

NAIST-IS-DD0361202

Doctoral Dissertation

Bioinformatics tool for genomic era: A step towards the *in silico* experiments-focused on molecular cloning

Akira Ohyama

August 1, 2006

Department of Bioinformatics and Genomics
Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
Submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of SCIENCE

Akira Ohyama

Thesis Committee:
Professor, Naotake Ogasawara
Professor, Shigehiko Kanaya
Professor, Kotaro Minato
Associate Professor, Ken Kurokawa

Bioinformatics tool for genomic era: A step towards the *in silico* experiments-focused on molecular cloning*

Akira Ohyama

Abstract

The thesis focuses on performing the molecular biological experiments *in silico*, one of the simulation tools of the biological experiments *in vitro*. Among the various fields of experiments, this thesis especially focuses on molecular cloning, because DNA information is more systematically and exhaustively collected than that of other molecules. There are some advantages in *in silico* experiments in consideration of planning molecular biological experiments, usage as lab notebook, educational tool to learn molecular biological experiments. However, performing *in silico* cloning requires some formulations such as recording of the end shapes of digested products by restriction enzymes or amplified products by PCR. For this purpose, a few extensions to Genbank/EMBL database annotation convention are introduced and incorporated into existing convention as new feature keys and qualifiers. In addition, features on a DNA sequence are occasionally truncated in the case of amplified products of PCR or digested fragments by restriction enzymes, therefore the annotations about the truncated features should also be formulated. The ambivalent nature of DNA requires frequent changes of the interested strand to and fro, therefore a reverse complementary operation of large size DNA is necessary to be implemented for such a software system. According to these definitions or data descriptions, a software tool for *in silico* experiments, named *in silico* MolecularCloning has been developed, and performed a few of typical molecular cloning experiments on computer, and verified that this approach might be effective.

In addition, several functions covering a number of requirements in this genomic era, are also implemented in the software. Genome information is rapidly being accumulated and this warrants performance in software which swiftly refers and edits

* Doctoral Dissertation, Department of Bioinformatics and Genomics, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD0361202, August, 2006

such enormous data. The software tools accomplished one of the fastest speed records of reading and editing such large sized genome information among the many software systems which run on ordinary PC. Together with these high performances, also implemented are comparative genomics facilities. As one of the applications to the exhaustive expression analysis, additional development of data management and expression profile viewer and analyzer has been finished for high-density tiling arrays which entirely cover a whole genome with complementary probes as if the building tiles cover walls. Thus, the software is expected to provide further acceleration to the expression analysis researches. On the other hand, software tools, named MetaGenomeGAMBLER, for DNA sequencing are also developed at the same time with facilities of comprehensive and semi-automated functions for such upstream experiments. Its high-quality controlling of sequencing raw data provides high-throughput processing and data accuracy for the projects. This software is also applied to cDNA sequencing and clustering projects for plant genomes which are said to be relatively difficult to be wholly sequenced.

As an application of the just-developed software tools, I investigated the asymmetry of nucleotide compositions in a number of the prokaryotic genomes and found that there is an unreported phenomenon in the local and transcription-coupled asymmetry of GC compositions. This phenomenon, which is currently found only among the prokaryotes with relatively low-GC content, is especially remarkable in species as follows, *Clostridii* including *Clostridium perfringens*, *Fusobacterium nucleatum*, and is also detected in the genomes of *Escherichia coli* and *Bacilli*. On the contrary, none of prokaryotes with high-GC content, shows any local asymmetry of GC composition coupled with transcription. This finding may lead to development of prediction algorithm to detect transcription units on prokaryotic genomes.

Keywords; in silico experiment, molecular cloning, GC Skew, Software, transcription-coupled, sequencing, *Clostridium*, cDNA expression analysis, tiling array, comparative genomics

ゲノム時代のインシリコ実験の実現を目標としたバイオインフォマティクスツール構築*

大山 彰

内容梗概

インビトロ（試験管内）での生物学実験を模倣することを目標としたインシリコ（コンピュータ内）における新規分子生物学実験シミュレーションシステムを構築した。生物学の数多くの分野の中でも分子クローニングに着目し、生物分野でもっとも系統的にまた網羅的に情報が収集されているDNA情報を利用する。インシリコ実験にはいくつかの利点があり、たとえば、分子生物学実験の計画に活用できる、実験ノートの代用となる、分子生物学実験の教育用ツールとして用いることができることなどが挙げられる。しかしながら、インシリコ実験を実装するためには、制限酵素による消化切断生成物の末端形状やPCR生成物などの記録方法などの定式化が必要である。このため、Genbank/EMBLデータベース注釈規約にいくつかの拡張を行い、新規FeatureおよびQualifierとして定義した。さらに、DNA配列上に表現されたFeatureはPCRや制限酵素消化切断などの際に部分的に削られることがあり、このためこの部分的に削られたFeatureを記述する規約の定式化を行った。また、DNAの二面的な性質により、通常興味の対象としては片方のDNA鎖のみを表示するが、インシリコ実験を実装するソフトウェアにおいてはこの対象としての長鎖DNA配列をその上にFeatureを保持したままその逆相補鎖と頻繁に切り替える操作が必要であることを示した。これらの定義やデータ記述に従い、「インシリコモレキュラークローニング」と呼ぶインシリコ実験ソフトウェアを開発し、いくつかの典型的な分子クローニング実験をコンピュータ上で実行し、この方法が有効であることを示した。同時に、このソフトウェアにはゲノム時代に対応した様々な機能を搭載した。ゲノム情報については、現在もなお膨大な量のデータが集積されつつあり、これらの大量情報を高速に参照かつ編集できることが望まれている。本ソフトウェアは、これらのニーズに対応し、通常のPC等で動作するソフトウェアとしては最高速のデータ参照・編集速度を実現した。この高速性能を利用して、異種ゲノム間の比較解析機能を充実させた。近年研究が進んでいる網羅的な発現解析への対応機能としては、ゲノム全体をタイルのように多い尽くす高密度アレイであるタイリングアレイ用のデータ操作、及び発現情報プロファイル表示機能を追加開発した。これにより、アレイ解析速度が大幅に向上し、発現解析研究に貢献することが期待される。一方、より上流の実験である塩基配列シーケンシングを一層容易に遂行するためのソフトウェアである「メタゲノムガンブラー」を開発し、配列決定からアノテーションまでの一連の作業の多くを自動化するとともに、配列などの品質評価機能を充実させる

* 奈良先端科学技術大学院大学 情報科学研究科 情報生命科学専攻 博士論文、NAIST-IS-0361202. 2006年8月

ことにより、データの正確性を向上させている。このソフトウェアはゲノムサイズが巨大であるため、全ゲノム塩基配列決定が困難である植物ゲノムのcDNA発現解析にも使用できる。

これらのソフトウェアを使用した応用例として、原核生物における塩基組成の非対称性を調べた結果、ある種の原核生物にはこれまで報告されていない転写に共役するGC組成の非対称性を確認した。原核生物の中でも比較的GC含量の低い種に確認されたこの現象は、特に以下の生物種、*Clostridium perfringens*を含む*Clostridium*属、*Fusobacterium nucleatum*などに著しく、*Escherichia coli*や*Bacillus*属にも、その存在が確認された。一方、高GC含量をもつ原核生物ではこの局所的GC組成の非対称性を確認できていない。この発見は、一部の原核生物における転写単位の予測などに応用できる可能性がある。

キーワード：インシリコ実験、分子クローニング、GC Skew、ソフトウェア、転写共役型、シーケンシング、*Clostridium*、cDNA発現解析、タイリングアレイ、ゲノム比較

Contents

Chapter 1 Overview of the thesis	
1.1 Motivation	2
1.2 Backgrounds	5
1.3 Concepts of software environments for molecular biology	8
1.4 Overviews of the thesis	11
Chapter 2 Assembling and quality controls of genomic DNA and cDNA	
2.1 Introduction	15
2.2 Methods and algorithms	18
2.3 Results: Implementation of the MetaGenomeGAMBLER (MGG)	27
2.4 Discussion	45
Chapter 3 Genomic DNA sequence analysis and its front end environment	
3.1 Introduction	49
3.2 Methods and algorithms	51
3.3 Results: Implementation of in silico MolecularCloning (IMC)	65
3.4 Discussion	84
Chapter 4 <i>in silico</i> experiments of molecular cloning	
4.1 Introduction	89
4.2 Methods and algorithms	92
4.3 Results	99
4.4 Discussion	108
Chapter 5 A viewer for tiling microarray data	
5.1 Introduction	113
5.2 Methods and algorithms	117
5.3 Results: Implementation of a software tool for the tiling microarray	128
5.4 Discussion	143
Chapter 6 Asymmetry found in the local composition of GC and its correlation to the transcript units and directions in the genomes	
6.1 Introduction	149
6.2 Methods and Algorithms	151

6.3 Results	152
6.4 Discussion	163
Conclusions	165
Acknowledgements	171
References	173
Appendix	187
A. List of functions of <i>in silico</i> MolecularCloning	189
B. Molecular Cloning protocols <i>in silico</i>	197
C. List of microbial genomes with the transcription-coupled GC skew	201

List of Figures

Figure 1.1 Conceptual diagram of a front end software tools for molecular biology	9
Figure 1.2 Consecutive operations of programs.	10
Figure 1.3 Diagram of relationship between sections in the thesis	11
Figure 2.1 Flow diagram of genome sequencing to annotation	16
Figure 2.2 Theoretical curves plotted by number of fragments	23
Figure 2.3 Characteristic pattern caused by erroneous assembling	24
Figure 2.4 Contig connectivity using mates information	26
Figure 2.5 Package design diagram of MetaGenomeGAMBLER (MGG)	27
Figure 2.6 Plot of contig generation and coverage of the genome	29
Figure 2.7 Naming rule parser window for extracting attributes of fragments	31
Figure 2.8 Importing window of sequencer raw data	32
Figure 2.9 Quality Viewer, a graphical presentation of quality evaluation	33
Figure 2.10 Vector registration window of MGG	34
Figure 2.11 Masking controls of raw sequencing data	35
Figure 2.12 Identification of misassembling caused by repetitive sequences	36
Figure 2.13 Detection of misassembled regions caused by repetitive sequences	37
Figure 2.15 Automatic annotation by MGG and IMC	38
Figure 2.16 Plate style viewing in quality per fragment	39
Figure 2.17 Viewing of quality per base in the bar style	40
Figure 2.18 Trace Viewer of MGG	41
Figure 2.19 Contig viewer with consensus sequence	41
Figure 2.20 The consensus index and its viewer	42
Figure 2.21 Contig Linkage Viewer	43
Figure 3.1 Conceptual data structure of IMC internal format	51
Figure 3.2 Package design diagram of <i>in silico</i> MolecularCloning	65
Figure 3.3 Feature map of the genome from <i>Bacillus subtilis</i>	69
Figure 3.4 Feature map of the genome from <i>Escherichia coli</i>	70
Figure 3.5 Reference map of <i>Bacilli</i> .	71
Figure 3.6 Plasmid map with an insert ballooned out from the circular plasmid	72
Figure 3.7 A circular genome map of <i>Bacillus subtilis</i>	73
Figure 3.8 Sequence viewer showing a CDS region	74
Figure 3.9 The Genbank/EMBL viewer	75

Figure 3.10 Feature statistics window	77
Figure 3.11 Codon usage table of <i>Bacillus subtilis</i>	79
Figure 3.12 Multiple alignment and phylogenetic tree	81
Figure 3.13 Dot plot between <i>Bacillus halodurans</i> and <i>B.subtilis</i>	83
Figure 4.1 Flow of cloning experiments	89
Figure 4.2 Example of Genbank format file	94
Figure 4.3 Examples of DNA fragments digested by restriction enzymes	96
Figure 4.4 Examples of ligation reactions	97
Figure 4.5 IMC Operation in consecutive routines	107
Figure 5.1 Schematic diagram of a quasi-tiling microarray (part)	114
Figure 5.2 Probes on the <i>Bacillus subtilis</i> whole genome map	115
Figure 5.3 Diagram of relationship between Probe, CDF, CEL and Genome Map	117
Figure 5.4 Tiling microarray data formats	119
Figure 5.5 Parameter setting dialog for importing a probe and a CDF file	128
Figure 5.6 The CEL files importing window	129
Figure 5.7 GenBank format file with tiling microarray annotation	130
Figure 5.8 Microarray data format and profiles for RNA hybridization	131
Figure 5.9 Histograms of means and standard deviations of all the probe intensities	132
Figure 5.10 Cutting off of outliers	133
Figure 5.11 R-I plot between intensities comparing two microarrays	134
Figure 5.12 Perfect match vs. mismatch probe intensity	135
Figure 5.13 Settings of the arithmetic operations between microarrays	136
Figure 5.14 Normalization by expression of genomic fragments hybridization	137
Figure 5.15 Parallel viewing of tiling microarray expression profiles with annotation	138
Figure 5.16 Superimpose of the profiles on the reverse complementary strand	139
Figure 5.17 Gene level expression table	140
Figure 5.18 Clustering of a gene expression matrix	141
Figure 5.19 Manually selected set of genes with similar patterns of expression	142
Figure 6.1 Local GC skew profiles	153
Figure 6.2 GC skews per gene cluster	157
Figure 6.3 Negative correlation is observed between GC content and GC skew	159
Figure 6.4 Various directions of mutation are observed among genomes.	160
Figure 6.5 Amino acid usages of 13 different genomes	162

List of Tables

Table 1.1 Traditional and widely used sequence analysis programs	6
Table 3.1 Importing performance of IMC	84
Table 4.1 Experimental functions provided by IMC (Part)	101

Chapter 1

Overview of the thesis

1.1 Motivations

One decade has passed since the first sequencing of complete genomes of *Mycoplasma genitalium* and *Haemophilus influenzae* (Fleischmann et al. 2005). Currently, more than 400 genomes have been completely sequenced. In addition, some model genomes of further complex organization including that of *Homo sapiens* have also been completely sequenced and these DNA sequences are publicly available via internet for every researchers and students.

Contrary to these achievements in the genome DNA sequencing, there have been delays in development of suitable software tools for the genomic data handling. The basis to the most of sequence analysis software tools had been established during 1980s, although their target was the merely a clone size sequence. For example, GCG (Genetic Computing Group) Wisconsin Package (Gribskov & Devereux 1991) was developed in early 80s, however it is still used by molecular biologists without adaptation to large-size genome data handling. Since the funding to the sequencing of complete genomes has been huge, and these genomic DNA sequences are surely the source of a lot of new knowledge about biology or medical sciences. Utilization of such treasures for human beings must be excavated. Therefore, software tools for the researchers to easily utilize numerous genomic DNA sequences should be developed.

In addition, since the late 1990s, high throughput methods to detect expression of biological molecules have been invented. Among them, the DNA array is the most successful device with very subtle techniques. However, DNA array experiments produce larger amount of data compared with that of DNA sequencing. Therefore, further high-throughput performance of analytical software is also required.

Previously, DNA sequences were usually stored on remote servers and multiple users are accessing the servers from client PCs because large capacities and high performances are required for maintaining and providing sequence data. This style of data server leads to a consensus of data gathering policies or restrictions to use these data. However, some of the researchers were not satisfied with these common and limited data, instead they like to possess their own private database. Thus, private data handling in a local PC is required.

Although many software tools for different purposes have been developed, when using these tools, data conversions usually become most disturbing obstacles because the input and output format of these tools are not standardized. Development of

software tools which enable to use different tools without data conversion, is also necessary.

DDBJ/EMBL/GenBank format file is commonly used data format for nucleotide and amino acid sequence data, and it is written in text format and visible by text editors. Therefore, data portability and visibility are higher than other formats specific to particular program. Thus, DDBJ/EMBL/GenBank format as a basic recording format is required.

Consequently, to handle sequence data together with other data such as mRNA expression data, complicated data linkages is necessary. Thus, simpler format for experiment data is required.

Earlier software tools for sequence analysis usually ran only on Unix machines. After Windows PC and Macintosh became popular among researchers, the developers of biological software tools usually release Windows version at first and Mac version much later. Thus, software compatibility between Windows, Mac and Unix is much requested.

in vitro experiments in the field of molecular biology are often performed consecutively. For example, amplification of DNA by PCR, digestion of a plasmid vector, ligation of the PCR product and the linear vector are performed in series. Thus, consecutive operations *in silico* are also required.

In addition, the results in molecular biology are always invisible because that the target molecules are too small to see directly and only recognized by indirect manners, such as gel electrophoresis. A graphical presentation of the micro phenomena is much helpful.

Experiments *in vitro* are regarded as cycles of designing, verifying and recording of experiments. During the cycle, a lot of works must be done by manually. To reduce the efforts, more comprehensive tools for supporting this cycle are required.

Here in this thesis, I propose newly designed integrated genome research environments with functions which satisfy the above-mentioned requirements: handling genome scale sequence, and high-throughput experimental data, compatibility between Windows, Mac and Unix, private data handling, and *in silico* experimental tools to facilitate the wider comprehensive understanding of biological experiments.

1.2 Backgrounds

(1) History of sequence analysis software tools

Since the DNA sequencing was started, a lot of software tools to analyze them in computers have been accumulated. These software tools consist of nucleotide or amino acids sequence pattern search (Bairoch 1989), prediction of protein secondary structures (Chou and Fasman 1978, Garnier et al. 1978, Garnier and Robson 1989, Kyte and Doolittle 1982, Lupas et al. 1991, Eisenberg et al. 1984), sequence homology search (Burge and Karlin 1997, Karlin et al. 1992, Pearson and Lipman 1988), protein 3D structure viewer (Sayle and Milner-White 1995) and others (Henikoff and Henikoff 1991, Henikoff and Henikoff 1992, Quandt et al. 1995, Riley 1993, Wootton and Federhen 1993) (see **Table 1.1**). Most of them are still used commonly among the molecular biologists and information scientists. The most widely used one is the BLAST program which was developed in NCBI (Altschul et al. 1990), and implemented with a powerful homology search functions. ClustalW (Higgins and Sharp 1988, Thompson et al. 1994) is also one of such software tools, which provides multiple alignment facilities of nucleotides or amino acids sequences. Thus, there are a lot of excellent algorithms which had been developed in early days of bioinformatics. However, most of such software tools had been developed in a command line style and most of potential users today will meet difficulties to run them on their PC.

Two solutions are possible to overcome the difficulties. One solution is that these software tools themselves have a web-based user interface for more comprehensive operations. Actually, there are web-based user interfaces, for ClustalW, GeneScan (Burge and Karlin 1998) and BLAST. Web-based user interfaces of the existing software tools are further comprehensive compared with the previous command line operation. However, it is not convenient when users try to analyze their data consecutively with the different algorithms. In addition, data is not tightly secured because of the internet communication is necessary to use the web-based facilities.

Another solution is that these software tools are incorporated into a front end software package which provides standardized user interfaces for their input commands and output results presentation. A front end software package runs the existing software algorithms and visualizes the results with graphical interface. GCG (Genetic Computer Group) Wisconsin Package and Staden Package (Gleeson and Staden 1991, Staden 1996) are the first representative cases of the front-end software interfaces. However, GCG Package has limitations in handling of genome-scale sequences, and thus it is out

of date. Further-more, in GCG Package, all the algorithms were written in FORTRAN language. Consequently, it required a lot of development time and efforts to modify the algorithms. In addition, a DBMS (DataBase Management System) is required to run GCG. Therefore, most of the biological data to be analyzed by these packages is necessary to be converted into a particular format and stored as entries of the database.

Table 1.1 Traditional and widely used sequence analysis programs

Object sequences	Programs Names	Functions and Features	References
DNA	Seg	Masking of low-specific region	Wootton and Federhen 1993
DNA	BlastN	Homology search against nucleotide datasese	Altschul S.F. and Gish W 1990
DNA	Repeat	Detection of repetitive sequences	Wiconsin Package Program Manual
DNA	FastX	Homology search against amino acids database	Pearson and Lipman 1988
DNA	BlastX	Homology search against amino acids database	Altschul S.F. et al. 1997
DNA	GenScan	Prediction of protein coding reations	Burge, C. and Karlin, S 1997
DNA	MatInspector	Prediction of promote regions	Quandt K. et al. 1995
DNA	tRNAscan	Prediction of transfer RNA	Lowe T. and Eddy S.R. 1997
Amino Acids	PepPlot(GCG)	Prediction of secondary structures	Chou and Fasman 1978
Amino Acids	PepPlot(GCG)	Prediction of secondary structures	Garnier et al. 1978
Amino Acids	PepPlot(GCG)	Prediction of hydrophilic regions	Kyte and Doolittle 1982
Amino Acids	PepPlot(GCG)	Hydrophobic moment	Eisenberg et al. 1984
Amino Acids	Blinps	Motif search on BLOCK database	Henikoff et al. 1997
Amino Acids	Prosearch	Motif search on Prosite database	Bairoch et al. 1997
Amino Acids	SPScan	Prediction of secretable peptide signals	McGeoch D 1985
Amino Acids	HTHScan	Prediction of Helix-turn-helix regions	Claverie, J.M. and Audic, S, 1996
Amino Acids	CoilScan	Prediction of supercoiled regions	Lupas 1996
Amino Acids	SOSui	Prediction of transmembrane helice	Hirokawa T et al. 1998
Amino Acids	FAMS	Homology modelling tool	Ogata K. and Ujneyama H. 1999
Amino Acids	PSI BLAST	Homology search with low similarity	Altschul S.F. et al. 1997

(2) Large accumulation of data resulted from high throughput experiments

Since the genome projects had started in early 1990s, exponentially increasing amount of DNA or amino acids sequences data has been accumulated in the international databases such as DDBJ/EMBL/GenBank. Further-more, huge biological data such as the results of micro-array experiments are also accumulating in the databases such as MIAME (Brazma et al. 2001, Ball et al. 2004, Stoekert et al. 2002, Spellman et al. 2002,). Integrated handling of both biological sequences with annotation and newly introduced experimental data are much required for the software environments in genome era.

