2
Chr17 (Size 92696390 characters)	Chr17q (Size 3959 characters)	10%	0,676%	99,63854%
		20%		99,71898%
		30%		99,73629%
		40%		99,74597%
		50%		99,75068%
		60%		99,75409%
		70%		99,75708%
		80%		99,76007%
		90%		99,76294%
		100%		99,76550%

Table 85 Mouse VS Human Chromosome 17

Database mouse chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2
Chr18 (Size 87601031 characters)	Chr18q (Size 2167 characters)	10%	0,033%	9,34599%
		20%		10,54817%
		30%		11,78732%
		40%		13,02966%
		50%		14,18873%
		60%		15,38415%
		70%		16,61677%
		80%		17,80799%
		90%		19,01038%
		100%		20,21270%

Table 86 Mouse VS Human Chromosome 18

Database mouse chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2
Chr19 (Size 58142430 characters)	Chr19q (Size 1438 characters)	10%	0,068%	15,36548%
		20%		17,18873%
		30%		18,99475%
		40%		20,76738%
		50%		22,53907%
		60%		24,30557%
		70%		26,01272%
		80%		27,73095%
		90%		29,35698%
		100%		31,00693%

Table 87 Mouse VS Human Chromosome 19

Database mouse chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2
ChrX (Size 163935371 characters)	ChrXq (Size 1658 characters)	10%	0,145%	38,24272%
		20%		42,17699%
		30%		45,96645%
		40%		49,27440%
		50%		52,35424%
		60%		55,22538%
		70%		57,92598%
		80%		60,78914%
		90%		63,18074%
		100%		65,82946%

Table 88 Mouse VS Human Chromosome X

Database mouse chromosome	Query part of human chromosome	Window % of query	Hit Rate %	Database Space % for threshold 2
ChrY (Size 56385016 characters)	ChrYq (Size 3175 characters)	10%	0,200%	83,59658%
		20%		88,64514%
		30%		91,52841%
		40%		94,07515%
		50%		95,31453%
		60%		96,37510%
		70%		97,21260%
		80%		97,88715%
		90%		98,33452%
		100%		98,67234%

Table 89 Mouse VS Human Chromosome Y

Appendix B – TUC PreBLAST Software Tools

preblast.c : produces the probable hits for the PreBLAST filter simulating the memories and the spliting of the w-mer

window.c : produces the windowed results of the PreBLAST filter

threshold.c : produces the space results of the PreBLAST filter

charcount.c : counts characters for the query and the database

hitcount.c : counts actual and probable hits

splitquery.c : creates the partitioned parts of the query

ncbispace.c : produces the spaces of the NCBI results

checkspaces.c : checks the spaces of PreBLAST in compare with NCBI

reverse.c : transforms the FASTA type datasets in simple format

hitsdistance.c : counts the distance between the hits and their distribution

extensions.c : counts the extension, their width and distribution

totalspaces.m : produces the total space, concatenating the spaces of the partitioned datasets and removing the overlapping spaces (matlab)

testbenchcreator1.c : creates the vhdl testbench for the query

testbenchcreator2.c : creates the vhdl testbench for the database