Post–Genome Applications Based on Multi–Parallel Computing over High Performance Network

Fumio Aoki†, Hirofumi Akashi†, Masamichi Goudge†, Minoru Toyota††, Yasushi Sasaki††,
Xin Guo† † †, Shengjun Li † † †† Takashi Tokino††, Haruyuki Tatsumi†
Sapporo Medical University †Information Center of Computer Communication, ††Cancer Research Institute
† † †Shenyang Institute of Administration, China
† † ††China Medical University, China
Email: kaku, hakashi, mguuji, tokino, tatsumi@sapmed.ac.jp

Abstract

To retrieve a particular sequence and to deduce a biological meaning from the monotonous arrangement of nucleotides, we need a powerful computer and a huge storage device[1]. One of the solutions is a multi–parallel network computing[2, 3, 4], which we devised for getting an arbitrary section of the human body using the Visible Human Project (VHP: National Library of Medicine in USA[5]) data set. We improved the system for the analysis of the function of a tumor suppressor gene, $p53$[6, 7], providing a Web–based system. The background engine is powered by multi-parallel computing clusters interconnected with TCP/IP network. The engine has a head computer, which issues control commands to data processing nodes for various kinds of jobs such as retrieving particular patterns, generating DNA mapping images, loading data sections from genome database, and so on. By using this flexible client/server structure connected over high performance network, we can efficiently modify our system corresponding to increasing data and new algorithms for investigation by slightly changing the control procedures and increasing the number of the processor node.

Keywords

Network Computing, Genome, $p53$ Prediction, Distributed System, $cDNA$ mapping

1 Introduction

The Human Genome Project (genome sequencing) has been almost completed in 2000, that was announced in Washington D.C. before the new century. This means a start of new age of a Post-Genome Project for biomedical–researchers to investigate the mysteries of life behind the simple arrangement of nucleic acids, applying the fruits to medical treatments corresponding to personal DNA, especially finding a way to overcome cancers. Since the human genome contains a great amount of biological information and indeterminate number of meanings, the investigation forces time-consuming tasks for us to retrieve and search a special portion through the whole sequence data using a new algorithm. This paper proposes a multi–parallel computing solution for the purpose of dealing with the genome sequence data. This system is based on the parallel processing architecture for VHP image viewer [8, 9], which had been used successfully in a G7 Information Society Project, GIBN (Global Interoperability for Broadband Network), in 2000[10].

Mutations of $p53$ gene[6, 7] are the most common genetic lesion in human cancer, present in more than 50% of all cases of the disease. At the first time, the $p53$ protein was considered to function as an oncogene, which causes the transformation of normal cells into cancerous tumor cell, however, several critical discoveries defined the normal function of $p53$ to be anti-oncogenic. The $p53$ gene product is a tetrameric, sequence-specific DNA-binding protein with a defined cognate binding site containing two copies of the 10-mer(5'-RRRCA/TT/AGYYY-3'). Using the binding characteristics of the protein, we have developed a post genome application for $p53$ analysis taking advantage of the multi-parallel network computing system.

2 The System

A Web–based interface of the system enables the accessibility through the Internet, that is a GUI interface of the DNA investigation tools. The data processing unit is a background engine powered by our parallel computer cluster and has a head computer, which issues control commands to data processing nodes for various kinds of jobs, such as retrieving particular patterns, generating DNA mapping images, loading data sections from the human

genome sequence, and so on. The Web–based interface communicates with the head computer(task controller) using a set of CGI(Common Gateway Interface) programs. The task controller exchanges data with data processing nodes via TCP socket connection which is widely used as the basic protocol over the Internet. Therefore our system is the Internet compliant and a very promising application for the Next Generation Internet, a high performance network.

The Web–based DNA investigation tools and the parallel processing engine are a client/server system, and the task controller and the data processing nodes are connected in the same way. By using this flexible client and server structure connected with high performance network, we can efficiently modify our system corresponding to the increasing database or new investigation algorithms, by slightly changing the control procedures of the task controller or increasing the number of the processing node.