(3) Data format for sequence analysis software tools

GenBank/EMBL formats are widely used for the description of DNA and

protein sequences with their annotation. Therefore, using the formats for the input and output of sequence analysis software tools is advantageous for data portability or visibility. However, current data description convention of GenBank/EMBL formats has limitation for the description of results in novel technology such as DNA array. In addition, when performing in silico experiments with GenBank/EMBL format, additional information is necessary. Thus, further extension to the current convention is required.

(4) Continuous advancement in the hardware and software technology

There is non stop advancement in the hardware technology in PC. Every year its performance is improved in CPU speed and memory size. It is said that the latest PC models have superior performance than the supercomputers of a decade ago. Because of the advanced PC hardware, any user can run software which had been impossible to run except in a large size supercomputer previously. Thus, today, large-scale computing of genome data is achieved with small PC level machines

Java language was invented by Sun Microsystems about ten years ago. As the technology was late-coming one compared with traditional developing languages of C, Perl and others, the Java has been furnished with integrated and superior functionalities compared with previously existing languages, ranging from web application capabilities, drawing functionalities, fast string handling and coding error detection. When Java was first introduced, the OS had been too heavily loaded and Java was not suitable for applications which require quick responses. However, advancement in computer hardware had gradually solved this problem. Recent version of Java provides excellent performance level.

Molecular biologists and medical scientists are famous about their preferring of Macintosh computers. By using Java language, efforts to provide multiple versions compatible to Windows, Mac and Unix, are much reduced for the developers of molecular biology software. Under the situation, development of molecular biological software tools by utilizing the Java technology has advantageous for up-to-date release of new functions to multiple platforms..

1.3 Concepts of software environments for molecular biology

I here propose that a concept of the software environments which cover the field of molecular biology. I summarize the main points of my concept for the environments as follows. (1) The software should have ability to handle the largest entries in the international nucleotide database. (2) The software should have compatibility between Windows, Mac and Unix. (3) GenBank/EMBL format should be utilized to ensure the data visibility and portability. (4) The software should have a user interface, as front end software, which has ability to run most of the existing software inheritances. (5) Graphical interfaces to describe invisible phenomena of molecular biology are required. (6) The software should have an *in silico* experimental facility to perform consecutive routines of cloning experiments.

(1) Handling of the largest entries to GenBank/EMBL

It is important to have ability of handling sequences of whole chromosomes, such as those of human genome. The largest chromosome of the human genome contains about 250 mega base pairs and numerous features. Non stressed handling of such a huge chromosome sequences is needed. In addition, editing of sequences and features is also necessary.

(2) Machine compatibility

As for the bioscience software tools, machine compatibility between Mac, Windows PC and Unix is required. Java language satisfies the required compatibility for simultaneous distribution of multiplatform software tools.

(3) GenBank/EMBL format

Integration of *in silico* experiments or DNA array into GenBank/EMBL format files, provide seamless handling of all the data. Especially, when performing *in silico* experiments, single data entries per molecule are necessary. Therefore, direct and sole handling of the GenBank/EMBL format files for reactions such as restriction enzyme digestion, PCR or ligation, should be implemented.

(4) User interfaces as a front end software

A front end software stands between its user and various programs such as BLAST, ClustalW and so forth. The user runs only the front end software to activate

these programs and access to the existing biological data. For example, when the user would like to know if there are homologous sequences to the gene what the user is viewing, what the user has to do is only clicking on the feature of gene and the interface program activates BLAST programs without knowledge of BLAST operation. In addition, the results from BLAST are also converted to graphical images, therefore it is not always necessary for the user to learn about its output format. These results are also incorporated into the annotated sequence file which the user is now viewing so as to be referred later. The conceptual diagram of the front end software is shown as **Figure 1.1**.

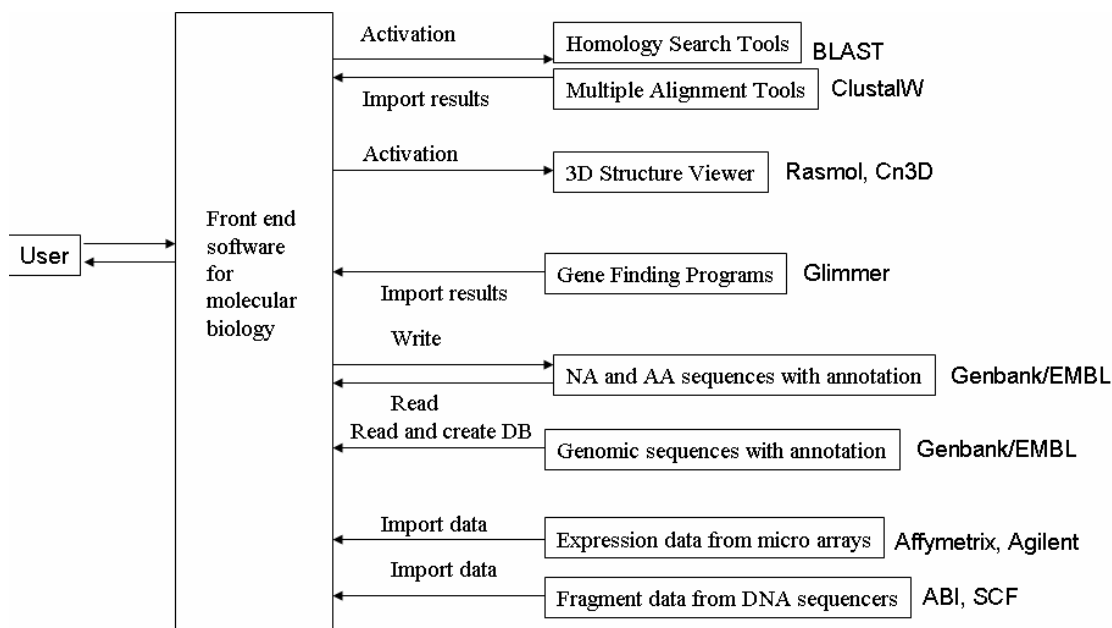


Figure 1.1 Conceptual diagram of a front end software tools for molecular biology

(5) Graphical viewers to describe invisible phenomena of molecular biology

When understanding molecular biological phenomena, most difficult matter is their invisible nature. To assist for understanding of the phenomena, graphical user interfaces should be implemented.

(6) Sequential and uninterrupted operation of sequence analysis

Biological sequence analysis usually requires consecutive operations of basic analysis tools. Most of software tools provide a single processing of input data, namely the input data are read and processed and the results are presented by particular formats. Consecutive operations are usually realized by making the output of the previous program to be the input for the following program. If the two formats are identical, these operations will be succeeded. The point is that in the consecutive operations of

different programs, all the programs participated in the operations, are required to share only one format for inputs and outputs (**Figure 1.2**). When considering about the biological sequence data, the shared format must be that of GenBank/EMBL annotated file which contains nucleotide or amino acid sequences and various information about the sequences. If an extension to the GenBank/EMBL format convention is possible, consecutive sequence analysis will be implemented.

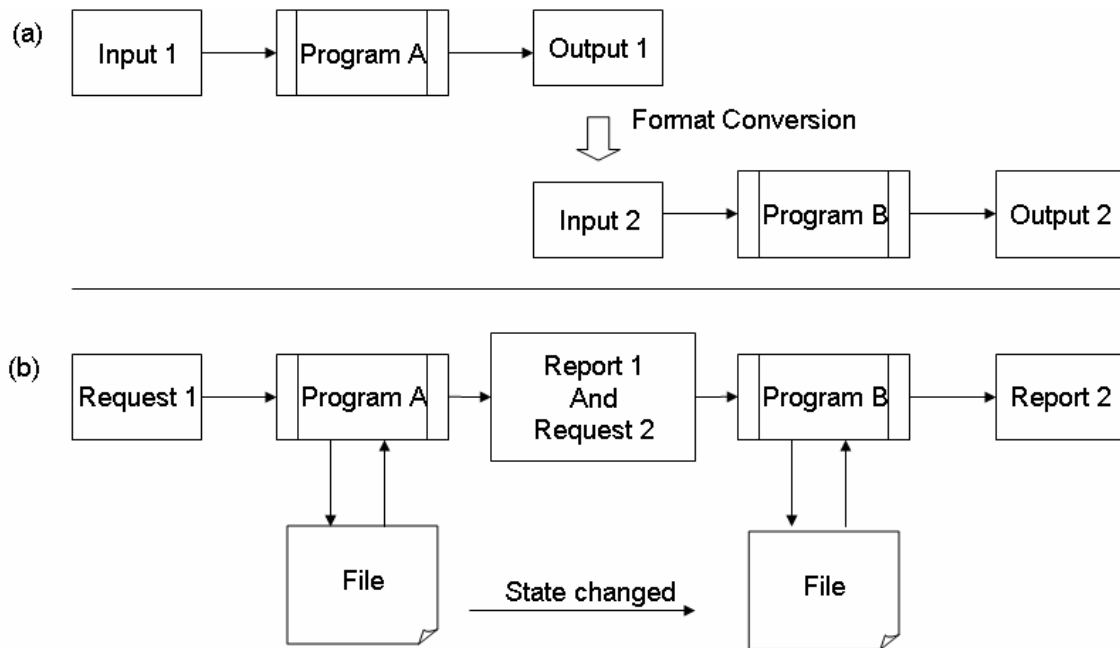


Figure 1.2 Consecutive operations of programs. (a) A single I/O programs can be consecutively operative if the output format of program A has same format with the input format of program B. If these two are different in format, a format conversion is necessary. (b) If all the programs share same format for their inputs and outputs, these programs can be performed consecutively.

1.4 Overviews of the thesis

I had developed two software packages for molecular biology, MetaGenomeGAMBLER (MGG) as a supporting system of microbial genome sequencing and *in silico* MolecularCloning (IMC) as a biological sequence analysis system, on which I achieved consecutive operation of the programs, especially that of *in silico* experiments. In addition, I implemented functions of handling huge expression data of tiling microarray in the second package.

To describe the above achievement, this thesis is divided into six chapters as shown in **Figure 1.3**.

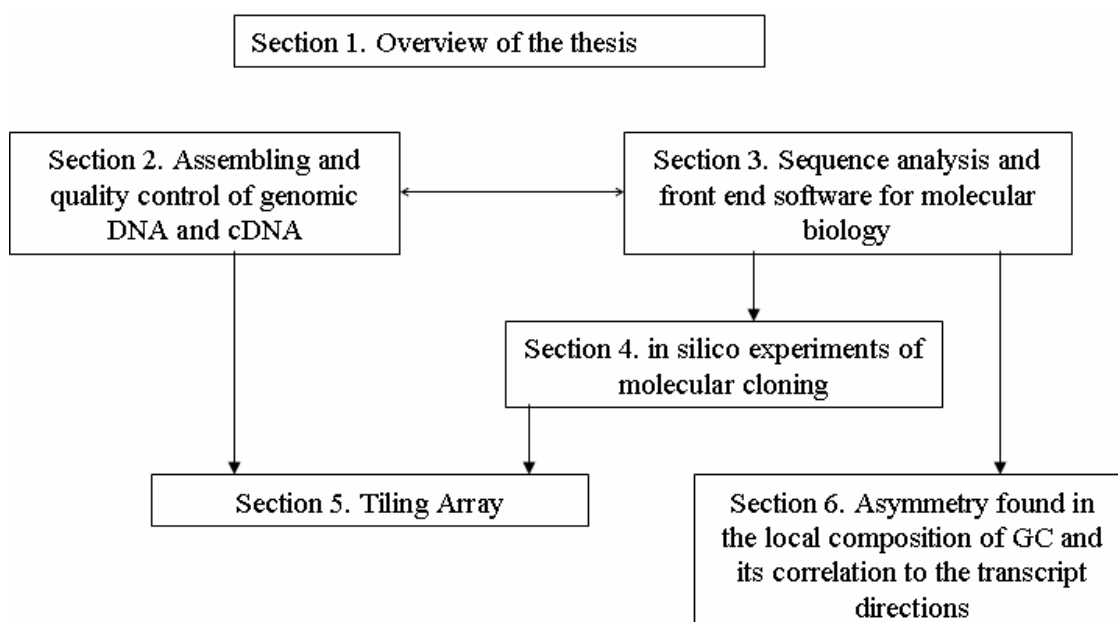


Figure 1.3 Diagram of relationship between sections in the thesis

In this **first chapter**, the motivation, backgrounds, and concepts of the software tool are described.

In the **second chapter**, microbial genome sequencing support environment, MGG, and its implementation is discussed. Assembling software tools are essential to genome sequencing project, and high-speed and precise data handling is also important. In MGG, I implemented all-in-one handling and analysis of a microbial genome in a small PC or Mac.

In the **third chapter**, sequence analysis software environment for genomic era, IMC, is described. I designed and implemented the environment which accommodates well-used software tools and provides comprehensive and graphical presentation of the results. Handling and editing of large chromosomes such as human chromosome No.1 is achieved in the software.

In the **fourth chapter**, *in silico* experiment software tool is described. This function is the part of IMC. A new data formalization to achieve consecutive operations of molecular cloning experiments is described. End type description is proved to be effective to perform digestion and ligation of DNA molecules.

In the **fifth chapter**, incorporation of the tiling array data handling and analysis functions into IMC is described. The software is enhanced with a unique data structure. The tiling array is a high-density DNA array with huge number of probes on it. However, suitable software tools to handle and analyze its expression data are not available previously. I describe how the software tool combines the nucleotide sequences and its annotation against tiling array expression data by extending the GenBank/EMBL database convention.

In the **sixth chapter**, as one of the case studies about the above-mentioned software tools, I found and discuss that transcription-coupled GC skew is observed on the genomes of *Clostridii*, *Fusobacterium* and others.

Chapter 2

Assembling and quality controls of genomic DNA and cDNA

2.1 Introduction

It is used to be said that the era of genome sequencing had passed and the post-genome era (Chait 1996) had come after the completion of human genome sequencing. This affected the decline in the efforts of developing sequencing project support systems such as quality controls of DNA sequence raw data, DNA assembler and genomic sequence annotation tools. However, genome sequencing projects are still necessary especially for microbial researches. Biological diversity has been shown to be wide and has further complexity than previously anticipated. Thus, the requirements for the sequencing support software tools are still large.

High-throughput sequencing of short fragments by DNA sequencers are accomplished due to the fast advancement of devices or reagents as multi-capillary DNA sequencers (Ansorge et al. 1987, Luckey et al. 1990, Lee et al. 1992), or high performance DNA polymerases (Peterson 1988). On the other hand, in the past decade, the information to be referred during sequencing projects, has been exponentially increased, and the shortage of software makes latter half of the projects, namely quality controls, assembling, and annotation, to be a bottleneck for the sequencing projects. The work flow of a genome sequencing project is shown in **Figure 2.1**, covering from DNA fragment sequencing to annotation.

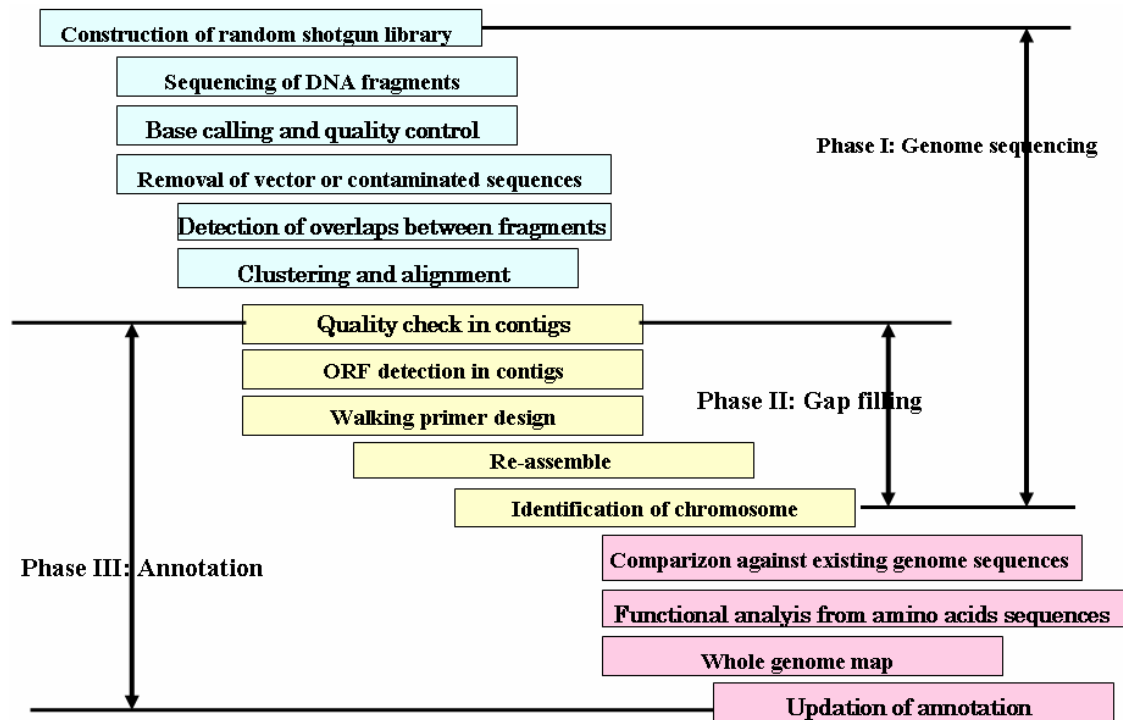


Figure 2.1 Flow diagram of genome sequencing to annotation. The boxes painted in blue are the processes in the sequencing phase, the boxes painted in light brown are the process in the gap closing phase and boxes painted in rose color are the processes in the annotation phase.

All-in-one implementation of all the functions of microbial sequencing support is necessary to achieve smooth operations of numerous data related to genome sequencing. Previously, this burdensome computing had been performed in high-performance remote servers. Only small part of the huge sequencing data is transferred to client PCs for editing or viewing the results because of the limitation of data transmission facilities between servers and client PCs. All-in-one implementation entirely solves this obstacle due to the no transmission any more if all the data are stored and processed in one PC or Mac.

A graphical and comprehensive operation of quality controls of sequencing raw data and integrated handling of numerous results of annotation processes are also necessary. Quality per fragment or even per base is recorded. However, checking all the data by eyes is almost impossible for the largeness of the data. Quality per fragment or per base is rather easy to be identified the low quality regions at a glance.

Automated functions such as detection of low-quality regions, selection of appropriate sequences to be assembled and transferring of such data to annotation phase, are also required because manual handling of these huge data consume a lot of time and efforts.

2.2 Methods and Algorithms

2.2.1 Management of DNA fragments

Even in a sequencing project of a small prokaryotic genome, it is required to manage at least tens of thousands sequences of clone fragments. Therefore high-throughput handling of data is inevitable. In addition, mistakes in naming of clone fragments often resulted in misassembling. To prevent misassembling, clone fragments management is necessary. In addition, automated handling of data is also inevitable.

When handling random shotgun fragments, each fragment has same weight of importance. It means every fragment must be handled equally. The fragments are referred several times during its sequencing project. (1) When imported, its file name is parsed and extracted information from the file name. (2) When quality checked, every fragment is called if qualified as quality checked, vector checked and trimmed. (3) When classified as a member of assembling, the specified fragments are transferred to an assembling unit. (4) After assembled, each fragment is retrieved again to be checked its quality. (5) For contig linkage check, each fragment is retrieved to be checked about extracted information of mate partners. (6) In all the time, each fragment is retrieved to be shown in the trace viewer.

(Results) In MGG, the trace data and sequence are stored as files, therefore access to one fragment requires a I/O. This will reduce the memory space, while require more I/O time. On the contrary, PHRAP assembler stores all the fragment information on the memory, and sometimes it can not work any more if the memory shortage occurred.

2.2.2 Extraction of information from fragment file names

Naming rules on the sequencing of clone fragments are not standardized. Therefore each sequencing team is applying its own naming rule. Simple naming is also used in certain projects teams and serial number is assigned to each fragment. Thus, rather flexible adaptation of the parsing rule is inevitable.

Information extraction from fragment file names is performed as follows. Multiple sets of rules can be registered for differently ruled file names. In one set of rules, four kinds of rules can be set. First is the delimiter rule which denote one or more

character to be treated as a separator of fields. Usually, dot (.) or hyphen (-), and so forth are denoted as the delimiter, followed by the clone id field position, plate id and well id. If plate id and well id are extracted correctly, graphical quality viewer can display the exact position of the fragment in the plate.

In addition, average sizes of clones are also registered. With this information, the distances between different contigs are estimated and hint for primer walking would be given.