In dealing with the human genome sequence, there are a lot of tasks such as loading data sections from the sequence data set, generating images to visualize the nucleotide portions and its associated biological information, retrieving the particular portions from the nucleic acid sequence data, dividing $cDNA$ into $EXON$s and mapping them onto the genome sequence, translating nucleotide portions into protein sequence, and etc. The task controller receives queries from users through Web browser and generates control procedures for each data processing node. The different tasks can be assigned to different processing nodes, furthermore, the same tasks can also be divided into sub–tasks and assigned to several processing nodes.

3 The Procedures

To prove the performance of our multi–parallel computing model for the DNA investigation tools, a project was initiated (1) to localize $p53$ protein binding site permitting some ambiguities, (2) breaking $cDNA$ into $EXON$s and mapping them to the genetic sequence, and (3) translating the DNA portions into its protein sequence. The $cDNA$ mapping to the genomic sequence is required to determine the affected gene. $p53$ gene was chosen for this experiment because the mutations of the gene disable an emergency brake on cell proliferation and lead to genetic instability, that is very important as a vicious circle for carcinogenesis.

To analyze the functions of $p53$ protein, we need to predict the $p53$ protein binding site to the DNA sequence, finding out the affected genes, from which we will deduce the function of the $p53$ protein and do an experiment to prove it. The binding rule is not so strict that it gives us a lot of possible binding sites and it is impossible to find out with human eye-search. Therefore, this is considered to be a typical interdisciplinary field between the genetics and information technology.

Since $p53$ protein has an indeterminate rule to bind DNA sequences, it is difficult to predict the binding site by general pattern matching as done in usual information retrievals. In our retrieval approach, a multi–step comparison algorithm with partial compatibility of $p53$ protein binding is used, that can search by both perfect matching scheme and partial matching scheme.

On the other hand, there might be the genes regulated by $p53$ beside the predicted binding site. It is very useful to visualize both these genes and the $p53$ binding site for the genetic researchers, but it is difficult to map $cDNA$ genes physically onto the genomic sequence, because their genomic structures are usually unavailable. For the structural gene mapping, we developed a segment portion expanding algorithm for breaking and mapping human $cDNA$ onto the genetic sequence data in $EXON$ level. Usually, $cDNA$ is composed by many $EXON$s without any marks between them. It is difficult to divide $cDNA$ into $EXON$s at correct points, the only available knowledge is that the $EXON$ portions exist on the genetic sequence and they have the contextual order in the sequence. Our algorithm divides the $cDNA$ into small segments with equal length to be mapped onto the sequence database with considering the contextual relation. When a segment is matched somewhere on the sequence, the algorithm starts expanding along both ends until there appears different nucleic acid. Then, the start and end pointers of this $EXON$ are stored as a matching record. After all the small segments are processed, we have a set of these pointer records, but there may be some records with the same $EXON$ pointers, these must be deleted automatically.

For the protein translations of a given DNA sequence, a table looking up algorithm is applied. The program gets query and DNA sequence from Web-based GUI, checks its validity and converts into internal format to be translated according to the protein table. On the Web browser, user can obtain both the protein sequence and the image marked with red lines

as stop sign starting from three different translation points. The correct translation is automatically determined by detecting the maximum length in the final output $JPEG$ image.

4 $p53$ Binding Site Prediction

In the human genetic sequences, there are four kinds of nucleic acid described by A, C, G, T as a flat text in a linear coding format. Occasionally, N is also used when purine base (A or G) or pyrimidine base (T or C) can not be determined. Table 1 shows the nucleic acid, coding and their descriptions.

Table 1: Nucleic Acid Coding

Code	Nucleic Acid	Description
A	A	Adenine
C	C	Cytosine
G	G	Guanine
$T(U)$	T	Thymine(Uracil)
R	A or G	puRine
Y	C or T	pYrimidine
S	G or C	Strong
W	A or T	Weak
K	G or T	Keto
M	A or C	aMino
B	C, G or T	not A
D	A, G or T	not C
H	A, C or T	not G
V	A, C or G	not $T(U)$
$N(X)$	Any Residue	aNy(uNknown)
.	Loss	

$p53$ protein DNA binding sequence[6] is represented as below using the coding method in Table 1.

$$[RRRCWWGYYY] + (N)_m + [RRRCWWGYYY]$$

The $(N)_m$ in this expression means there must be 0 to 12 any residues in that position. All the codes here are indeterminate elements except C and G, for example, R, W, Y may be two possible nucleic acid respectively. Consequently, this part has upto 256 x 256 = 65,536 possible patterns statistically. When the maximum 12 possible nucleic acid codes exist defined by $(N)_m$ are put together, we have about 65,536 x 13 = 851,968 patterns to match in the prediction process. This is a heavy task of generating comparison patterns, and it needs considerable computer resources and costs much time to complete the general pattern matching.