2.2.3 Quality evaluation of raw sequencing data

In the early days of microbial genome sequencing, when a small DNA fragment, manual viewing of the chromatograms and removal or editing of low quality regions of them were performed. However, in a whole microbial genome sequencing project, the number of clones amounts to tens of thousands, and quality checking is an exhausting work even by talented personnel. Therefore, today, majorities are employing somewhat automated methods without visual and manual works has been developed, for screening low quality regions on the fragments.

The primary output from a DNA sequencer is accompanied by chromatogram data for fluorescent intensities corresponding to four bases of Thymine, Cytosine, Adenine and Guanine. The chromatograms are used for basecalling which determines the base code for the each peak formed by the four color curves. Green quantized the quality of the basecalls by using a statistical algorithm. The software tool, named PHRED (Ewing 1998A, Ewing 1998B), has been implemented with unique ability of providing absolute error rate per base. Through this method, one can compare quality between the sequences of different manufacturers of DNA sequencers. By this advantage, PHRED software tool is used commonly among the genome sequencing teams. The base score of PHRED is derived as follows. If there is a chance of error of one base per 100 bases, the PHRED score is denoted as $QV=20$, where the QV (abridged for quality value) is derived in the following formula (**Eq.2.1**) (where p is defined as the absolute error rate).

$$QV = -10 \times \log p \quad (2.1)$$

However, this error rate p is not obtained initially. Therefore he used the number of sample data from the actual sequencing data. Although, the borderline for the

error rate (P_{base}) for acceptable affordances is different from case to case, it is actually obtained by **Eq. (2.2)** using object error rate (P_{final}) and mean average covering rate (R) which is derived from number of clones to cover a whole genome.

$$P_{base} = P_{final} \times R \quad (2.2)$$

For example, let us make an objective of the error rate of one per 10,000 bases along a whole genome sequence. If the genome is covered by clones with average 10 fragments redundancy, the error rate for a single fragment should be controlled within one error per 1000 bases. According to this rule, the bases which have lower quality than $QV=30$, should be removed from the raw sequences.

I proposed much simpler and relative method of sequence quality evaluation. The new quality check algorithm is described as follows. A base quality is defined as a product of two values. One is the difference between the top and the other is the average difference between top peak and neighboring valleys. With multiplying by a coefficient, the score is calculated (**Eq.2.3**).

$$QS = A (TP - SP)(TP - (LV + RV)/2), \quad (2.3)$$

where QS is defined as quality score, TP as top peak intensity, Sp as the second peak intensity, LV as left valley intensity and RV as right valley intensity of the measurement point, and A is a coefficient.

2.2.4 Screening of vector and host genome contamination

Another issues related in the quality control of raw sequencing data are detection and removal of contamination of xenogenic sequences such as those of the vector or host chromosomes. Among such contaminations, the most disturbing problem is those from the vector sequences used for making sequencing clones. Typical vector removal software is CROSSMATCH (Green 1999A, Green 1999B).

A new method of vector sequence detection is applying homology search against possible vector sequences. After detecting homologous sequence to the vectors, the region will be assigned as the vector sequence. However, in case that the dynamic range of homologous regions are wide in length, only single performance of vector

removal is not adequate and repetitive vector sequence removal with different parameters are required. When detected a longer vector sequence, the end of alignment are becoming ambiguous because the quality of the cloning site is usually not good compared with the other regions of the clone fragment. Therefore, there is a possibility of remaining short vector sequence even single scanning of vector by the above method. To avoid the remains of short vector sequence, the method employs a repetitive process with changing the parameters of sequence homology analysis.

2.2.5 Trimming of low quality sequences and vector contaminated regions

As low quality regions or xenogenic contaminated regions should not be used when assembling, then either actual removal of bases with low quality or merely masking of the bases are required. After classifying these contaminated regions, trimming of contaminated sequences are performed in a single operation. When it comes to handling of repetitive sequences, the regions are masked in the same manner. However, these masked sequences are removed from the mask after the fragments are correctly assembled. Therefore, there are two types of making data handling.

2.2.6 DNA fragment assembling

A number of fragment assemblers have been developed in this twenty years. Before the genome sequencing projects, assembling of shotgun fragments of only cosmid or lambda sized clones were adequate. Among such assemblers, *bap* (X*bap*) by R.Staden or *GelAssemble* of GCG obtained popularity in 1990s (Batzoglou et al. 2002, Bonfield et al. 1995, Iris 1994, Myers et al. 2000, Waterman 1995, Weber and Myers 1997). One of the assemblers runs on Macintosh, named *Sequencher*, has been also widely used. However, since the first prokaryotic genome was wholly sequenced at TIGR in 1995, whole genome shotgun method has been employed and these software tools can not afford to assemble it because of large number of fragments. Thus, assembler with ability to assemble tens of thousands of fragments was required and TIGR has developed the TIGR assembler (Sutton 1995). At the same period, at University of Washington, a random shotgun assembler named PHRAP was developed (GREEN 1999). The PHRAP assembler has been widely utilized by microbial sequencing projects. When human genome sequencing had started, Celera Genomics had developed an assembler named Celera Assembler (Myers et al. 2000) to support whole genome shotgun assembling at the company.

Overview of the assembling process is given as follows. The objects of assembler are DNA fragments after screening low quality regions or xenogenic regions from the raw data. Tens of thousands of such fragments are required to reconstruct the whole prokaryotic genome sequence. In the first stage of the assemble algorithm, (1) detection of overlaps between all the pair of fragments are performed with the round robin method. (2) If an overlap is detected between any two fragments, it is interpreted as the two fragments are both originated from overlapped and adjacent regions of the genome. (3) The detection of overlaps among the tens of thousands of fragments requires high-speed algorithm to compute hundreds of millions order sequence comparisons. The BLAST algorithm is fast enough to be used in assembler. In addition, multiple CPUs, such as those of PC clusters, help to accelerate such time consuming computation. (4) The results which are derived from this computation are overlap length in bases, homology scores, directions of each overlap.

In the next stage of assembling, by using these overlap detection results, clustering is performed. The clustering algorithm is as follows. All the bi-directional best hit pairs are listed, here the bi-directional best hit pair is defined as the any two of the fragments found in a set of fragments where one of the two has most homologous to the other among all the fragments in the set and *vice versa*. However, when the overlap length between the two fragments is smaller than the previously set parameter, it is not accepted as a bi-directional best hit pair. Allowance of too short length for overlap detection may include accidentally overlapped fragment pair, namely, the two originated from the absolutely different regions of the genome. That is, a short and true overlap should be allowed to be clustered while a long and false overlap should be expelled. In the assembler originally developed this time, overlap length of 15 to 30 bases are usually adopted. In addition to this parameter, the homology score itself is also required. In the homology search algorithm, low homology score is allowed if its user requests. In the case of this assembler, the score is 90 to 95 %. These bi-directional best hit pairs are used as core clusters. Next, the rest of fragments which are not included in the bi-directional best hit pairs are examined if it had single path best hit to any of the pairs and if detected this fragment is joined to the cluster. This process is performed until no fragment is left. The results are defined as final clustering sets. This is the final process in the clustering phase.

In the third stage, alignments in all the clusters are performed. When aligning each member (fragment) of a cluster, if there are accidentally overlapped pairs existing, this leads to contradiction and without disposing some of overlaps, it can not be aligned

at all. These contradictions are not detected previously before alignment, therefore the algorithm must have capability of comparing the score of any possible alignment excluding one fragment by another. However, this computation requires a lot of CPU time, in fact, this is one of NP-complete problems (Waterman 1995), consequently it is impossible to compute it in an exhaustive manner. It is merely possible to compute the multiple alignment in a very small cluster exhaustively. Then, so-called greedy algorithm must be required. In fact, the differences between such assemblers are meant as differences of such greedy algorithms adopted by each assembler.

The consensus sequence obtained from such aligning of the cluster members and represented by the bases of consensus between aligned bases, is called a contig.

2.2.7 Termination of random shotgun assembling phase

When assembling a prokaryote genome fragments, it leads to make hundreds to thousands of contigs generated. Thus, contig generation process is described as a two dimensional plot. For example, if plotted the total number of assembled fragments for X axis and the number of contigs for Y axis, the plot shows the good progress in a sequencing project. The plot graph is also theoretically drawn as shown in **Figure 2.2**.

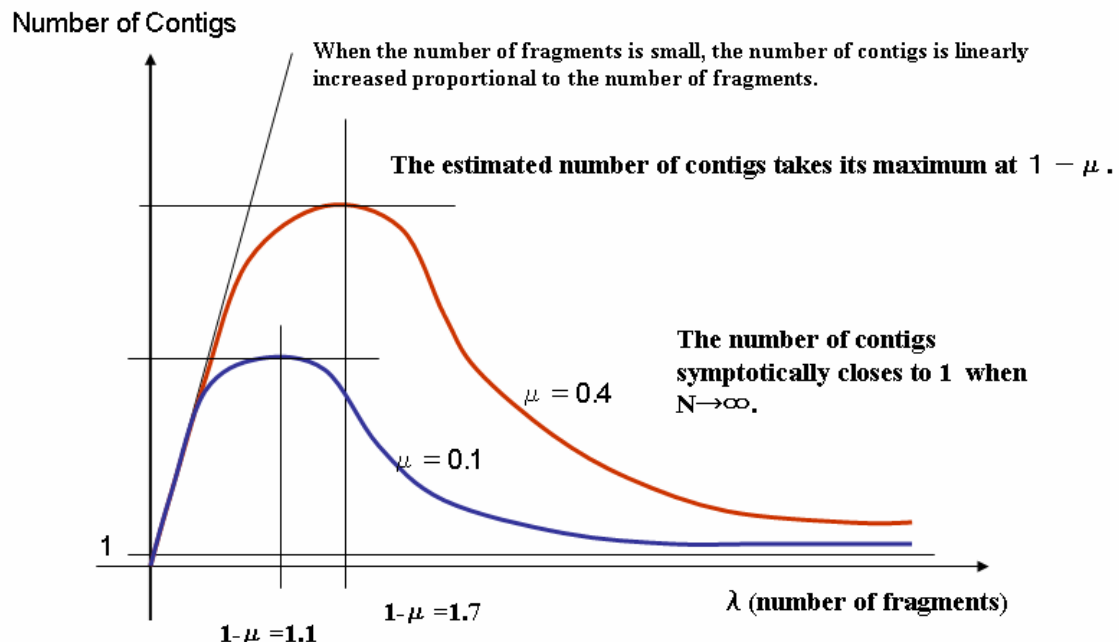


Figure 2.2 Theoretical curves plotted by the number of fragments. X axis denotes the number of fragments. Y axis denotes the estimated number of contigs.

2.2.8 Detecting of misassembled regions of a contig

There are two obstacles in an assembling algorithm. One is the treating of repetitive sequences (Jurka 1994) which are likely to cause misassembling, and another is how to solve the NP-complete problem within a practical time period. There are a lot of repetitive sequences even in a prokaryotic genome, although the number or variety of repeat elements is not comparable with huge sets of those in eukaryote genomes. Nevertheless, these repetitive elements in the genome would be obstacle to obtain a precise assembling result. Actually, if there are higher homologous regions in a genome sequence than the homology criteria of overlap detections, this problem occurs. Transposons, rRNAs, tRNAs and prophages are typical well known repetitive elements in a prokaryotic genome. If, prior to the random shotgun phase, these kinds of repetitive elements are detected experimentally, it is possible to mask out such regions from assembling and to obtain more clear contigs. However, the random shotgun protocols are much simple and in the recent genome sequencing projects, the random shotgun experiments are performed without such previous knowledge about repetitive sequences. After this phase, such repetitive sequences are detected on the sequence by computer programs (Brown et al. 1998, Shavilik 1994).

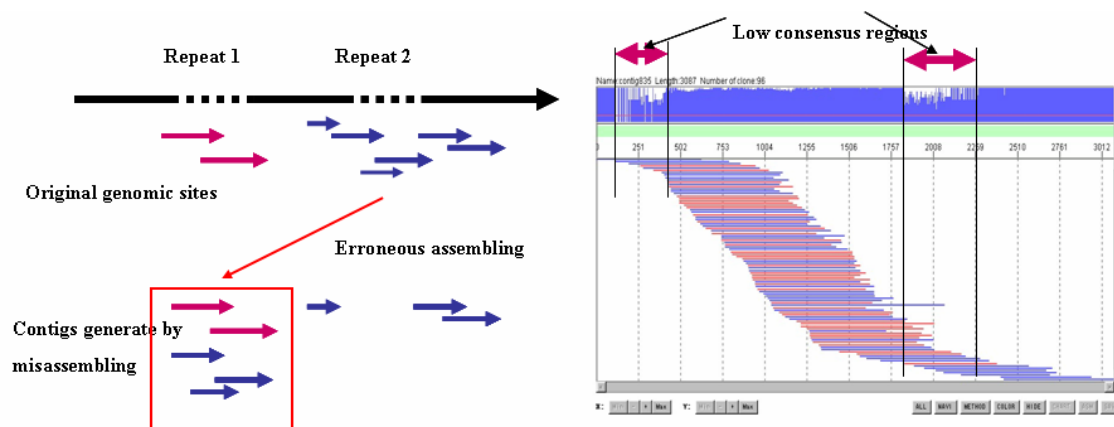


Figure 2.3 Characteristic pattern caused by erroneous assembling. If there are more than two repetitive sites among the genome, when assembled, they are becoming the cause of misassembling. (1) Original genomic sites of repeats. (2) Erroneous assembling. (3) Contigs generated by misassembling.

When it comes to align a contig with repeat, it becomes clear where such repeat sequences are located. If aligned with fragments which include repetitive sequences, the typical resulting alignment map is drawn as shown in **Figure 2.3**. The repetitive sequences which are originated from the different regions of the genome are

clustered as one single site and this creates densely covered region from average coverage of clones. This characteristic appearance of a contig alignment can be used to detect repetitive sequences posterior to the assemble process. In a typical pattern of clustering caused by repetitive sequences, repetitive regions are highly homologous however the consensus sequences just outside the region have no consensus at all. I introduced a simple scoring system of consensus index (**Eq. 2.4**) to detect the repetitive clustering pattern.

$$\mathbf{CI = MB / NF,} \qquad \qquad \qquad \mathbf{(2.4)}$$

where CI is defined as consensus index, MB as the number of fragments which are the majority of population and NF as number of fragments which are joined in the alignment on the base.

Most of misassemble patterns can be detected by monitoring the consensus index along the contig and used as a prediction tool of repetitive sequences and same time as a disassembling tool of such error clustering. The software is implemented with such functionality of automatic detection of repetitive sequences. To resolve such erroneously assembled results, making of repetitive regions for the fragments are usually performed. The masked sequences will be unmasked again after the correct consensus sequences are obtained.

2.2.9 Contig Linkage by Pair Partner Finding

When employing a random shotgun assembling, the distances between contigs are usually unknown. Namely, even if two contigs are actually located in the neighborhood each other with a very short distance, it is not known until the genome is completely sequenced. To estimate the distances between adjacent contigs, a bridging clone method was invented. One sequencing clone can be sequenced by the both ends if there are the priming sites on the vector sequence close to the inserted clone. When the size of clone is appropriately selected, two different contigs share one or more clones with one end sequence belonging to one of the contigs, while the other end sequence belonging to another. Using this information, the distance between two adjacent contigs can be estimated if they share one of mates each other (**Figure 2.4**). The end sequences of such a clone are called mates, and the two contigs which is connected by the mates are called scaffold or unitig (Myers et al. 2000).

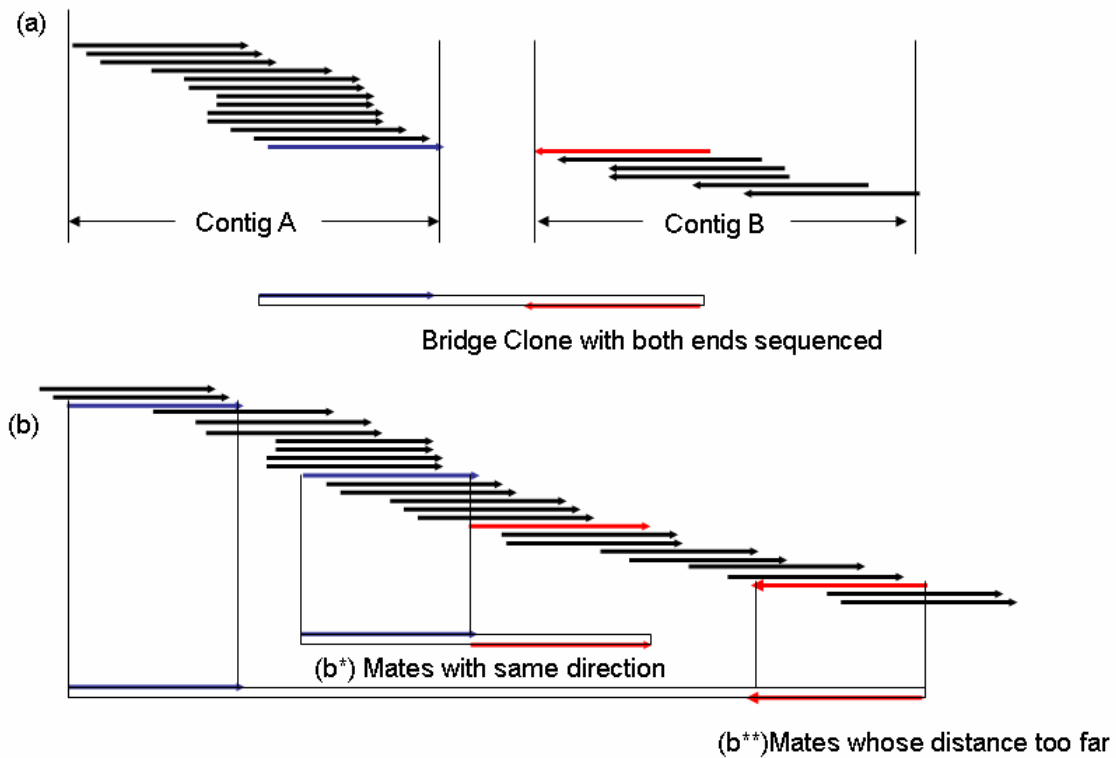


Figure 2.4 Contig connectivity using mates information. (a) A bridge clone has been sequenced from both ends, namely both strands of the DNA. If Contig A has one end of the bridge clone and Contig B has another end, it means that the two contigs share the clone. The average distance of such clones is verified experimentally. Thus, the distance between the two contigs can be estimated as shown. (b) This method is also used for detection of misassembling. (b*) If mates are fallen in a same contig and indicated same direction of read, one of the mates may be regarded as misassembled because mates must have different directions each other. (b**) If mates are fallen in a same contig and the distance between the mates is too long or too short, one or both mates may be misassembled.

In the Celera Assembler (Myers et al. 2000), this method was employed to connect large genome of *Drosophila melanogaster* (Adams et al. 2000) and the method was proved to be effective and automated connection of the contigs can be performed if the error rate of mate match information is lower than 1%. This means that keeping accuracy of mates information is critical for reducing misassembling.

2.3 Results: Implementation of the MetaGenomeGAMBLER (MGG)

I have developed an integrated environment of software tools for microbial genome sequencing projects, named MetaGenomeGAMBLER (MGG). The software is featured with all-in-one functions which are required for the sequencing projects. **Figure 2.5** shows the schematic diagram of the software package. Functions of MGG are divided into three groups. Graphical user interfaces are implemented to provide users with comprehensive operability of MGG functions. Various routines such as Project Manager, Naming Rule Parser, Sequence Importer and so forth, are working as main commands to analyze data. Data handler consists of a set of the basic data management tools.