By considering the portions on both sides of $p53$ protein binding have the same pattern, here we use a multi–step searching algorithm, the primary searching retrieves the pattern from the genetic sequence, and the secondary searching collects the satisfying pairs from the records of primary searching by checking if the interval length is within $(N)_m$. To improve the response speed of the program, the sequence data are loaded into the matching processors in the pre–loading step, and matching rate checking operations for immediate stop with reaching unmatched nucleic acid are done during both the primary and secondary searching as well as the interval length checking.

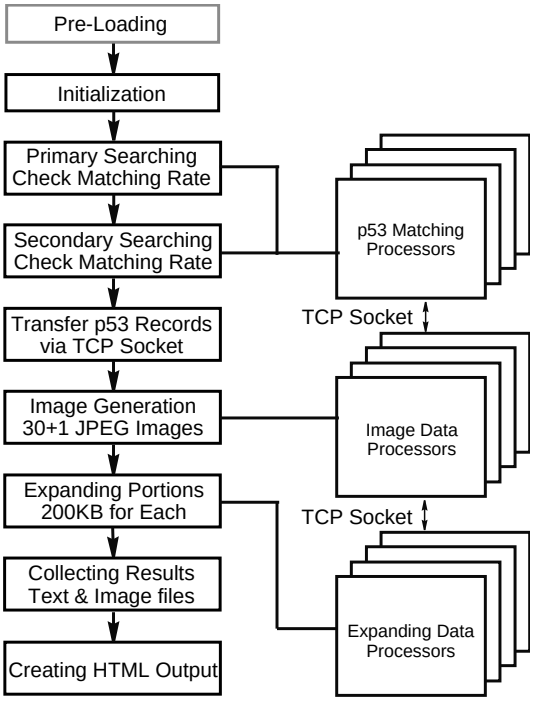

Figure 1: $p53$ Prediction Algorithm

After collecting the satisfying records of the prediction site, they are transferred from $p53$ matching processors to both image data processors and expanding data processors via TCP socket connections, for the generations of DNA mapping images and the expanding output. Here, 31 $JPEG$ images are generated by a group of image processors, one is for the top image indicating the positions of total $p53$ protein binding portions on the sequence data, 30 images are for detail visualizations. Expanding portions nearby each $p53$ protein bind portion can be retrieved and output upto 200KB length according to the user's query. The last task is to write the text data and image files onto the disk of the Web server, and to create a HTML content to the standard output. The algo-

rithm is shown in Figure 1.

5 $cDNA$ Mapping

$cDNA$ is often given only including $EXON$s without breaking points. To separate the portions and to mapping them onto the sequence data takes much computer power and time. We sometimes need to know how the $cDNA$ is located on the sequence, there is no typical algorithm at present. Here, we propose an adaptive expanding algorithm of sectioning the $cDNA$ into pieces and mapping them adaptively onto the sequence data. Furthermore, unreasonable solutions are automatically eliminated according to $EXON$ context, maximum distance between first and last $EXON$, and minimum length of each $EXON$. The algorithm is shown in Figure 2.