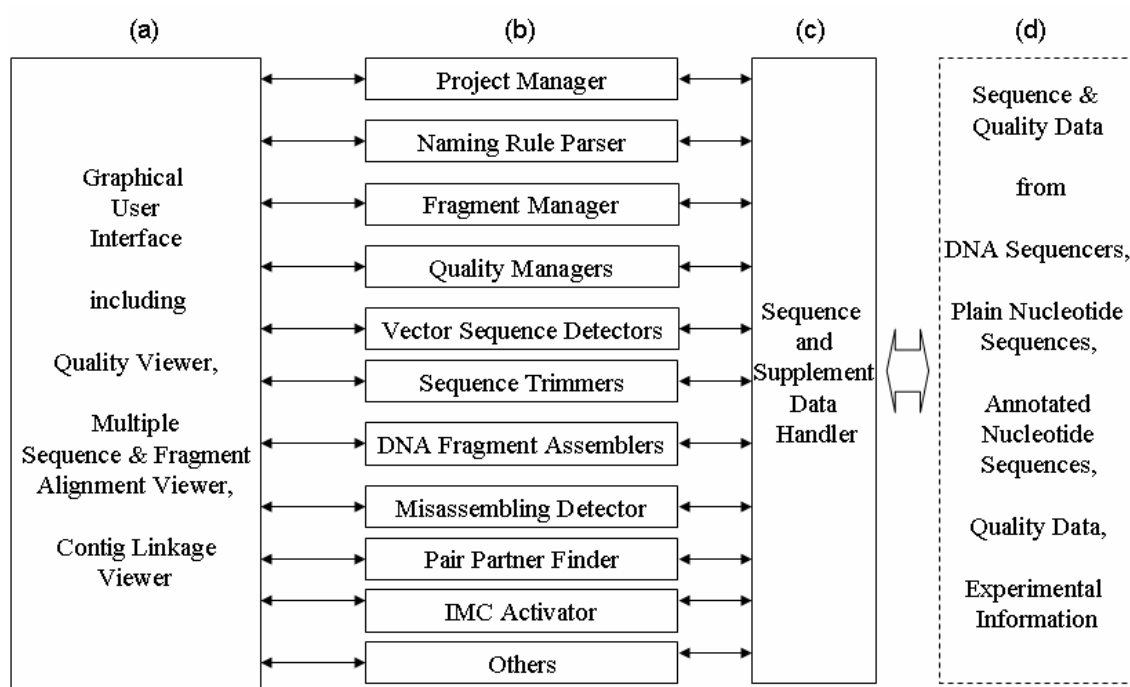


Figure 2.5 Package design diagram of MetaGenomeGAMBLER (MGG). The structure of the software package is shown. (a) On the left, the graphical user interfaces including Quality Viewer, Multiple Alignment Viewer, Contig Alignment Viewer and Contig Linkage. (b) On the middle, various routines are shown, including Project Manager, Naming Rule Parser, Fragment Manager, and so forth. (c) On the right, Sequence and Supplement Data Handler which manages the basic sequence and quality data handling. (d) On the further right, sequence and quality data which MGG can handle is shown.

2.3.1 Structure of MGG system

MGG has wide range of functions from quality control to annotation. These functions are divided into three major categories as follows, (a) Graphical User Interfaces, (b) Data Analysis Routines, (c) Sequence and Supplement Data Handler. The Graphical User Interfaces include (1) Quality Viewer, (2) Multiple Sequence and Fragment Alignment Viewer, and (3) Contig Linkage Viewer. Data Analysis Routines consists of (1) Project Manager which manages assembling data and results, (2) Naming Rule Parser, (3) Fragment Manager, (4) Quality Managers which evaluate the quality of each base in a fragment, (5) Vector Sequence Detectors which detect vector and host sequences in a fragment, (6) Sequence Trimmers which trim vector sequences and low quality regions in a fragments, (7) DNA Assemblers which assemble many fragments into contigs, (8) Misassembling Detector which detects misassembled regions from a contig, (9) Pair Partner Finder which connect the both ends of the same clone, and (10) IMC Activator which activates IMC (*in silico* MolecularCloning; see **Chapter 3**) for further annotation works. Sequence and Supplement Data Handler manages sequence conversions and data rearrangements between various format data. Some of these functions are also automatically processed if necessary.

2.3.2 Data Analysis Routines

(1) Project Manager: Project management of assembling data

I here define that assemble is a unit of routine to assemble a given DNA fragments, and a project is a set of different assembles selected from . When assembled, exchanging of the set of the DNA fragments which are participated in an assemble unit, often occurs, and assemble itself is repeated until a good contig set is obtained. To satisfy these requirements, the concept of project and assemble is adopted in MGG. An assemble unit is an execution unit of assembling process, for each assembling execution, one assembling id and working area for assemble are newly assigned. Multiple assembles are belonging to one project, therefore the project is defined as the unit of the assembling. Even if using same fragment set, the purpose of assembling may be different. In such case, more than one project should be registered for a same set of fragments. For each assemble, the results are stored in the same place. And for each project, number of repeated trial of assembling are included. To operate this management process, MGG provides the project pane for handling of numerous assembles and projects. One click on a project node on the project pane of MGG, open the assemble list which are belonging to the project. Then after selecting an existing or

creating a new assemble and selection of a set of fragments to be assembled, assembling is executed immediately. It will take time to finish a large set of fragments to be assembled. To monitor the progress of assemble, a progress message is shown during assembling execution.

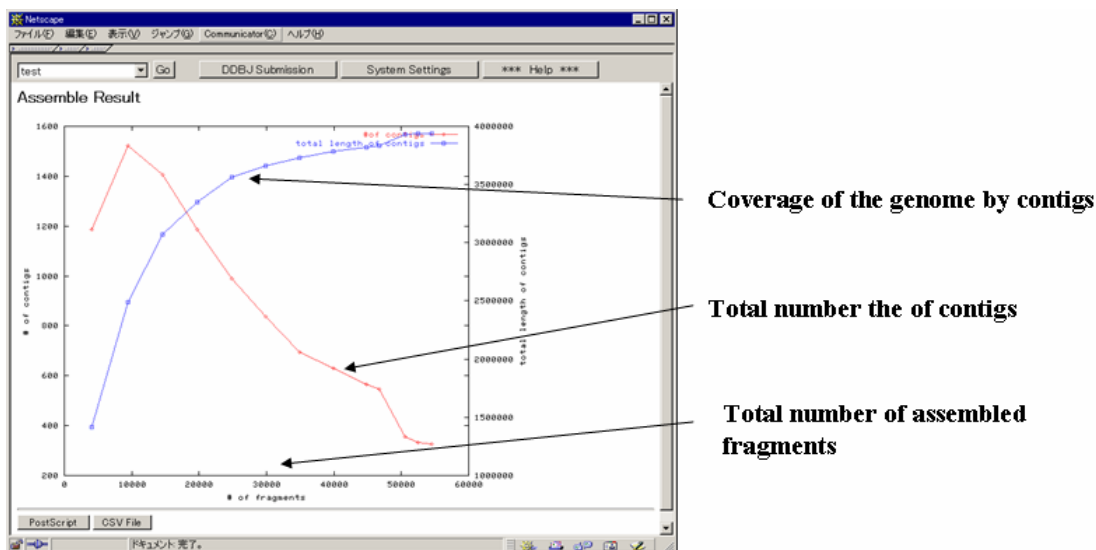


Figure 2.6 Plot of contig generation and coverage of the genome. This graph indicates the number of fragments participated in assemble as X axis while the number of contigs and same time coverage ratio as Y axis, and is used for decision assistance of termination of random shotgun phase.

As mentioned in the **Chapter 2.2.7**, termination of the random shotgun phase is determined by monitoring the plot of contig generation and coverage of the genome sequenced. MGG is implemented with this function by a graphical presentation. Usually, in this plot (**Figure 2.6**), the line indicating the number of total contigs generated has a peak point when the number of fragments reached certain level, and then declines gradually until the infinite limit of one contig. On the contrary, another line which indicates the genome coverage ratio, shows a monotonic increase to the infinite limit of 100%. It takes an infinite time to reach both limit, therefore at one of reasonable level, the random shotgun sequencing should be stopped. This plot can be used to determine optimal timing of the termination of random shotgun phase. After termination of random shotgun phase, more certain method of sequencing such as primer walking is employed to close gaps between the contigs until they are combined into one chromosome. Therefore, it is necessary to design primers to be used to make a bridge

clone across gaps.

When designing such primers for the primer walking, another algorithm is employed. In this algorithm, firstly, after picking one fragment which has no partner fragment in the same contig, the partner's location is searched against any other contig. If the partner fragment is found in any other contig, these two contigs should be located in the clone size distance in the genome. This process loosely combines some number of contigs together with certain ambiguity. In the software, this relationship is drawn in a map including the two contigs. The result provides a good hint to design primer sets for the primer walking. Without this information, randomly selected primers are designed and it takes a lot of PCR experiments. Such walking processes would be repeated until obtaining a single contig finally.

(2) Naming Rule Parser: Information extraction from fragment file names

There are various formats for DNA fragment data, such as ABI format, SCF (Dear and Staden 1992) format, simple FastA (Pearson and Lipman 1988) format, GenBank (Benson et al. 2006) and plain text. Some of them have detailed information about fragments, while others have nothing but plain sequences. MGG can import mixture of such various format files in a single operation. As mentioned previously, naming rules on fragments are still not standardized among the different sequencing teams. To accommodate this situation, MGG is implemented with an information extraction capability from the DNA fragment file names. Usually, typical file name for such DNA fragment is constructed with such as primer direction, type of the clone, plate id, well id in plate, and serial number. These attributes are very important to process assembling projects efficiently. Therefore, the information is automatically extracted from the file name of the imported fragments. The operation window for the naming rules is shown in **Figure 2.7**.

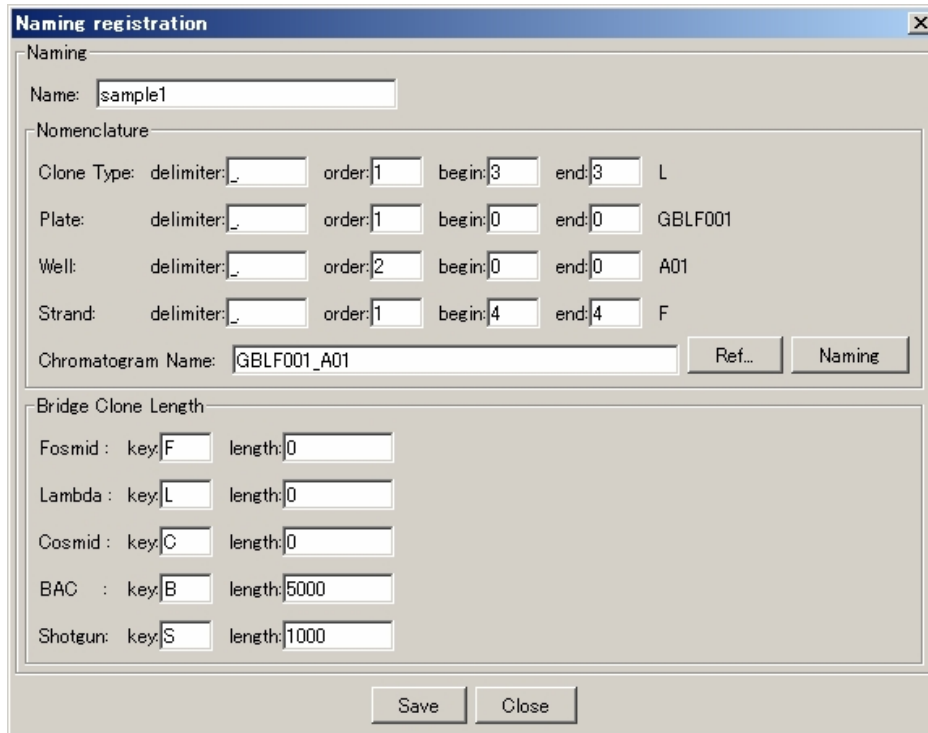


Figure 2.7 Naming rule parser window for extracting attributes of fragments. This window is partitioned by two parts, one is for nomenclature and another is for bridge clone information.

(3) Fragment Manager: Importing and transferring of raw fragments

After the extraction, the attributes are stored for later use. If filename of the fragment are accompanied with plate id and well id, MGG can show the plate format view of the raw data which is convenient for checking data. As shown in **Figure 2.8**, on the import pane of MGG, imported list of fragments are listed on the side pane, while the length of each fragment and the detailed information about each fragment are displayed at top right and bottom right panes.

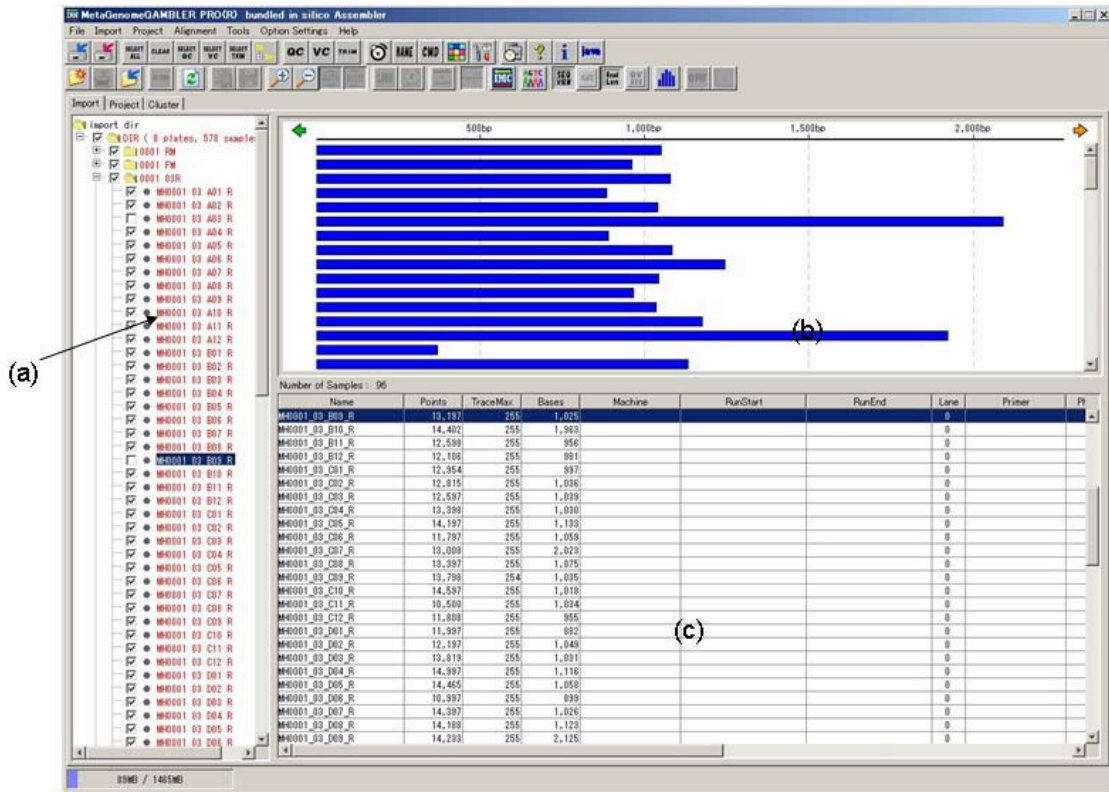
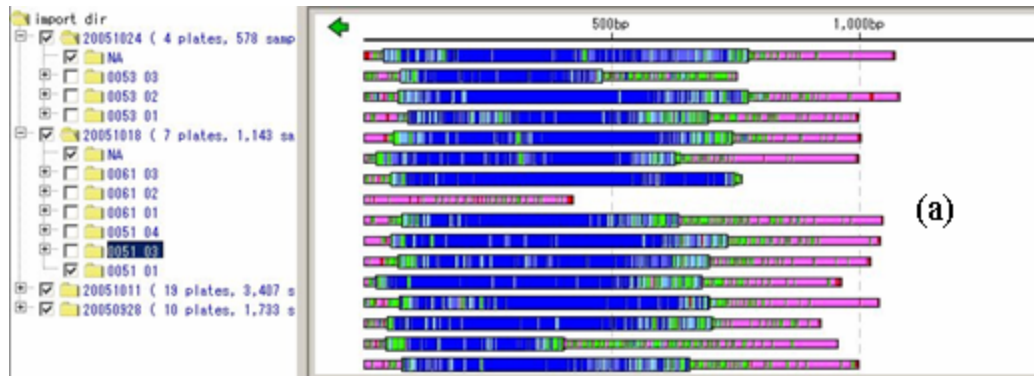


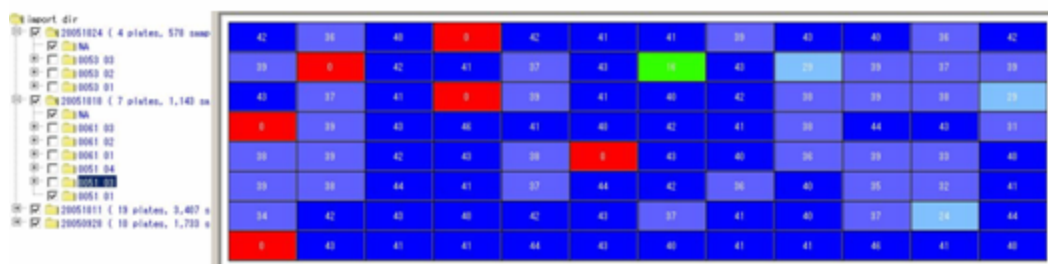
Figure 2.8 Importing window of sequencer raw data. (a) DNA fragments are imported and stored into hierarchical data structure. (b) Every fragment is shown as bar whose length indicates number of bases in the fragment. (c) A list of fragments is shown with various attributes of fragments.

(4) Quality Manager: Quality controls of fragments

In MGG, the quality controls are performed in two ways. If the user has a license of PHRED software, MGG can activate PHRED for the quality control of imported data. If not, MGG activates its own quality check algorithm alternatively. MGG can import the results files of each quality checking tool automatically, and construct them into its own data structure. The Quality Viewer is shown in **Figure 2.9**.



(a)



(b)

Figure 2.9 Quality Viewer, a graphical presentation of quality evaluation. It is important for keep quality of every DNA fragment which participates in assemble. (a) View quality per fragment or quality per base with the viewer. The regions painted in rose or red are of low quality. (b) Quality viewer in a 96-well plate format. The wells painted in red are denoted as that of low quality sequences or absence of data.

(5) Vector Sequence Detector: Detection of vector sequences

MGG provides a detection method of vector sequences with utilizing BLAST algorithm. The vector sequences are registered into MGG through the registration window as shown in **Figure 2.10**. Editing of registered vector sequences is performed on this window.

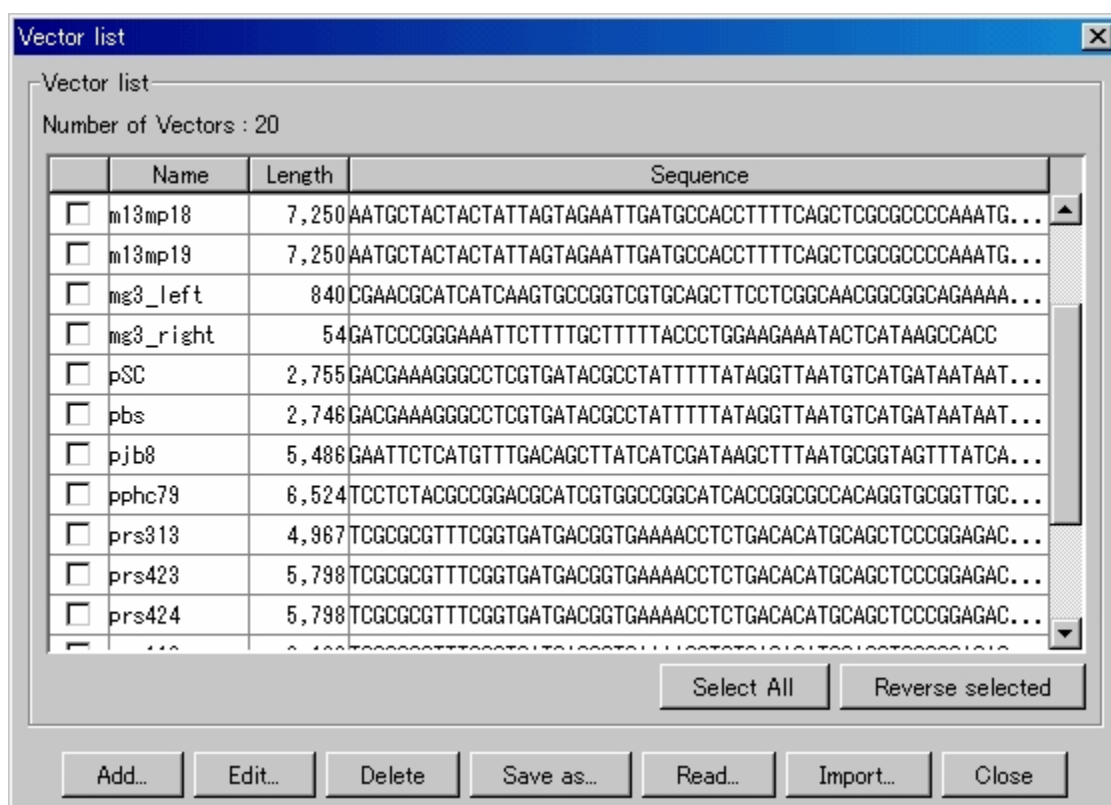


Figure 2.10 Vector registration window of MGG. Name of vectors, their length in bps and nucleotide sequences are listed.

(6) Sequence Trimmers: Trimming of low quality regions and vector sequence

After processed by quality check and/or vector sequence detection, any low quality region or vector contaminated region should be disqualified from participating in the coming assembling process. For this purpose, the trimming function is implemented in MGG. By using this function, low-quality regions or vector masked regions are excluded from assembling. In addition, fixed regions of each fragment can be removed from assembling. Especially, the upstream portion of each fragment is usually contaminated by vector sequences or host sequences, while the last or downstream portion of each fragment is likely to be low quality as shown in **Figure 2.11**

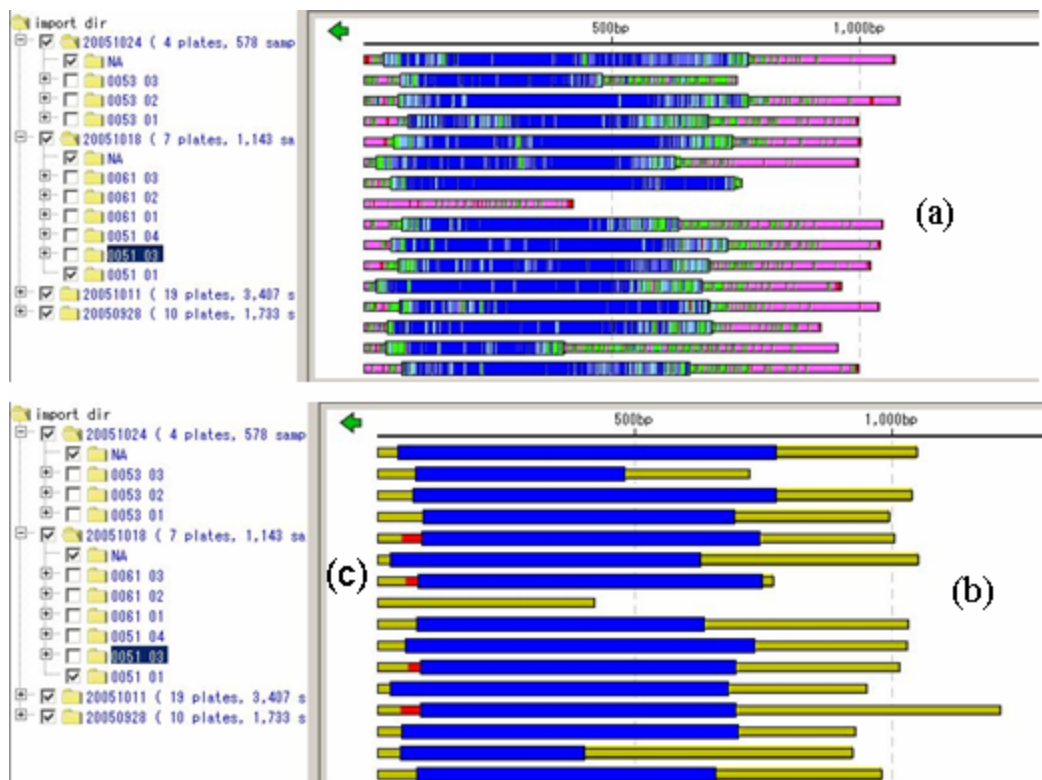


Figure 2.11 Masking controls of raw sequencing data. (a) View quality per fragment or quality per base with the viewer. (b) Masking of low-quality regions, painted in light brown. (c) Masking of vector contaminated regions, painted in red.

(7) DNA Assemblers

Actually, no assembler is incorporated in MGG itself, instead that MGG only activates some registered assemblers and obtains results such as contigs automatically. Thus, MGG and assemblers are regarded as if they were a single identical system. Two of assemblers are currently activated by MGG. One is the PHRAP assembler which was developed by P.Green (Green 1999), another is *in silico* Assembler which I originally developed.

(8) Detection of misassembling caused by repetitive sequences

By utilizing above scoring method, MGG is implemented with manual and semi-automated detector of misassembling caused by repetitive sequences. A profiling graph is implemented on the contig viewer to detect the possible locations of misassembling as shown in **Figure 2.12**. If consensus index goes under the level line, indicated by red bars below the level, this means that the alignment should be visually checked.

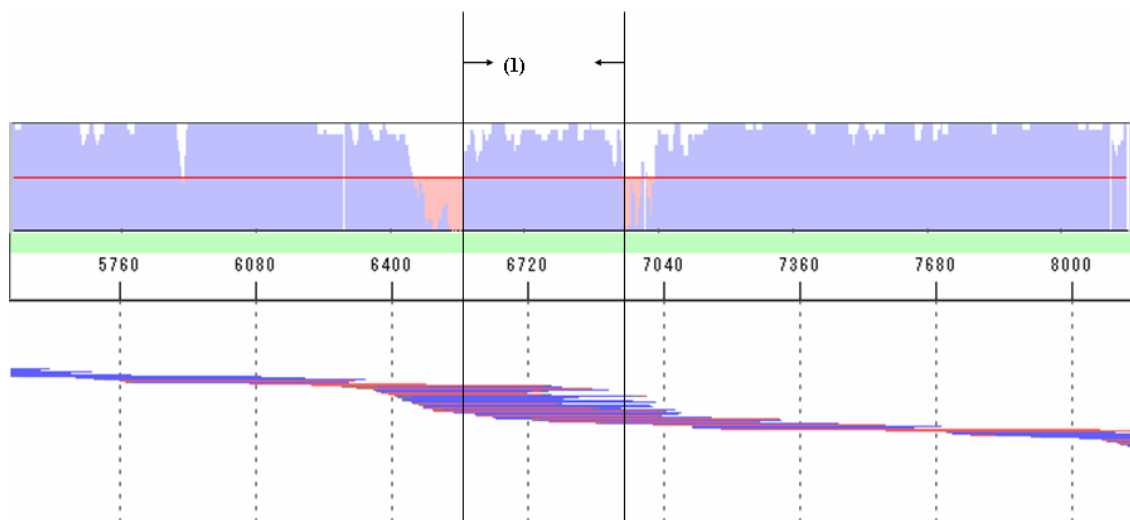


Figure 2.12 Identification of misassembling caused by repetitive sequences. The graph shows that a plotting of consensus index profile along a contig sequence with misassembling. The regions shaded with rose are low consensus regions due to the alignment of different bases. The graph below shows the graphical alignment of all the fragments which is belong to the contig. (1) Cluster of fragment caused by misassembling.

When looking a contig with more globally, one typical contig with some misassembled regions is shown as **Figure 2.13**. IMC provides automatic detection

function of low consensus index regions with a list of contigs which have predicted regions of misassembled. From the list, user can directly jump to the local and detailed alignment, after masking the sequences, following process disassembles the regions and reassemble them without clustering repetitive sequences this time.

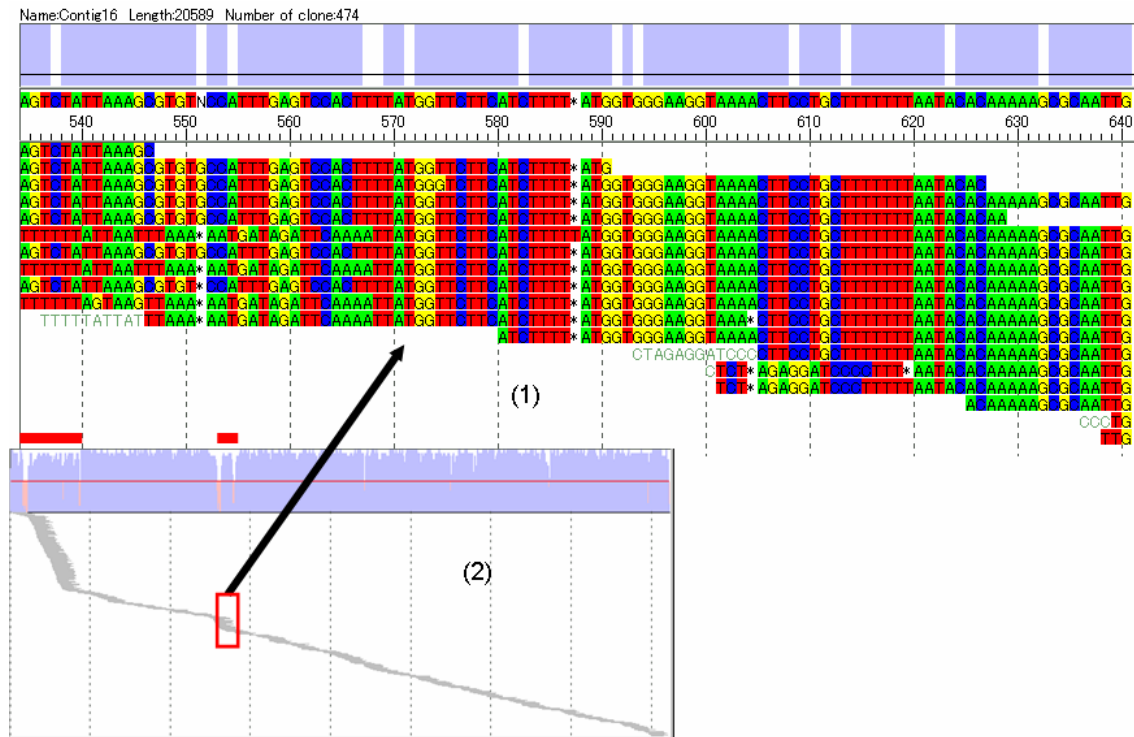


Figure 2.13 Detection of misassembled regions caused by repetitive sequences. (1) Multiple alignment of the contig. (2) Enlarge from, A graphical alignment map of a contig with low consensus regions indicated by shaded with rose color.

(9) Pair Partner Finder: Finding of the partner fragment

At the final stage of random shotgun sequencing, this phase is taken over by usually primer walking phase. In this phase, a number of primer sets are designed from each contig to another. To avoid exhaustive and costly design, prior knowledge of estimated distance between any pair of contigs is required. One random shotgun clone is usually sequenced from both ends of the double strands. This means that there are two fragments originated from one clone, namely, every fragment has its pair partner among many library of shotgun fragments. MGG is implemented with this capability with the following manner. Initially, when a project is registered, the average length of the clone is also registered as parameters. Therefore if the sequencing protocol adopts reading of

both ends of such clone, almost every fragment has its partner fragment with the average length of the distance apart in the genome sequence. After an assemble is terminated, the searching process of all the partner pairs is manually activated and the results are listed and viewed in a panel.

(10) Annotation function with IMC

MGG is not implemented with annotation assistance functions, nevertheless it can activate *in silico* MolecularCloning (IMC), of which is described in the next chapter, and IMC provides the annotation functions. From the project pane, if a contig and its consensus sequence is shown, one click of the IMC button activates the IMC attached with its consensus sequence, then IMC automatically extracts ORF candidates from the simple nucleotide sequence, followed by changing of the feature key to CDS and translate it into amino acid sequence, finally, the homology of every CDS is searched against the previously designated databases such as orthologues(GO 2000, Lee 2002, Overbeek 2000, Tatusov et al. 2001, Uchiyama 2003, Yuan 1998). All the results are stored as qualifiers to CDS features and if previously specified, automatically transferring annotation on the databases to currently annotated sequence (**Figure 2.14**). Multiple contig sequences are also processed in a batch mode, in this case, all or specified set of just generated contigs are becoming the objects of the above automated processes.

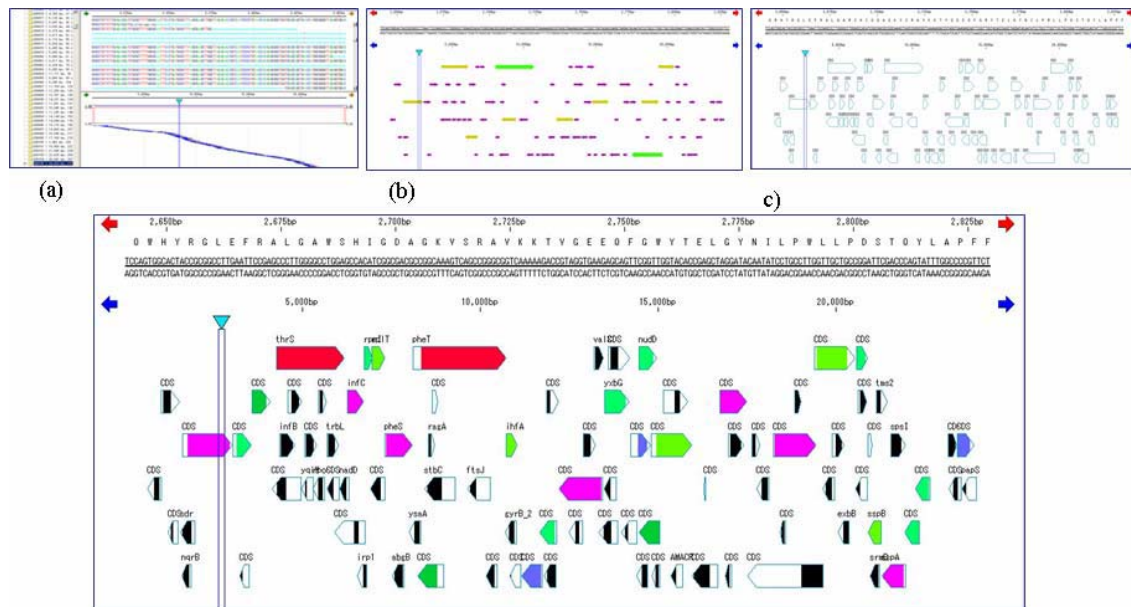


Figure 2.14 Automatic annotation by MGG and IMC: (a)The consensus sequence of the contig is directly transferred from MGG to IMC. (b)Extraction of ORF candidates; c)ORFs are

changed to CDS with amino acids sequences.

2.3.3 Graphical User Interfaces

(1) Quality Viewer

After quality checking, every fragment is provided with the quality value which is described in **Chapter 2.2**. To overview the distribution of quality in a plate or in a library, MGG provides a graphical viewer as shown in **Figure 2.15**. The quality per fragment is overviewed in a plate style or list style appearance. Likewise, quality value per base is also visible in a bar style viewer as shown in **Figure 2.16**.

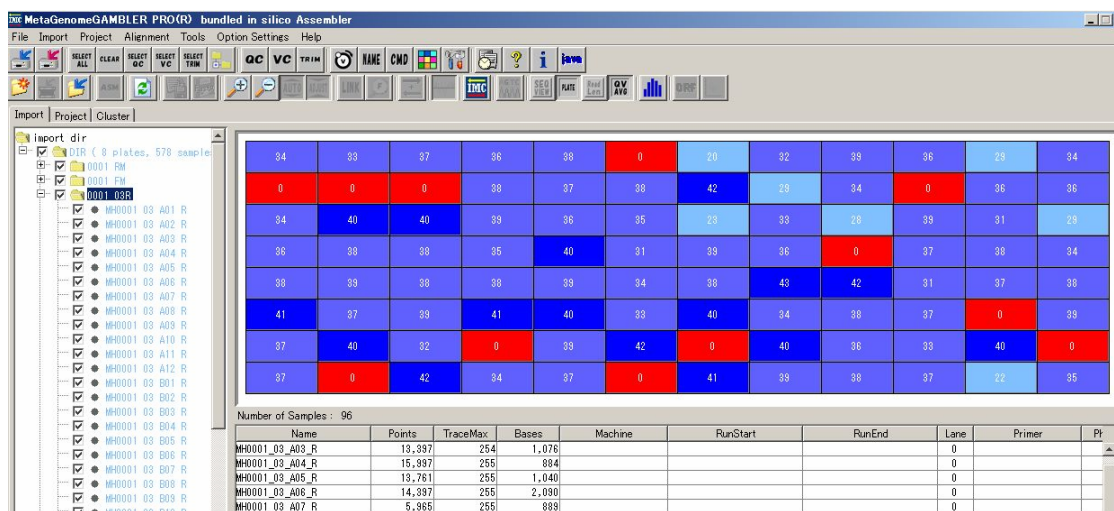


Figure 2.16 Plate style viewing in quality per fragment. Every well of a 96-well plate is classified by the colors designated by the grade of quality score.

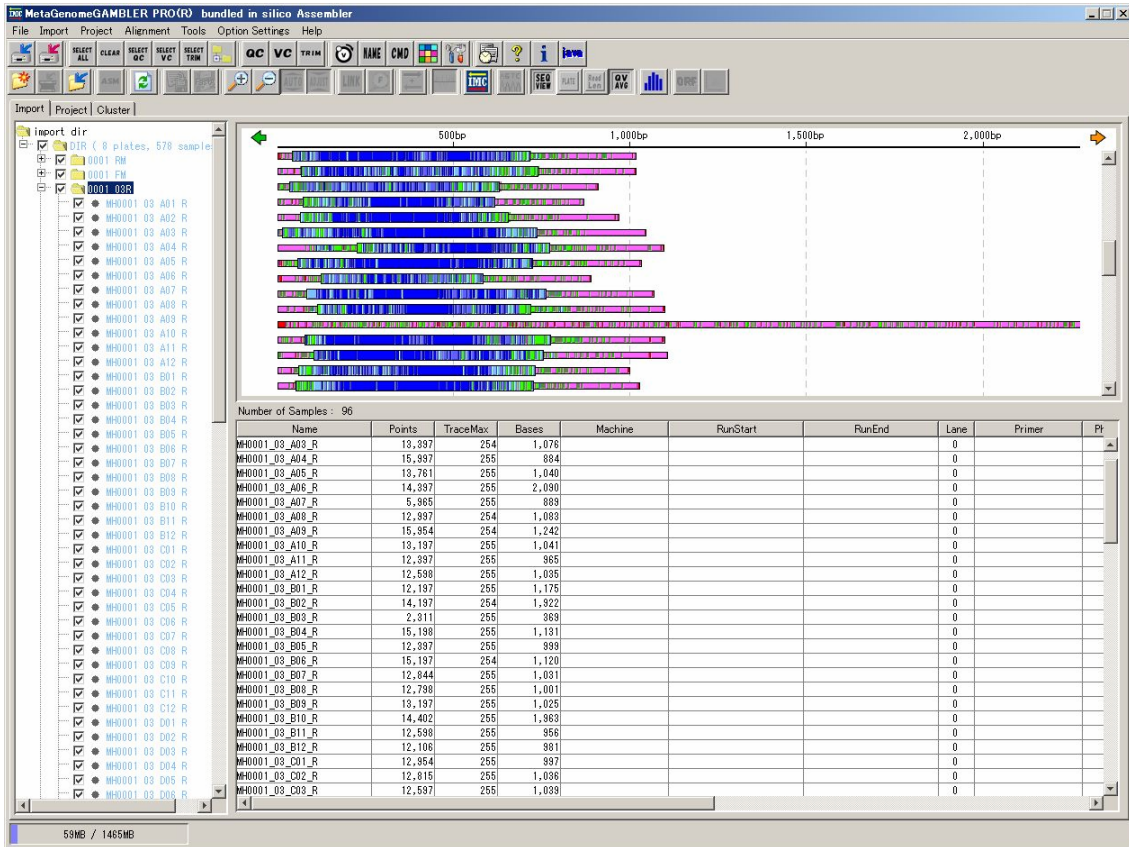


Figure 2.17 Viewing of quality per base in the bar style

(2) Trace Viewer

A trace viewer is a tool for monitoring the chromatograms of sequencing clone fragments (Figure 2.18). To draw these profiles, trace data which is included in the primary data reported from a DNA sequencer, is necessary. A trace data is created for each sequencing fragment. As for the 4 dye DNA sequencer, the trace data consists of four different series of fluorescent intensity which are derived from electrophoresis results. The four series of trace data are converted into spline curves of different colors which are assigned to 4 nucleotides of DNA.

The trace viewer is activated by clicking a fragment entry of lists in the quality window. Intensity and time interval can be zoomed.

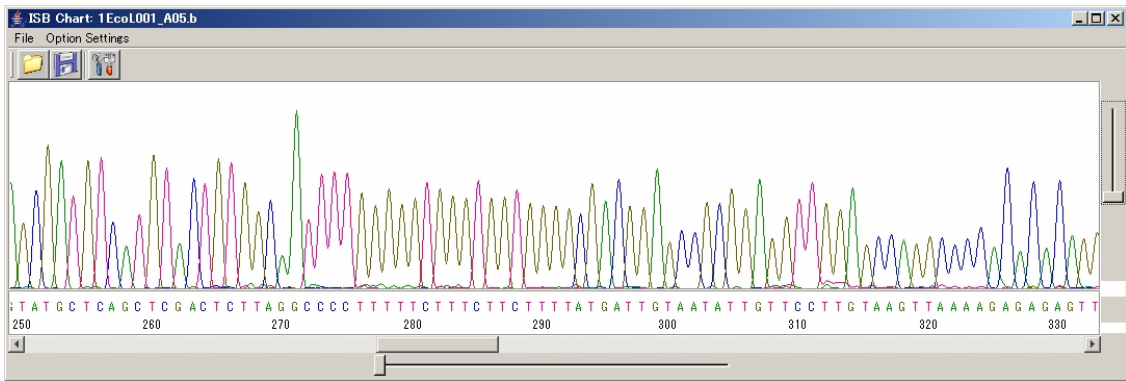


Figure 2.18 TraceViewer of MGG.

(3) Multiple Sequence and Fragment Alignment Viewer

Most of assemblers merely perform clustering and alignment of DNA fragments which are participated in one assemble trial, and produce a long list of contigs and their consensus sequences in a text format. Consequently, a graphical viewer for contigs and consensus sequences are much required. For this purpose, MGG is also providing such a graphical viewer named Contig Viewer. This viewer is incorporated in MGG and can be shown in a pane of MGG software as shown in **Figure 2.19**. Clicking of an assemble node which had at least finished assembling process, would activate the viewer.