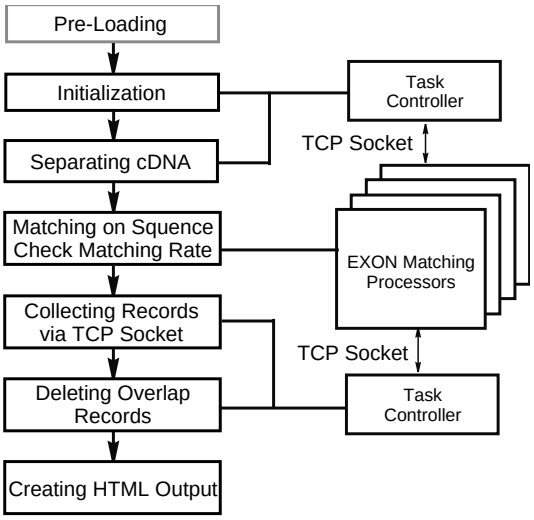

Figure 2: $cDNA$ Mapping Algorithm

By pre–loading the sequence data into $EXON$ matching processors, the overhead for accessing tens megabytes of data can be cut off. $cDNA$ portion is received from Web browser when users summit the query, these are pre–processed for checking the content legality and cutting off the illegal bytes, reading the parameters for minimum number of $EXON$ length and maximum number of unmatched bytes.

The task controller receives the information in the query to divide $cDNA$ into equal length small pieces, and send them to $EXON$ matching processors via TCP socket connections. Matchings on the sequence are performed parallelly with checking matching rate based on maximum unmatched bytes, $EXON$ context and the distance between first and last $EXON$. After collecting the records from matching processors, task controller evaluates the overlapped records with the same start and end pointers and those included within other $EXON$s. Here, only one image is generated to indicate the $cDNA$ location on the genetic sequence data. The last task is to write the text data and image file onto the disk of the Web server, and to send a HTML content to the standard output.

6 Protein Translation

Protein translation process is much simple, it just converts every byte in given DNA portions into protein description by looking up the conversion table installed in the system. The algorithm has a pre–processing step for checking the content legality and cutting off the illegal bytes, getting the text output switch from user's query. We do not use parallel processors for the conversion because this task is very light for present computers. Image data processor is used only for generating protein mapping image of $JPEG$ format for Web browser. At last, all the text data and image file are written on the Web server's disk, and a HTML content is created and send to the standard output. The algorithm is shown in Figure 3.

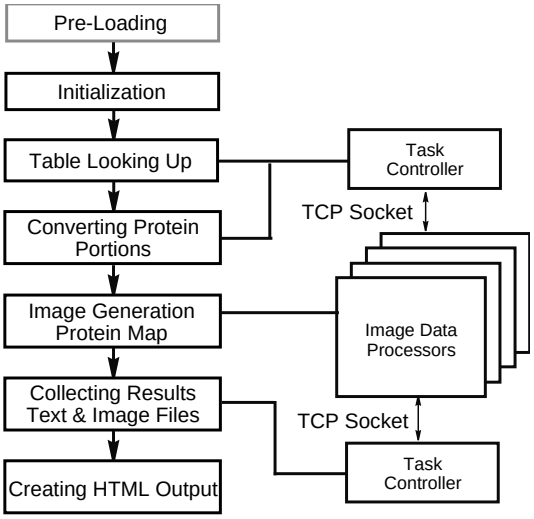

Figure 3: Protein Translation Algorithm

7 Prototype and Results

A prototype system was developed to verify the validity of our proposal. The processor cluster was implemented by five low cost DOS/V computers connected with 100MB fast ether network. These computer has a 533MHz CPU, 128MB memory, 20GB

hard drive, and installed with RedHat6.2J, a Linux system software, because the CPU power in the project is more important than its memory space, unlike our previous approach for VHP image viewer which needs a lot of memory upto 13GB(male) or 35 GB(female). Web server software and task controller program are working on the same computer, the rest computers acted as image processors and matching processors. The architecture of our prototype system is given in Figure 4. The Web server and task controller can be installed on different computers, and also no necessity for the task controller and processor cluster computers are located in the same network. The system can be used through the Internet, if your terminal is installed a widely used Web browser, such as Netscape or Internet Explorer.

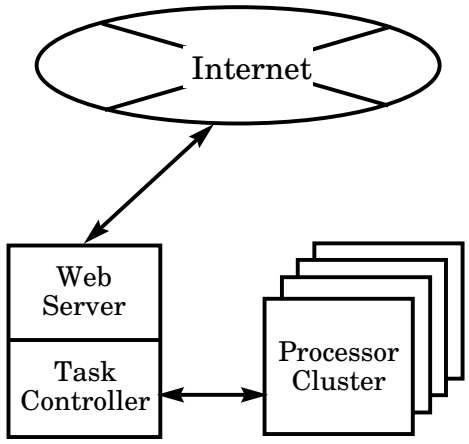

Figure 4: Prototype System Architecture

The chromosome No.21 and No.22 sequence data released from GenBank are used for $p53$ protein binding site prediction and $cDNA$ mapping in this prototype system. The chromosome No.21 sequence has about 28MB nucleic acid code, and chromosome No.22 has 23MB. The programs including processing modules, data communication modules, and CGI modules were developed using general UNIX-C language in Linux OS, by the staff of our information center.