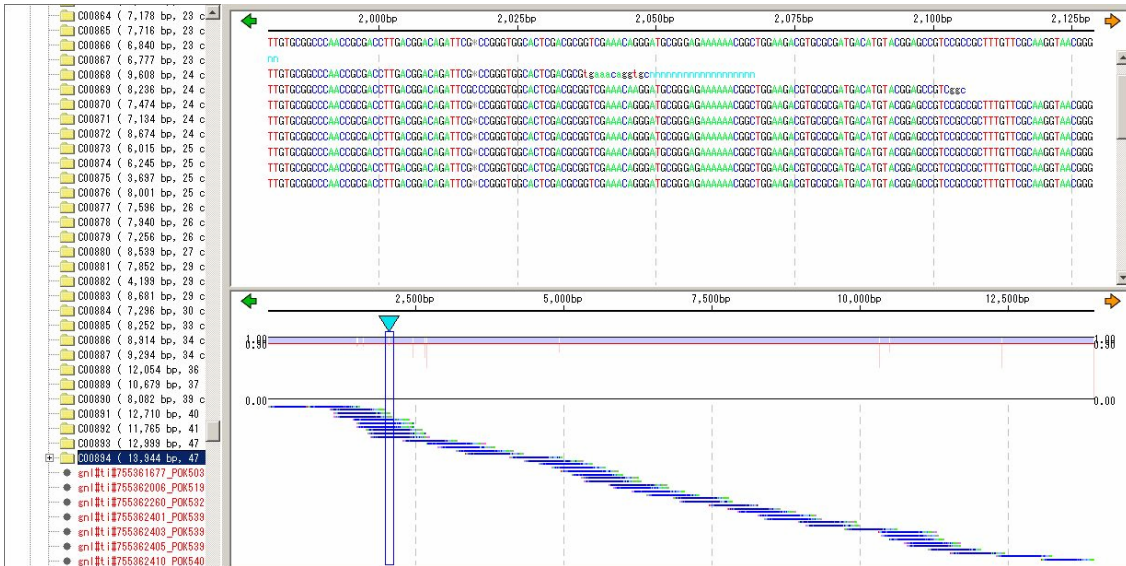


Figure 2.19 Contig viewer with consensus sequence (**Upper panel**), while more graphical drawing is shown in the **lower panel** which describes the quality profiles of the contig alignment with quality per base. The left pane shows the list of contigs and fragments with hierarchical. The consensus index profile is shown just over it.

In this viewer, on the upper pane the consensus sequence and aligned sequences are shown in different coloring methods which can be modified by changing parameters. The upper panel board is editable by base. As for the lower panel, the profile shows the at-a-glance indicator of low quality alignment. To detect the misassembled sites on a contig, a scoring method, named consensus index is used as shown in **Figure 2.20**. The consensus index indicates that how many bases are having consensus each other in a particular alignment position. As previously mentioned, misassembled regions appeared in a characteristic clustering shape as shown in **Figure 2.3**. The detection algorithm is based on the characteristics of misassembled clusters which are described in **Section 2.2.8**.

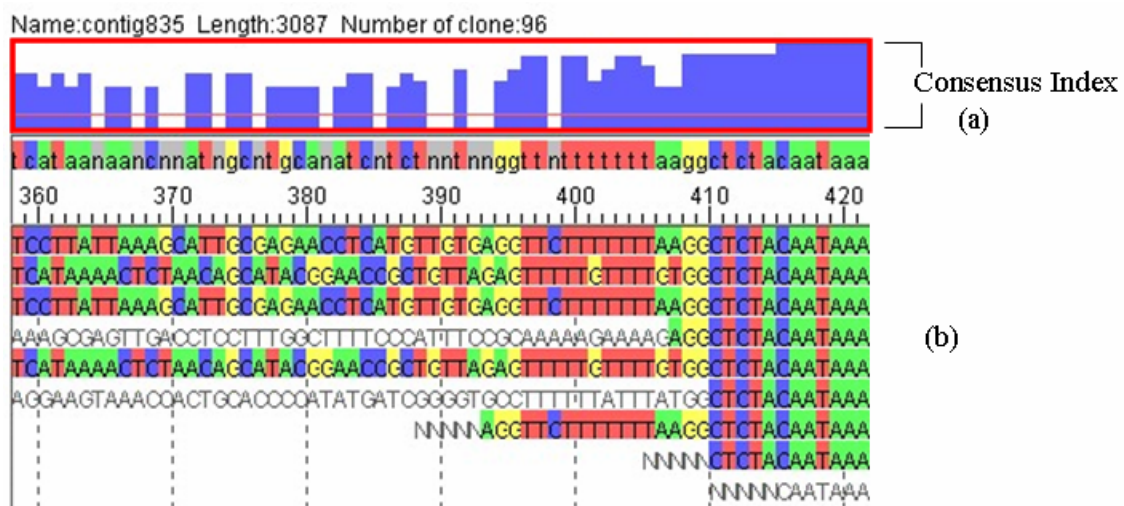


Figure 2.20 The consensus index and its viewer. (a) Consensus Index Profile is denoted by bars in blue color. (b) Multiple alignment of fragments is shown with four color presentation of each base. The alignment is in disorder at the low score region of the Consensus Index.

(4) Contig Linkage Viewer

As described in the **Chapter 2.2.9**, the mate information provides estimated distances between contigs when the sequencing clones are sequenced from the both ends. MGG provides the function as Contig Linkage Finder and Viewer (**Figure 2.21**). When a contig is shown on the Alignment Viewer, finding of mate pair partner can be performed. If done, every sequence in the contig is related to mate pairs in other contigs or the same one and the list of mate pair location is shown as a dialog. Just below the Alignment Viewer, new window called Contig Linkage Viewer is displayed. When specifying one fragment in the contig which has mate pair partner in the other contig, the graphical presentation of the positional relationship between the two contig is

shown.

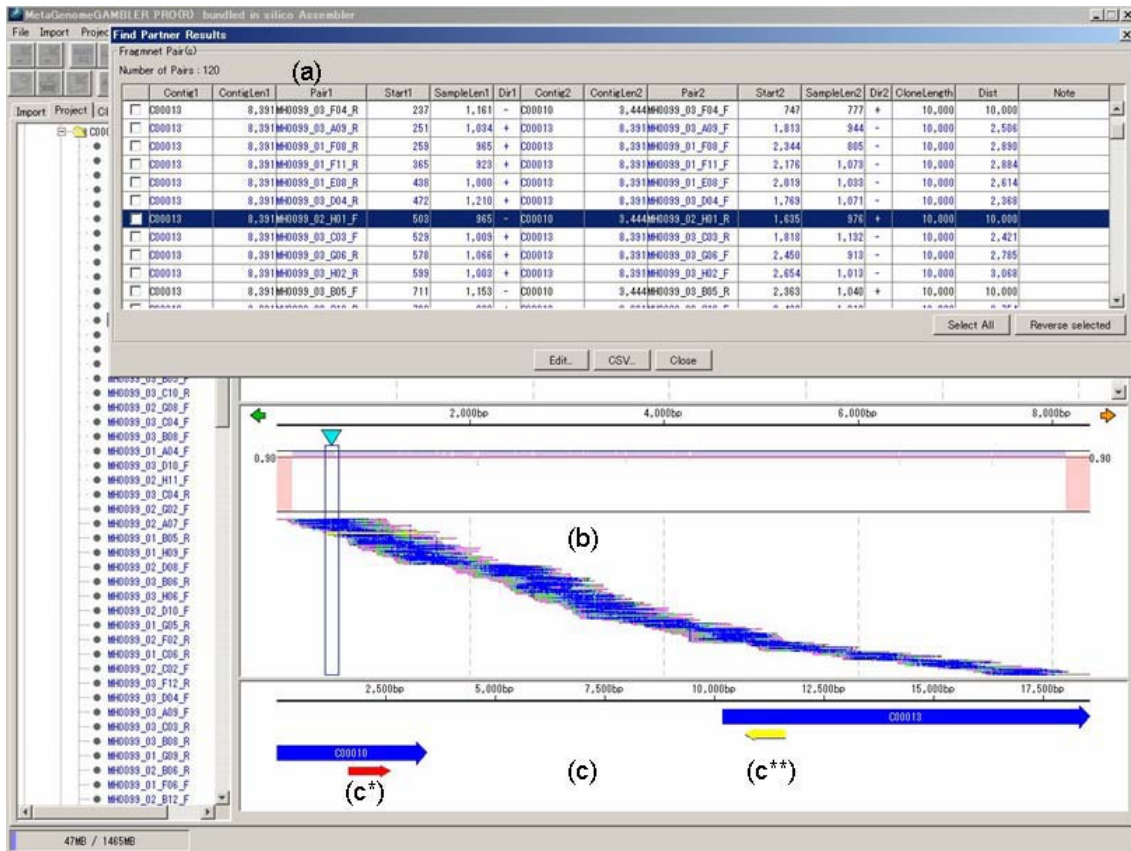


Figure 2.21 Contig LinkageViewer. (a) A window of the mate list of the contig is popped up after finding of mates are done. (b) Alignment Viewer is displaying the contig. (c) Contig Linkage Window is showing the relationship of the two contig which share one clone.

2.3.4 Sequence and Fragment Handler

Sequence and Fragment Handler is providing basic function of sequences and fragments handling. After imported, all the sequences are stored under the user-specified directory or folder and retrieved exact sequence when requested by viewers or managers of MGG. Information on the data location or other attributes, such as assembling parameters, results of assembling or quality data, is not saved anywhere else other than this directory. Therefore, this method realizes the multiple data handling with simple operation. Assembled data directory is transferred or exchanged like a cassette. Especially, the data is stored in a directory or folder of a portable disk drive, only inserting of the drive to the other PC or Mac is enough to use the data if MGG is installed on the machine.

2.4 Discussion

MGG is featured with a set of unique functions, such as automatic extraction of clone information from the DNA fragment files, handling of a huge set of fragment, searching of the clone pair partners and estimation of distances between contigs, and viewing various quality and their results viewers. MGG runs on Windows xp and Mac OS X and is developed by using Java language. Consequently, as for the software compatibility, it is advantageous to other application software written in other language such as C, C++ and perl.

(1) All-In-One handling of sequencing data in a small PC or MAC.

The most remarkable accomplishment by the development of MGG, is its all-in-one capability of all the required processes for microbial genome and cDNA sequencing. MGG acts as a front end software system and provides integrated interfaces to existing quality controller, DNA assembler and other programs, as well as its own unique tools such as detection of misassembling.

(2) Assembling of wide range of sequences

MGG can import various files with different formats or different naming rules together. This ability is advantageous when it comes to adding supplementary existing sequences which are registered in GenBank or EMBL. The user can assemble any part of genomes with only recruiting existing sequences which are stored everywhere in the world. This is new way of sequencing without cost. This type of functionality has not been implemented on the existing sequencing support systems.

(3) Supporting of sequencing projects without a server

Formerly, a large scale computing power is necessary when assembling. MGG provides rather small and personal research environment of compact assembling. Namely, all the processes of MGG run on one small PC and never rely on other computing facilities. Although being compact, MGG is still capable of assembling more than 100,000 fragments at once. Most of prokaryotic genome projects are within a reach of such MGG capability. No reliance on computer experts would create wider possibility of research on genomes.

(4) Assembler independency

MGG is assembler independent software, namely, it has no inner functionality of assembling, instead of this, MGG can activate stand-alone assemblers with parameters transferred. Historically, the previous version of GenomeGAMBLER (Sakiyama et al. 2000, Takami et al. 2000) had been developed as a web user interface to PHRAP assembler (Green 1999). Therefore this is inherited to MGG, too.

(5) Searching of the mate pair for assisting the primer walking

Random shotgun method is superior to primer walking due to its simplicity. Almost every process could be automated in a random shotgun phase. However, when it comes to the phase of gap closing, accidental closing of the gaps are becoming less and less. Consequently, primer walking as a more steady method would be adopted even if there is some complexity preparing suitable primer sets in a limited cost. Therefore, to simplify the primer walking phase with inferring optimal primer sets is requested. MGG provides this capability in it.

(6) Graphical quality viewer per base or per fragment

Even if more fragments are provided to be assembled, fragments with low quality regions or with contamination of vectors and other xenogenic sequences may become obstacles to obtain good results. In this case, prior processing of quality control is necessary. To perform this, a fully automated process may be the best, however it is not yet accomplished. In the next best solution, a graphical viewer of quality would help it a lot.

(7) Furnished with many outputs for reporting the status and results of assemble

Reviewing of previous sequencing project is much important to obtain higher performance in the following projects. For evaluation of the previous projects, variety of statistical and graphical presentation is required. MGG provides wide range of outputs with graphical presentation.

Chapter 3

Genomic DNA sequence analysis and its front end environment

3.1 Introduction

A front-end molecular biology software system which is described in **Chapter 1**, may be a mandatory software for molecular biologists. I designed and developed two of the software tools which run on the ordinary note PC and Macintosh environments. The software systems are actually already being used by molecular biologists, medical scientists and students. Producing such software tools may lead to more rapid acceleration of research in the related fields. In this chapter, major functions of the software system, named *in silico Molecular Cloning* (IMC), are described.

Most of sequence analysis software tools first appeared in 1980s. They are roughly classified into two types. One type is integrated user interfaced software packages which incorporate a variety of then existing single function tools, the other are single functioned tools themselves. Typical tools of the former are GCG (Genetic Computer Group) Wisconsin Package (Gribskov & Devreaux 1990) which had been developed in University of Wisconsin since early 1980s, and Staden Package (Staden 1996) which was developed by Roger Staden in the same periods. Later, a biotech company named Intelligenetics, developed GeneWorks which was popularly used by experimental molecular biologists. However, since a large size data of genomic sequences are publicly available around year of 2000, these software packages could not follow the rapidly increasing sequence data.

As for the single functioned sequence analysis tools, BLAST (Altschul et al. 1990) is the first to be nominated. The software is widely used among the molecular biologists. BLAST provides a single function of homology search accompanied with pairwise alignment capability. Due to its fast and flexible processing of nucleotide and amino acids sequences, BLAST also has been used as basic component of integrated software tools. The second to be noted is a program named ClustalW which provides multiple alignment algorithm to align many homologous sequences and obtains a consensus sequence. It is also used to obtain a phylogenetic tree. As for 3D structure viewer of proteins or nucleotides, RasMol (Sayle & Milner-White 1995) or Cn3D (Wang et al. 2000) have popularity. After large scale sequencing projects were launched in 1990s, numerous gene finding software tools were developed. Among them, Glimmer (Salzberg et al. 1998, Delcher et al. 1999) and GeneMark (Borodovsky et al. 1993)) are commonly used to find out ORFs in newly-sequenced microbial genome sequences. To find out the ORFS in eukaryotic genomes, GenScan (Burge 1997), MZEF (Zhang 1997), Morgan (Salzberg et al. 1998) and many other software tools were developed. In such

gene finding software tools, further basic algorithms such as codon preference (Staden & McLachlan 1982), TestCode (Fickett 1982) or simply calculation of nucleotides or amino acids composition, are more or less incorporated. Thus, one software can not be developed independently, it must depend on numerous previous tools. A number of the single function software tools are publicly available via web-sites which provide the prediction results to the internet users. However, they usually offer different procedures to a query, the users must customize their queries for each web-sites. Thus, it is not easy to analyze one biological sequence with different algorithms provided by these sites.

Molecular biologists were confronted with lack of suitable integrated front end software tools for the genome era and also confused with the too many selections of web-sites for analysis of their data.

In silico MolecularCloning (IMC) was designed to handle a chromosome level sequence data up to the size of the human chromosome number one which has about 25 mega bases. IMC also has a loose connection with single function software tools. For examples, IMC can activate those programs of BLAST, ClustalW (Higgins and Sharp 1988, Thompson et al. 1994), RasMol and Cn3D, and import the results from Glimmer, tRNAscanSE (Lowe and Eddy 1997).

IMC also provides many editing and viewing functions for the users to customize their biological sequences. These implementations are performed using only on the GenBank/EMBL format files.

3.2 Methods and Algorithms

3.2.1 Data handling

(1) Internal format data

Although IMC uses GenBank (Benson 2006) and EMBL (Cochrane et al. 2006) format for its basic file format, internal handling of the data is quite different. The structure of the internal data is optimally designed to fast access. After imported, sequences and features are converted from GenBank/EMBL format to the internal format of IMC. During editing, viewing or analyzing biological sequences, these internal processes are performed only against the internal format data which are mostly on memory to accelerate the processing speed. These internal format data are never exported as they are. Instead, the internal format data are immediately converted and exported as GenBank/EMBL format files if requested. The structure of the internal format is described in **Figure 3.1**.

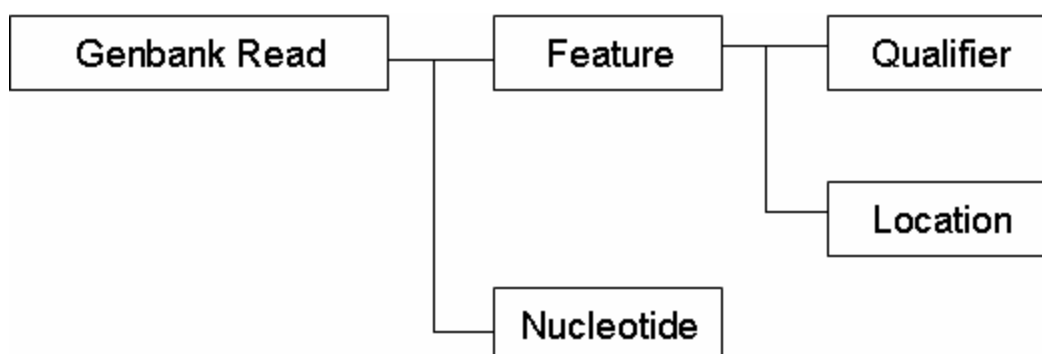


Figure 3.1 Conceptual data structure of IMC internal format. Genbank format file is roughly divided into two sections, one for features and the other for nucleotide sequence. Features are classified with feature key, namely, features with same feature key are assigned in the same list. In the individual list of a feature key, every feature has its qualifiers and location in the DNA.

(2) Compressed data

IMC reads compressed sequence files and entries directly and uncompress

them inside the software. IMC uses the Java library of un-compression.

(3) Automated structuring of taxonomy tree arrangement of sequence entries

In IMC, entries imported can be automatically renamed according to taxonomy description in the source line or individual entry names, and transferred to exact position of the taxonomy tree. For example, in the GenBank format file of *Bacillus subtilis subtilis* 168 genome, the source line is described as followings.

```
SOURCE      Bacillus subtilis subsp. subtilis str. 168
ORGANISM    Bacillus subtilis subsp. subtilis str. 168
            Bacteria; Firmicutes; Bacillales; Bacillaceae; Bacillus.
```

The last line indicates lineage of the species. IMC pursues this description and re-construct it into a rooted tree structure. When a huge number of entries are imported, this structure grows into a large tree of the taxonomy. However, when referring a species or strain on the tree, its routing is easy to be located than the other methods.

(4) Data link to other software tools

IMC imports and exports simple text format, FastA format or CSV format files, as well as GenBank/EMBL format files. The CSV format file is written in an original format and used to exchange features with other software tools. In the CSV format file, the positions of features are written according to the convention of the international nucleotide databases of DDBJ/EMBL/GenBank. A CSV format file with features attached with their nucleotide sequences are used for inheriting the old annotation to new version of genomic sequences. This is simply performed by using homology between genomic sequences and the nucleotide sequences attached with features. The annotation on features is transferred to the new version genomic sequences if the homology is detected (see also 3.2.3).

(4) Feature manipulation

Features are defined as the stretch of sequence which has certain biological or physical functionality. Features are denoted on GenBank/EMBL format file as a set of

lines which describe the functionalities. Every feature is accompanied by several qualifiers which provide the feature values or characteristics.

IMC creates new feature keys with new qualifiers, as well as editing, inserting or deleting existing features with qualifiers. This is performed on the tree structured internal format memory space. The direct editing of GenBank/EMBL format data is not performed. The tree size and depth are optimized to accomplish tolerate speeds of feature operation. In IMC, some original feature keys and qualifiers are initially assigned. These originally incorporated features and qualifiers can be filtered out when exporting as a GenBank/EMBL format file.

As for the original feature keys, IMC uses the ‘Tiling_Array’ and ‘Tiling_Info’ feature key for microarray applications (see **Chapter 5**) and ‘blast’ feature key for storing the information about the locations of the blast databases, as well as the ‘endtype’ feature key in *in silico* experiment functions (see **Chapter 4**).

As for the original qualifiers, a wide usage of them is implemented. When a feature is modified or inserted, a time-stamp qualifier ‘/update=*datetime*’ is added to the modified feature. A feature designated as to be deleted is given with a qualifier ‘/delete=*datetime*’. A ‘/color=*code*’ qualifier is given to the feature which should be painted in the specified color code in the *code* value. Text files of referenced papers are also linked in a similar way by using a ‘/journal=*file location*’ qualifier. The three dimensional structure data is also linked using a ‘/3D_Structure=*file location*’ qualifier. A ‘/classification=*code*’ qualifier is used for classifying the CDSs with different functions. A ‘/annotation_grade=*grade*’ qualifier is used for identifying the CDSs on different grades of annotation.

3.2.2 Sequence Analysis

(1) Nucleotide compositional profiling

GC content and AT content are complementary number each other and used to locate the candidate regions of genes or promoters or other biological functions on unknown genome sequences. These numbers are simply obtained by dividing the sum of guanines and cytosines with total number of bases (GC content) in a window. In the same manner, A, C, G and T contents are obtained by dividing corresponding number of bases by the total bases in a window.

GC skew and AT skew is introduced to determine replication origin and terminus in genomes (Lobry 1996). GC skew can be calculated in the case that a

genome sequence is determined completely, because it is defined as the difference between G and C composition in one DNA strand in a genome. The detection of GC skew in a global genome is provided with a window approach. In IMC, GC skew for i th nucleotide in the plus strand of a genome was represented by **Eq.3.1**.

$$\text{GC skew}_{w,i} = (G_{w,i} - C_{w,i}) / (G_{w,i} + C_{w,i}), \quad (3.1)$$

where, w is the window size, $G_{w,i}$ and $C_{w,i}$ are the numbers of guanine and cytosine nucleotides, respectively, with the i th nucleotide as the center of an window size with w nucleotides. AT skew is also obtained likewise (**Eq.3.2**).

$$\text{AT skew}_{w,i} = (A_{w,i} - T_{w,i}) / (A_{w,i} + T_{w,i}), \quad (3.2)$$

where, w is the window size, $A_{w,i}$ and $T_{w,i}$ are the numbers of adenine and thymine nucleotides, respectively, with the i th nucleotide as the center of an window size with w nucleotides.

Cumulative GC skew is more sensitive for the detection of replication origin and terminus in genomes (Grigoriev 1998, 1999). The cumulative GC skew is calculated as a sum of $(G - C) / (G + C)$ in adjacent windows, from arbitrary start to a given point in a sequence. The cumulative GC skew value reaches its global maximum at the terminus, while the minimum resides over the replication origin.

(2) Codon usage

Codon usage is defined as the frequencies of codons within the CDS regions which are translated to amino acids. There are 64 possible codons including the stop codons and start codons. Therefore, codon usage is tabulated in 64 cells. The stop codon appeared in a CDS only once. Codons which are translated into a same amino acid, are not equally used (Nakamura 2000, Ikemura & Wada1991). Especially, several codons are rarely used. In IMC, the codon usage table is calculated for a single CDS or a group of CDSs and all the CDSs on a genome.

(3) Finding ORF candidates

As the first step of ORF finding in a given DNA sequence, all the stop codons are searched in six frames of the sequence. After the search, each frame can be divided into many subsequences separated by the stop codons. Hereafter, the subsequence is

called as an ORF box. An ORF box is a basic unit used for ORF finding. Then, ORF candidate is define as nucleotide sequence started by the start codon which is first encountered after scanning from the first codon of the ORF box. The start codons can be selected among ‘ATG’, ‘CTG’, ‘GTG’ and/or ‘TTG’ codons.

(4) Translation

A ribosome complex translates mRNA sequence into amino acids sequence. When translating, a different genetic code table should be used for the different organization, such as mitochondria, chloroplast or virus. Therefore, appropriate genetic tables must be given in translation. When no start codon is selected, the first codon of the candidate ORF is designated as a tentative start codon. In the prokaryote genome, nevertheless the start codon has its translated amino acids code, the first amino acids are translated as methionine (formylmethionine) except no start codon is selected.

3.2.3 Feature Handling

(1) Editing of features and affected annotation

Features are modified in its position on the sequence and in the values or characteristics. In a circular DNA sequence, feature assignments across the last nucleotide and first one are possible. Features position is affected by the reverse complementary operation of DNA. Features are also affected by nucleotide modification. When the nucleotide sequence on which features are allocated, is modified, namely, deleted, replaced or inserted, features allocated on the downstream of the modified nucleotides are also shifted.

(2) Feature handling with cloning functions

When performing *in silico* cloning experiments, further consideration is necessary for feature handling. When digested by a restriction enzyme, DNA sequence with features is correctly divided at the exact site of RE digestion. If there is a feature which is located just across the digested site, the feature is also divided into two parts. If divided, the feature might loose the functions or characteristics. If so, the feature key of the feature must be changed. For example, if a CDS is digested in the midst of the sequence, the upstream half of the CDS may lose its stop codon while the downstream half of the CDS may lose its start codon. One of these digested fragments could be ligated into a vector then if it includes the upstream half of the CDS the inserted would

have a new stop codon. Thus, the resulted CDS would become longer or shorter in size. Then the annotation must be modified according to the changes. The restriction site itself may also be changed its recognition pattern in case of ligation between two ends cut by different restriction enzymes.

Further description about feature handling in cloning is given in **Chapter 4** and **Chapter 5**.

(3) Feature handling in a circular DNA

Genomic DNA takes one of the shapes of linear or circular. When handling a circular DNA, several considerations must be taken. In a DNA, a reading frame is defined as a contiguous and non-overlapping set of three-nucleotide codons. There are six possible reading frames in a DNA molecule. In addition, a peculiar phenomenon is observed when a DNA has nucleotides whose number is the multiple of three plus one or two. When making a circuit of the whole DNA, reading frame is shifted to another across the starting point of DNA. This type of reading frame shift exist only small circular genome of virus or plasmid. Actually, any viruses have overlapping genes in different reading frame.

In IMC, this kind of frame shift is expressed by the following methods. If there is a open reading frame (ORF) which goes around the DNA and enters into second round, this ORF is expressed as the join of two intervals, one is from (1.. N) and the other is (1.. m), where N is the largest nucleotide number in the DNA and the m is the last position of the ORF.

(4) Importing and exporting of features

Sometimes separate and independent handling of nucleotide sequences and features is necessary. Namely, features should be extracted from the annotated sequence and pasted again on the same nucleotide sequence. This is a simple case. However, if there is modification to the nucleotide sequence after feature extraction from the sequence, re-pasting is not so simple. In such a case, every feature should be extracted with its nucleotide sequence. When importing onto the modified nucleotide sequence, homology searches by the attached sequences assigned to corresponding features are performed. By this method, most of the features would be pasted onto the newly updated sequence.

When ligation performed between two different DNAs, ‘source’ feature performs an important role. To identify the two DNA fragment, IMC generate ‘source’

features if there are no 'source' feature. Compared with rather short spans of other features, a 'source' feature usually spans the entire sequence. A chimeric or ligated DNA is used to have more than two 'source' features. For example, IMC uses the 'source' features to identify the inserted DNA into a vector sequence to draw a plasmid map with protruding insert sequence from the circular plasmid.

(5) Laboratory notebook functions

Experiments *in vitro*, requires a strict record about the procedures of the experiments. A notebook made of paper is usually used for this purpose. All the procedures using IMC, are recorded sequentially and can be retrieved at random. In IMC, all the modifications on features are recorded with exact time stamp qualifiers. Since additional and overlain modification can be described with multiple entries of '/update=*time-stamp*' qualifiers, a history of modification will be also reproduced.

(6) Reverse complementary function

Reverse complementary function against DNA sequence is important when performing ligation *in silico*. If the both ends of two fragments are of same shape, two types of ligation are possible. If one fragment is inserted reversely against the other fragment, the sequence of one of the fragments must be reversely complemented. In addition, all the features on the sequence must also be reversely annotated. In IMC, this reverse complementary operation is performed on the tree structured internal data on memory. All the positions of features are re-calculated at once on memory. However, the order of features is not affected this time. When exporting the data, features are re-sorted by the location on the DNA sequence.

3.2.4 Homology search and feature alignment

In IMC, homology searches are performed in integrated and automated manner. A click on a feature immediately creates a query nucleotide or amino acid sequence derived from the feature, and activates homology search programs, such as BLAST, against temporarily created BLAST databases. On the other hand, its corresponding BLAST databases for nucleotide sequence and amino acids sequences are automatically created when a DNA sequence with translated amino acids is imported to IMC. The created BLAST databases are stored in a specified location on PC. The homology results are also stored automatically as qualifiers (*/blast=results*). These results can be transferable with GenBank/EMBL format files.

(1) Database creation

In IMC, BLAST databases are created in two different ways, automatic creation and manual creation. When a DNA file with CDS or mRNA features is imported in the feature map or in the reference map, a BLAST database including all the CDS or mRNA is automatically created. When removed from the maps, these databases are automatically removed. Thus, these automatically created databases exist temporarily. On the other hand, a BLAST database is manually created for permanent use. Nucleotide or amino acid sequence files with FastA format, GenPept format can be currently handled.

(2) Feature alignment

Features can be aligned according to the homology of nucleotide or amino acid sequence between each other. Since features are assigned on a sequence and given the absolute positions on the sequence, their nucleotide sequences are easily derived. As for the amino acid sequence of the features of CDS or mRNA must be translated using an appropriate genetic code table. In eukaryotic genomes, CDSs usually contain introns inside a gene. A feature alignment between two intron-containing genes about amino acids sequences is performed as if separate exons are aligned in nucleotide sequences. Namely, the corresponding DNA regions against aligned amino acids sequences are aligned between the feature map and the reference map.

(Discussion) Let all the features on the feature map and the reference map be automatically registered in a BLAST database for later homology searching just when they are imported, this method provides implicit databases without specifying the names. By this method, selection of query, specification of databases and viewing of the results are easily operated. A query feature on the feature map or the reference map is fixed on the map and the homologous features are aligned one by one after specifying the entry of the homology result list.

(3) Multiple alignments

A multiple alignment is one of the hard problems among the sequence analysis. Finding of the best alignment requires a lot of computing time and memory. To avoid this, greedy algorithms are developed applying heuristics to compute it within a limited computing time and memory. Although, these algorithms do not always provide the optimal alignment, they are practically useful. These algorithms are classified into four families (Duret & Abdeddaim 2000). (1) Algorithms that guarantee to find the optimal

alignment for a given scoring scheme. (2) Progressive pairwise alignment approach. (3) Global alignment based on local alignment. (4) Local multiple alignment. Among the four, practically the progressive alignment algorithms are commonly used because of their computing speed and reduced memory space. ClustalW (Thompson et al. 1994) is the one of the programs which uses the progressive alignment algorithm. As for the first step of the algorithm, ClustalW provides selection between dynamic programming and heuristic algorithm. Dynamic programming gives more accurate but slower than heuristic methods. The early version of Clustal (Higgins and Sharp 1988) used the UPGMA algorithm (Sneath & Sokal 1973). However, UPGMA is sometimes giving incorrect branching orders. Therefore, ClustalW currently uses the Neighbor-Joining (Saito & Nei 1987) algorithm to create the dendrogram. In IMC, the results from BLAST search are provided to ClustalW which is also automatically activated and after manual selection of sequences to be aligned, the results are obtained.

(4) Phylogenetic tree

A phylogenetic tree is defined as a tree-structure presentation of the evolutionary interrelationship among the various species or other entities that are believed to have a common ancestor. In a phylogenetic tree, each node with descendants represents the most recent common ancestor of the descendants, and edge lengths correspond to evolution time estimated. A rooted phylogenetic tree is a directed tree with a unique node corresponding to the most recent common ancestor of all the entities (Feng and Doolittle 1990, Felsenstein 1981, 1988, 1997, Swofford and Olsen 1990). The most common method for rooted trees is the use of outgroup. Unrooted phylogenetic tree can be generated from rooted trees by removing the root. Optimally arranged drawing of un-rooted trees is not easy problem. IMC uses a modified algorithm of Feng and Doolittle.

3.2.5 Viewing algorithms

(1) Feature Map

Most of annotation software tools are provided with feature map drawing facilities such as Artemis (Rutherford et al. 2000). A feature map is usually equipped with feature presentation in multiple figures and multiple colors along the entire stretch of long chromosomal nucleotide sequence. Selections of profiles, such as GC content or GC skew plots are always accompanied. Since a chromosome has wide dynamic range of features, zooming and locating functions are necessary for these viewers. IMC uses a frame request algorithm. Namely, the map drawing module of IMC, requests only

enough and adequate information to draw the map at one time, while the memory handler module prepares and hands the data immediately. When the map displays comparatively small region of the genome, features contained in the region are not many. Therefore, this computing is easily performed. However, the larger becomes the region to be displayed on the map, the burden on the computing increases rapidly. IMC is avoiding this rapidly increase load on computing using a thinning-out algorithm. This algorithm skips or thins out some portion of features from drawing when the map contains a lot of features and can not be identified in vision.

(2) Circular Genome Map

Since the first two microbial sequences are completed, a circular genome map presentation became popular among the authors of sequencing project papers (Fraser et al. 1995, Fleischman et al. 1995). The drawing software tools of the circular genome map are continuously being developed, for examples such as Microbial Genome Viewer (Kerhoven 2004), GenomePlot (Gibson and Smith 2003) and GenoMap (Sato and Ehira 2003). Most of the drawing software tools lack of integrated handling tools of genome sequences, therefore it is not easy to draw these circular map instantly. IMC draws the circular map from the annotated genome sequence of GenBank/EMBL format.

(3) Plasmid Map

Drawing of a plasmid map is one of the most requested functions in the sequence analysis software tools. Many software tools to draw a plasmid map were developed such as PlasMapper (Dong 2004). Commercial software packages such as GCG Wisconsin Package (Dolz 1994, Butler 1998)) also provide this kind of functions. However, most of the plasmid map drawing software tools lack of automatic identification methods of chimeric DNA, and must be manually specified. Therefore with them, it is not easy to draw a plasmid map with insert DNA protruding from the circle. IMC utilizes the 'source' feature for the automatic identification and draw the protruding plasmid map easily.

3.2.5 Other search algorithms

(1) Keyword search

A keyword search in IMC means that query keywords, such as 'dehydrogenase' or 'human', are searched against the features of GenBank/EMBL format files. Thus, the answered entities are the features themselves. Namely, features with qualifiers including the query words are detected. Keyword searches in IMC are performed using string

search function of Java language. Logical operations between keywords are limited but allowed a combination of 'AND' and 'OR'.

(2) Feature key search

Feature key search is defined as a searching method of feature key. In IMC, features with a same feature key are stored in a list structure so as to achieve the shorter computing time to add a new feature, modify a feature entry or delete an existing feature. This method is appropriate for the immediate listing of numbers of all the feature keys. After selecting one or more feature keys, all the lists of each feature locations are returned.

(3) Sequence pattern search

Search by short and conserved nucleotide or amino acid sequences is defined as sequence pattern search. This search is performed with exact match detection with the query pattern. However, ambiguous pattern searches with ambiguous nucleotide or amino acid code are also performed.

In IMC, pattern search is using regular expression. A regular expression is a string that describes or matches a set of strings, according to certain syntax rules. Programming languages, such as Java, support regular expressions for string manipulation. As for one nucleotide or amino acids mismatch searching, IMC searches target DNA sequence or amino acids sequence with changing the query sequence patterns one character by one from leftmost to the rightmost of the pattern sequences.

(4) Priming sites search

When DNA amplification by PCR is performed in IMC, several basic functions are processed such as priming sites search, duplication of DNA regions between two of the priming sites and attaches of primer sequences to the both ends of the amplified products. A priming sites search in IMC is using string search method of Java language. Exact matches are required as detecting of priming sites. However, one or two mismatches are also allowed when specified.

3.2.6 Amino acid sequence analysis algorithms

(1) Profiling

When describing characteristics of an amino acid sequence, a window approach for amino acid profiling is often used especially for its secondary structure

predictions. Since a protein has multiple characteristics or functions, it is useful to show several different profiles in parallel. Currently available profiles are prediction of alpha helix, beta sheet and beta turn regions (Chou & Fasman 1978), average chain flexibility (Bhaskaran & Ponnuswamy 1988), relative mutability (Dayhoff et al. 1978), hydrophilicity for prediction of antigenic regions (Hopp & Woods 1981), prediction of accessible residues and buried residues (Janin 1979), prediction of chain flexibility (Karplus & Schulz 1985), hydrophobicity for prediction of trans-membrane regions (Kyte & Doolittle 1982), polarity (Zimmerman et al. 1968) and bulkiness (Zimmerman et al. 1968).

(2) Linear combination of weighted profile

I introduced a linear and weighted combination of the above profiles to predict a specific aspect of the amino acid sequence. Namely, the total index of j th window with size s amino acids, $T_{s,j}$, is defined as the following equation (eq. 3.3).

$$T_{s,j} = \sum w_i P_{i,(s,j)}. \quad (3.3)$$

where w_i is weight of i th profile and P_i is denoted as the value of i th profile and s is the window size indicated number of nucleotide in a sliding window, with the j th amino acid as the center of the window size. By changing the weight w_i , a new linear combination profile can be obtained. Optimal weights can be obtained by solving the linear programming of simultaneous equations derived from experiments.

3.2.6 Genome comparison

(1) Dot plot between genomes

When comparing two closely-related DNAs, dot plots are powerful means of sequence analysis, useful for searching out regions of similarity in two sequences. However, when it comes to compare two whole genomes, computing time required to obtain the result is very long. Therefore, several so-called greedy algorithms are developed. In IMC, megaBLAST (Zhang et al. 2000) is used to obtain adequate results within affordable time. A graphical presentation of dot matrix is usually used. The megaBLAST provides segment pairs from the two genomes. Then, the segment pairs are plotted on a dot matrix if the pair segment size is beyond the threshold size for plotting (Gibbs and McIntyre 1970, Devereux J. et al. 1984, Sonnhammer and Durbin 1995, Junier and Pagni 2000).

(2) Finding of repetitive sequences in a genome

Finding of repetitive sequences in a genome is a special case of the above-mentioned comparison between two genomes. There are a lot of software tools to find out repetitive sequences from DNA sequences such as FORRepeats (Lefebvre et al. 2003, Lavorgna et al. 2005), MUMmer (Delcher et al. 1999), REPuter (Kurtz et al. 2000) and RAP (Campagna et al. 2005). If the two genomes are identical, this means that similarity regions are denoted as candidates for its repetitive sequences. According to the threshold given to determine if repetitive or not, repetitive regions in a genome is indicated. IMC uses megaBLAST (Zhang et al. 2000) for its searching engine.

(3) Finding of unique regions between genomes

Finding of unique regions between different genome sequences, is also the special case of the two genome comparison method. In IMC, at first, pairwise comparison between any two genomes is performed. Afterwards, combination of the pairwise comparison can be obtained.

3.2.7 Other algorithms

(1) Importing and pasting of cDNA sequences on genomes

If the genome sequence is known, cDNA transcribed from the genome, is expected to be pasted on, namely cDNA nucleotide sequences have the homology in exon regions with its genomic sequence. By using this phenomenon, most of cDNA can be allocated and assigned as mRNA features on the genome. In addition, eukaryotic genomes usually divided into many chromosomes. Therefore, when pasting, homology search against all the chromosomes of the genome are examined. If one of the cDNA has homology against multiple chromosomes, it means that they homologous regions are coded as paralogues each other. If one of cDNA has partial homology against the genome, it means that the cDNA may be derived from the truncated mRNA.

(2) Primer design

In IMC, primers optimized for amplifying the specified region, can be obtained. The optimal primer set is obtained as described in the following method. All the sub sequences limited by the primer design parameters are examined. Any violation of the criteria means the drop from the primer candidates. To differentiate the results, two scoring systems are introduced. The product Tm value is given the first priority factor in Score T while the product length is given the first priority in Score P .

In the last part of the algorithm, homology search with primer sequences

against the whole DNA is performed to detect other possible priming sites in the genome.

3.3 Results: Implementation of *in silico* MolecularCloning (IMC)

Covering wide range of functionality may be the major role of a front end software environment. The major functions implemented in IMC are described in the **Figure 3.2**.

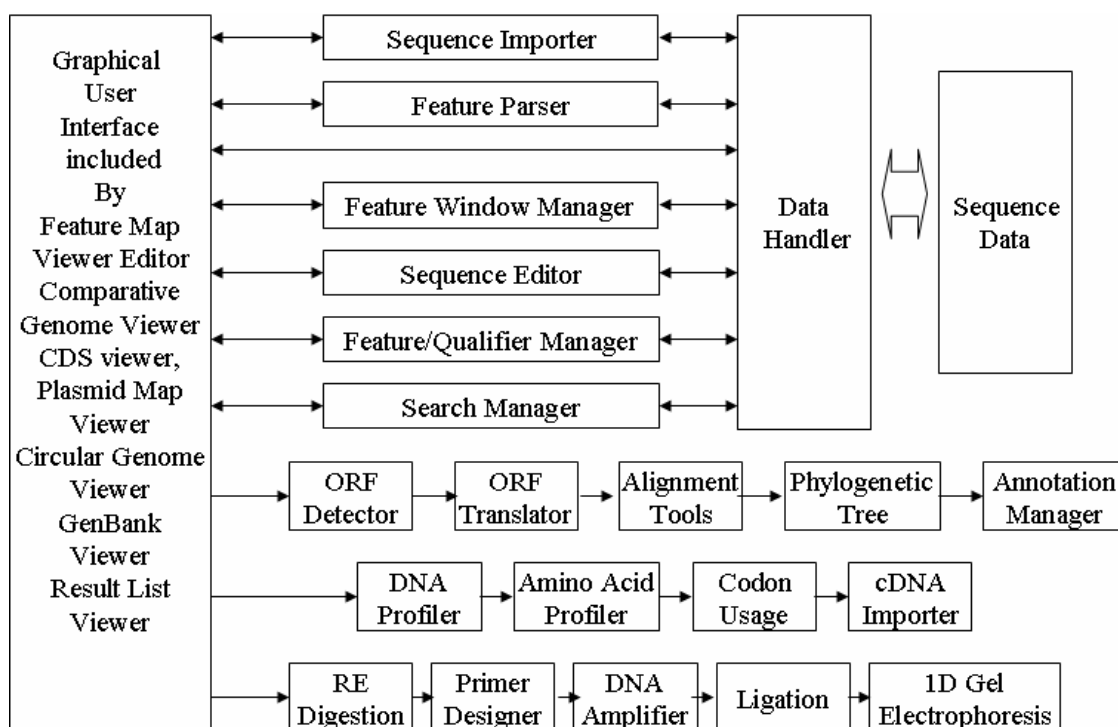


Figure 3.2 Package design diagram of *in silico* MolecularCloning. The structure of the software package is shown. The user operates IMC via Graphical User Interface (GUI), such as Feature Map Viewer Editor, Comparative Genome Viewer, CDS viewer, Plasmid Map Viewer, Circular Genome Viewer, Genbank/EMBL file Viewer and Result List Viewer. Sequence data is indispensably handled with Data Handler. Between the GUIs and Data Handler, several data managers, such as Sequence Importer, Feature Parser, Sequence Editor, Feature/Qualifier Manager and Search Manager are processing to and fro. Applications are also controlled by GUI.