The $p53$ protein binding site prediction processing time and the matching records were measured for both choromosome No.21 and No.22 sequence data, corresponding the primary searching and secondary searching, the perfect matching and error matching. The results are given in Table 2, 3.

The rebuilding of this system for opening to the public service is ongoing. The screen copies of $p53$ protein binding site prediction, $cDNA$ mapping and protein translation on test run of this system are shown in Figure 5. The result pages are in Figure 6, 7, 8, respectively.

Table 2: $p53$ Prediction on No.21 Chromosome

Primary Searching

Matching Rate	100%	90%	80%
Time Costed	9 sec.	16 sec.	23 sec.
p53 Records	7,729	126,709	822,356

Secondary Searching

Matching Rate	100%	95%	90%
Time Costed	0.42 ms	12 ms	120 ms
p53 Records	45	811	9,276

Table 3: $p53$ Prediction on No.22 Choromosome

Primary Searching

Matching Rate	100%	90%	80%
Time Costed	7 sec.	12 sec.	19 sec.
p53 Records	6,390	104,481	630,819

Secondary Searching

Matching Rate	100%	95%	90%
Time Costed	0.36 ms	11 ms	92 ms
p53 Records	42	649	7,167

8 Conclusion

This paper proposed a multi–parallel computing solution over high performance network, for the purpose of retrieving particular DNA sequence and investigating through the human genome sequence data, instead of expensive super computers. The prototype system is implemented to work for (1) $p53$ protein binding site prediction by ambiguous searching, (2) $cDNA$ mapping by separated $EXON$ matching, (3) translating DNA portions into its protein sequence.

A multi–step comparison algorithm with portion compatibility is proposed for $p53$ protein binding site prediction, an adaptive expanding algorithm is proposed for $cDNA$ mapping, both are proved efficient by

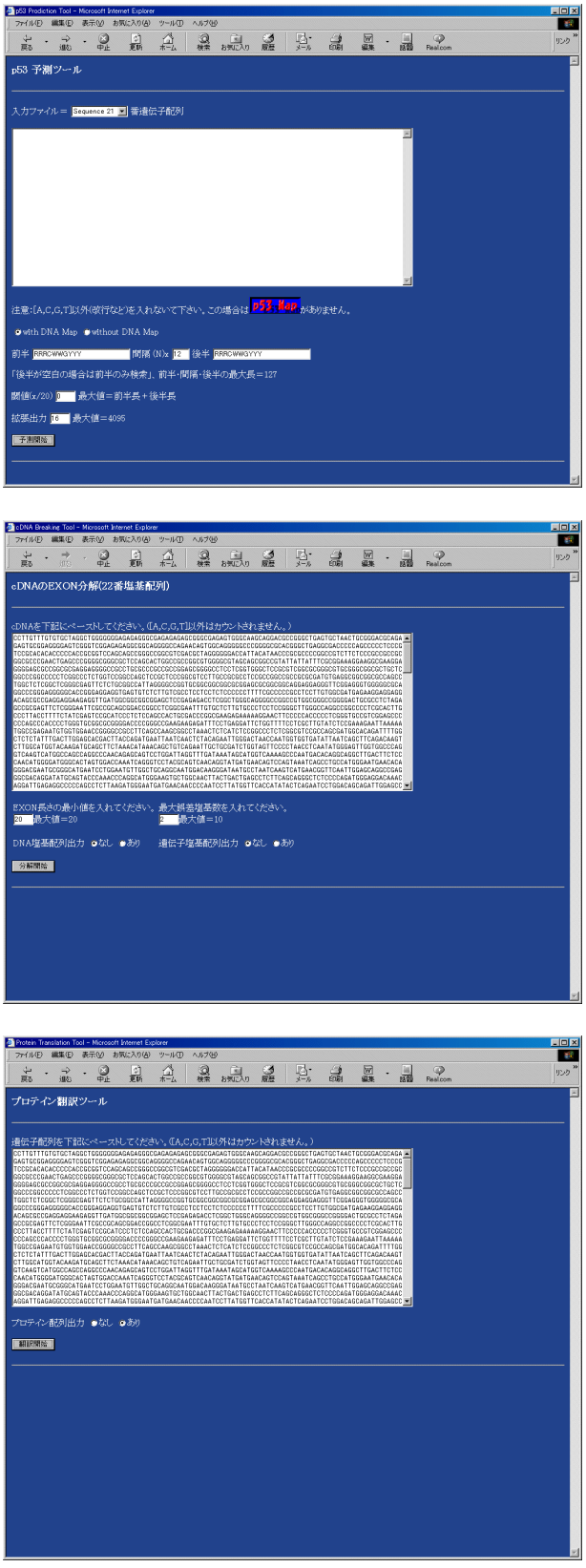

Figure 5: Top Pages of The System

our prototype system. A Web–based interface is used for easy access, user can send query to the system and get results, including text data and mapping images from his Web browser. This system uses the parallel processing cluster as its background engine powered by task controller and data processing nodes. This flexible client/server structure enables the efficiency of modifying system architecture corresponding to the increasing database or new investigation algorithms.

As the future work, the release of a Web site for public service based on our solution, the implementation of more algorithms related to genome research, the improvement of control procedure corresponding to the increasing parallel processing nodes and genome sequence database, and the feature detection method based on artificial intelligence are expected.

Acknowledgement

This study was supported in part by Industrial Technology Research Grant Program in 2000 from New Energy and Industrial Technology Development Organization (NEDO) of Japan, and Research for the Future Program of Japan Society for the Promotion of Science under the Project "Integrated Network Architecture for Advanced Multimedia Application Systems" (BJSPS-RFTF97R16301).

References

[1] Hoshi Masanori, Challenge to Genome Informatics – The Fusion of Computer and Bio, **Kyoritsu Publishing**, 1994. (Japanese)

[2] Aoki Fumio, A Parallel Approach for VHP Image Viewer, **IWS2000**, Proceedingd on Medical Session-I, pp.209-214, Feb.2000.

[3] Aoki Fumio, Distributed Processing for Large Medical Image Database, **JAMIT Annual Meeting 2000**, Proceedings pp.58–59, July 2000.

[4] Aoki Fumio, Distributed Computing Approach for High Resolution Medical Images, **16th World Computer Congress 2000**, Proceedings on Software: Theory and Practice, pp.611–618, Aug.2000.

[5] VHP, Visible Human Project of National Library of Medicine. http://www.nlm.nih.gov/research/visible/

[6] Ko LJ, Prives C, p53: Puzzle and Paradigm, **Genes & Development** 10: pp.1054–1072 (1996)

[7] Taya Yoichi, A Variety of Physiological Function of p53, **Experimental Medicine**, Vol.16, No.15, pp.1876–1879, Yodosya, 1998. (Japanese)

[8] GIBN (Global Interoperability for Broadband Network) Project of G7 Information Society. http://www.sapmed.ac.jp/gibn/

[9] Tatsumi H, Nakamura M, Aoki F, Nakahashi N, Akashi H, Guuji M, Medical Use of the Internet and Development of the Infrastructure, **J.ISCIE** 44: pp.554–565, 2000. (Japanese)

[10] Tatsumi H, Murakami G, Nogawa H, Aoki F, Nakamura M, Akashi H, Nakahashi N, Anatomical Co–laboratory of G7 Information Society as a US–Japan Collaboratory Experiment, **Proceedings of NORTH Internet Symposium 2000**, pp.54–63 (ISSN 1342–0690). (Japanese)

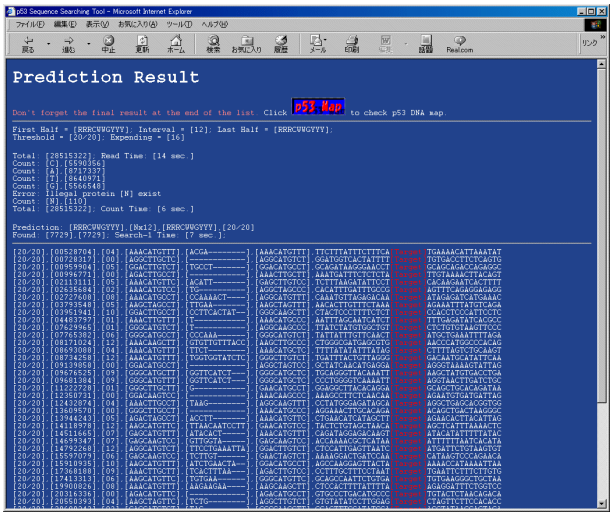

Figure 6: Result Pages of $p53$ Prediction

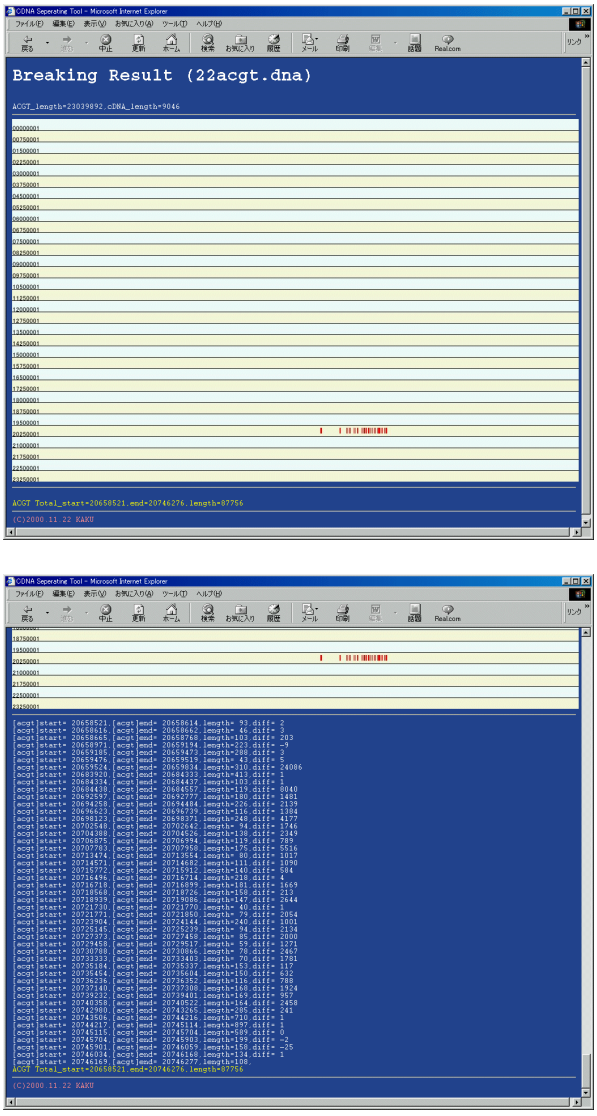

Figure 7: Result Pages of $cDNA$ Mapping

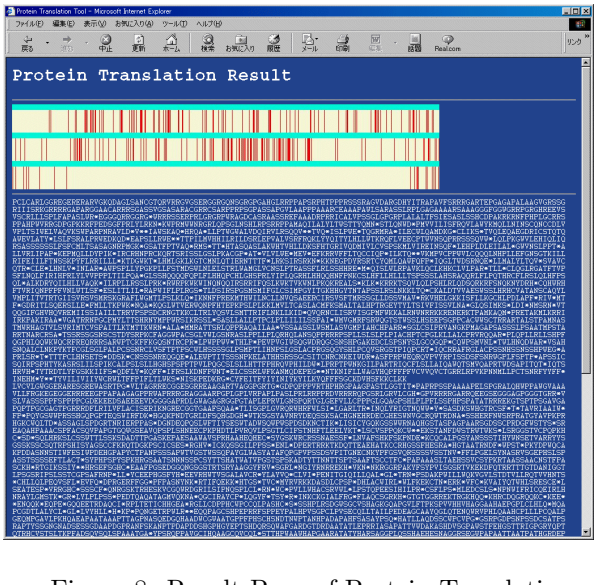

Figure 8: Result Page of Protein Translation