3.3.1 Handling of sequence data

IMC imports annotated nucleotide sequences, and creates a feature map for any region in any scale. IMC reads and writes both GenBank and EMBL format annotated nucleotide sequence files. Format conversion between the two formats is done in an

automatic manner. IMC also supports plain nucleic acid sequences like FastA format file or plain text sequence file. IMC can read several different annotated sequence files concurrently, and switch views from one annotated feature map to another instantly. These files are editable with many editing functions.

(1) FastA format files are imported and can be annotated freely

Un-annotated nucleotides sequence files can be imported into IMC, a blank feature map is shown at first, then feature assignment works could be done on it as its user prefers.

(2) Compression files are directly imported

Since large genomes were sequenced, the international databases tend to provide sequence data with annotation in a compressed format to reduce the file size. When using the compressed data, one of uncompress software utilities is usually necessary. In IMC, most of compressed data are handled as they are. Namely, uncompress process is not necessary.

The annotated sequence entries of GenBank/EMBL databases are released as many multiple formatted compressed files. The largest of the files may contain tens of thousands of single entries in one file. When uncompressed and expanded the file, very large directory or folder consists of same number of single format files, is generated. IMC provides a unique function of expansion of such a file. Since every entry of GenBank/EMBL data has a taxonomy annotation in its source line.

(Discussion) With this function, local storage for such data is much reduced because the files are uncompressed only when imported into IMC.

(3) Importing features from other software

The results of gene finding software like Glimmer can be directly imported to IMC, imported features do not interfere with existing features. Most of sequence analysis software tools accept text format or FastA (Pearson and Lipman 1988) format files for the input. Since these formats are simple, they can not accommodate feature information inside. CSV format are usually used for the purpose of transferring feature information in GenBank/EMBL annotated files.

(4) Multiple GenBank/EMBL format can be expanded by scientific names

Downloaded GenBank files are archived and compressed into large size files. So it is not so easy to create ones own nucleic acids database on the local PC. IMC provides its users a unique function to create such database locally. By extracting scientific name from the source and organism of GenBank or EMBL format files, IMC expands such files into tree-structured directory like scientific taxonomy tree. This is called Lineage Expansion.

(4) Fast reading and feature map drawing algorithm

A Genbank or EMBL format nucleotide sequence file with features are read in a data structure described in **Figure 3.1**. This structure entry is entirely created on memory to speed up any process using this data structure and this entry is maintained on the memory until intentionally removed from it. When reading data from a GenBank file, IMC uses the stringbuffer of Java specification to accelerate reading speed. After reading, a GenBank format parsing process will start and store features information and its nucleotide sequence separately. The feature descriptions are also divided its location and qualifier values. These entries are created as a single path list to avoid a time-consuming creation of bi-directional path list.

To accelerate drawing speed of a feature map, one return per one request policy is taken, namely if one event such as mouse button click has detected, only one response will be returned. This policy may lead to require a lot of time to draw a entire map of large genome due to the fact that it contains thousands of features on a map. However, this policy has a lot of advantages such as simplified program structure, scale and distance independent access speed to any location of the DNA sequence. A reverse complementary operation requires only conversion of feature locations and reversing all the nucleotides. Deletion of one or more nucleotides among the sequence also requires simple conversion of feature locations downstream of the deletion sites as well as nucleotides insertion or substitution case.

3.3.2 Search functions

(1) Search by feature keys

After selecting feature keys to be searched, all the feature of selected feature keys are listed with its position, the feature containing DNA sequence, upstream gene

and downstream gene and if CDS, amino acids translation. The list can be saved as a CSV file and a multiple FastA file or single FastA files. Each line of the list is linked to the region of the feature map, which contains the pointed feature. Feature map jumps to show the pointed feature of the list in same scale.

(2) Keyword search

Keyword search among all the feature of current file is possible. Actually IMC searches qualifier's contents and, if found, features containing found qualifier's contents are listed in a window. Multiple keywords searching, with space delimiter, is also possible. AND operation would be applied to the results. For example, double keywords such as dehydrogenase pyruvate. can be specified.

(3) Search by sequence patterns

DNA sequence files are searched by short DNA patterns which are previously registered. For the registration of DNA patterns, mouse draw and right button click enables to register a DNA pattern easily. A DNA pattern file can be imported and registered as IMC DNA patterns. Copy and paste operation from other software is also provided. Pattern search regions can be restricted by inside CDSs, intergenic regions and upstream of CDS. Search results are listed in a window. One click of line of the list makes the feature map jump to show the site of found pattern. Found pattern sites can be registered as new features and saved in the annotated sequence file.

In DNA pattern search, IMC uses IUPAC international nucleotide code for the notation of nucleic acids and realize the ambiguous pattern search like N for any base, R for purine, Y for pyrimidine, etc.

3.3.3 Viewing and editing tools

IMC is implemented with various viewers of maps and sequences such as feature map viewer-editor with profile plot capability, reference map viewer with comparative genome functions, plasmid map viewer for cloning, circular genome map viewer for whole prokaryotic chromosome, sequence viewer with text formatting functions and GenBank/EMBL annotated file viewer with direct linking to feature map. These maps are closely linked to each other and can be located at desired positions of any feature immediately.

(1) Feature map viewer-editor with profile plot

This is also the main window of IMC, every sequence are the objects to be drawn on this map. Any feature on this map, is editable, namely, to be added or deleted, or modified with its context or by drawing colors or shapes. Up to the order of sub-billions of nucleotides are easily imported and shown as a feature map although adequate memory allocations are required to do this.

The horizontal scroll speed of the map is also high enough to move to any desired position of the map. In addition to that, zooming functions of the map is also quick. It can be especially mentioned that the map can scroll without stop, namely, in the case of circular chromosome, there is no end point on nucleotide and IMC can scroll along the circumference of the circular chromosome without stop. This function is becoming important factor when it comes to perform *in silico* cloning. For an example, the feature map of that of *Bacillus subtilis subtilis* 168 and *Escherichia coli* K-12 MG1655 are shown in **Figure 3.3** and **Figure 3.4**.

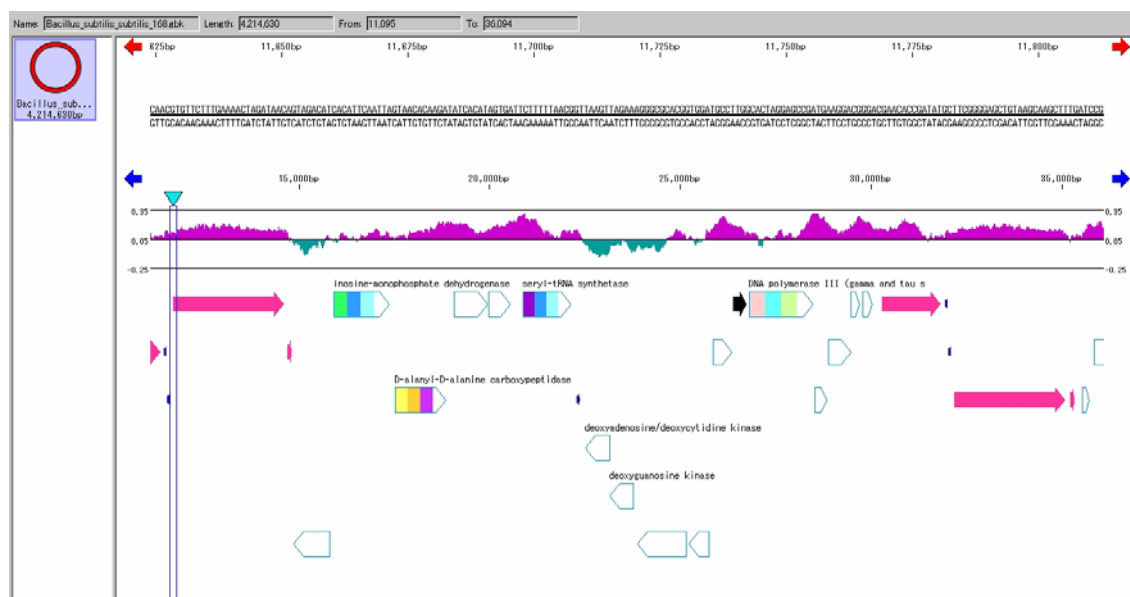


Figure 3.3 The feature map of the genome of *Bacillus subtilis*. (1) The wide arrows with tricolors are denoted by CDSs with EC number. (2) Red arrows are denoted by rRNAs.



Figure 3.4 The feature map of the genome of *Escherichia coli*. (1) On the top left, the icon for the currently displayed circular DNA is shown. (2) Nucleotide and amino acid sequence of specified region by the line cursor is shown. (3) GC skew profile is graphically presented as a two colored graph. (4) CDS assigned on the DNA are shown by arrows. The colors assigned for each CDSs can be changed by setting parameters.

(2) Reference map viewer

There are two types of reference map, one is for standard purposes and the other type is for comparative genomes. On the standard type of reference map, only one genome or one stretch of DNA is shown, while on the comparative genome reference map, as many genomes as the memory permits, are drawn in parallel. An example of comparative genome map is shown in **Figure 3.5**.

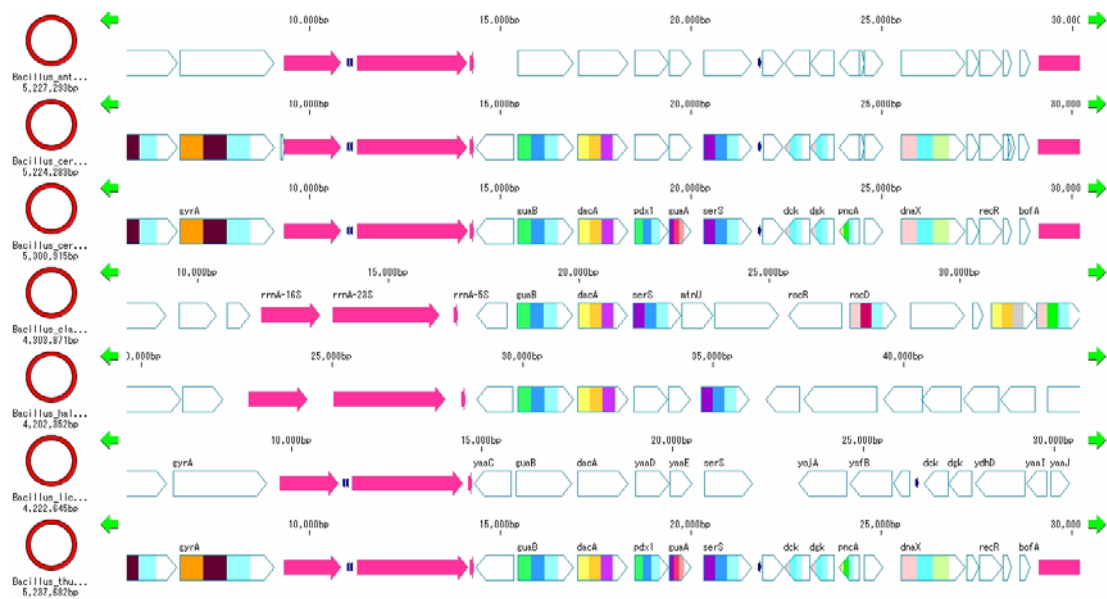


Figure 3.5 Reference map of *Bacilli*. Seven of *Bacilli* genomes are aligned by *dacA* genes. Any feature on the map can be searched by homology against each others or against the genome on the feature map. (1) On the left, icons denoted as the circular genomes, are shown. (2) Arrows painted in red are the features of rRNA, while the arrows painted in mosaic style are denoted as those of CDS with EC number given. Arrows without color painting are those of CDS with no EC number. The lines can be exchanged by dragging a genome icon to preferred position.

(4) Circular genome map viewer

To overview a circular chromosome, drawing of circular genome map is much requested. Although several drawing tools are available, they lack of integrated works with other functions. IMC provides this integrated and easy-to-drawing function of a genome map. There are four extra concentric circles on which any of registered feature keys are assigned to be drawn such as those of rRNAs, tRNAs, replication origins and/or promoters for example. A circular genome map of the *Bacillus subtilis subtilis* 168 is shown in **Figure 3.7**.

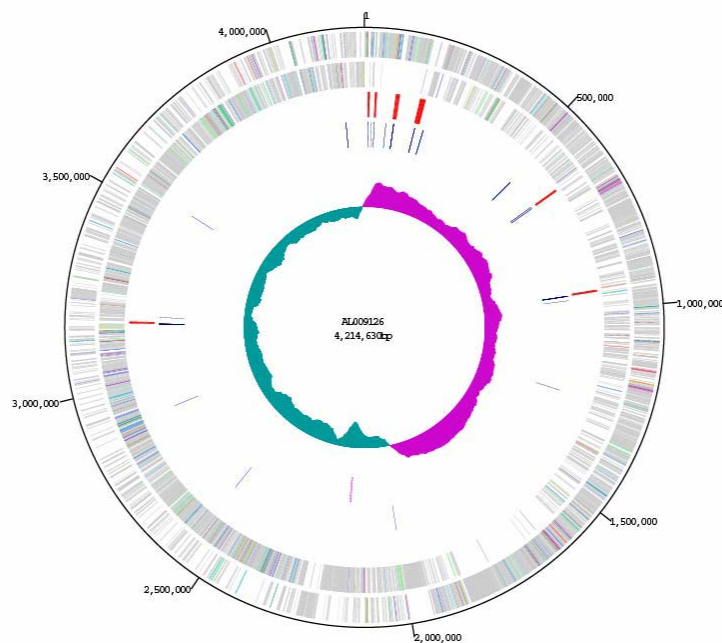


Figure 3.7 A circular genome map of *Bacillus subtilis*. From the exterior circle, CDSs on the forward strand, CDSs on the reverse complementary strand, rRNAs, tRNAs and two introns and GC skew plot are drawn. The feature keys designated on each circle are changeable by setting drawing parameters. The diameter of the circle can be also changed. This map can be written in as a PDF file.

(5) Sequence viewer

Any portion of imported sequences is shown in the sequence window with selections of formatting of nucleotides or amino acids such as bases per line or numbering method, or showing of the reverse complementary strand. If a CDS region are specified to be shown, all the candidates of initiation codons are located, in addition to that, fixing of the codon is performed by only click on the codon. If SD sequences are known and registered as features, they are also shown on the map. **Figure 3.8** is an example of them.

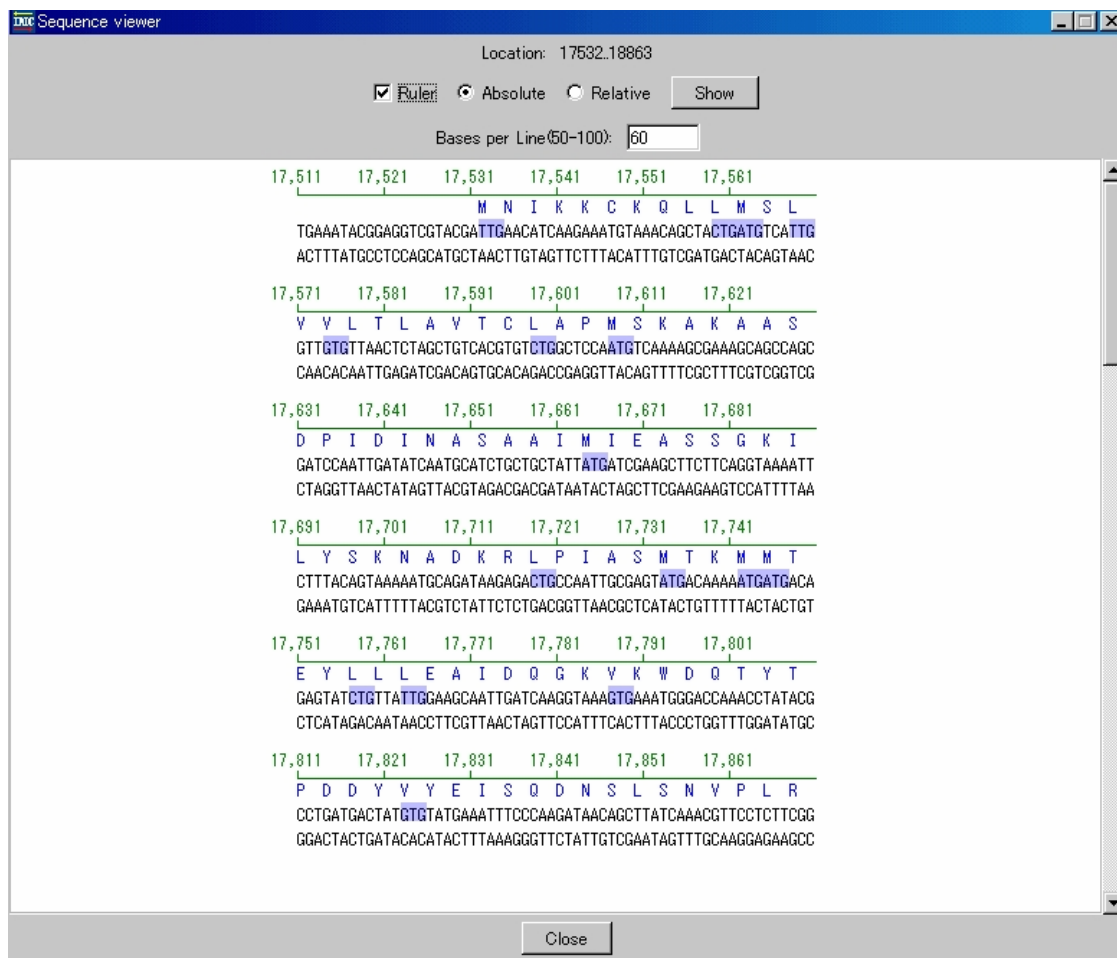


Figure 3.8 Sequence viewer showing a CDS region. Blue shaded bases are the candidates for the initiation codon. When another candidate for the start codon is considered to be actual start codon, a single click on the newly nominated start codon will replace the current start codon and also change the amino acids translated regions. If a RBS (Ribosomal Binding Site) is assigned around one of the start codon, it is used to be a good hint for the prediction of the actual start codon.

(6) GenBank/EMBL file browser

IMC imports annotated sequence files of GenBank or EMBL format and draw a feature map using the information recorded them, in addition to that IMC can also show the GenBank/EMBL format file as text viewer. Any line of text, including the description of the features or nucleotide sequence are directly linked to the feature map and one click on either of a text line or feature on the feature map, immediately bring the user to the exact corresponding location of its counterpart. In **Figure 3.9**, the GenBank/EMBL viewer is shown with its corresponding feature located on the feature map.



Figure 3.9 The Genbank/EMBL viewer. The blue shaded region of the viewer is directly link to the corresponding feature on the feature map behind and *visa versa*.

(7) Profile plotter

The profile plotter is a presentation method of continuous values obtained along DNA or amino acid sequences. A window approach is adopted to draw a profile on the top of the feature map. The profile drawn is precisely corresponding to the currently showing region of the map. Most of nucleotide compositions can be plotted such as GC content, GC skew, AT content, AT skew, cumulative skew of GC, that of AT

as well as single compositions of T, C, A and G. A sample of the profile is also shown in **Figure 3.9**.

3.3.4 Other supporting functions

(1) Inheritance of features in overlay fashion

IMC can import features with position, direction and annotation from other software tools. These features can be incorporated into current feature map in overlay fashion, namely this operation does not overwrite any exiting feature on the map. Therefore, co-researchers can share results without interfering each other.

(2) Lab note tools

As described in **Chapter 4**, IMC is implemented with a set of *in silico* cloning functions. For the purpose of recording the history of such experiments *in silico*, this function is implemented. Every experiment *in silico* is recorded with a time stamp on it and later recalled and compiled as a lab note book.

(3) Feature statistics

IMC can afford to accommodate sub-million order features in a recording unit. When handling such a huge set of data, a statistical summary is very convenient for the user to overview and reach to his desired target. IMC provides this function as Feature Statistics. One click of the button, when a map with a huge number of features is shown, pops up a window of statistics such as shown in **Figure 3.10**. First, each number of existing features is indicated on the window, after selecting some, further details of compositions or attributes of each feature is listed.

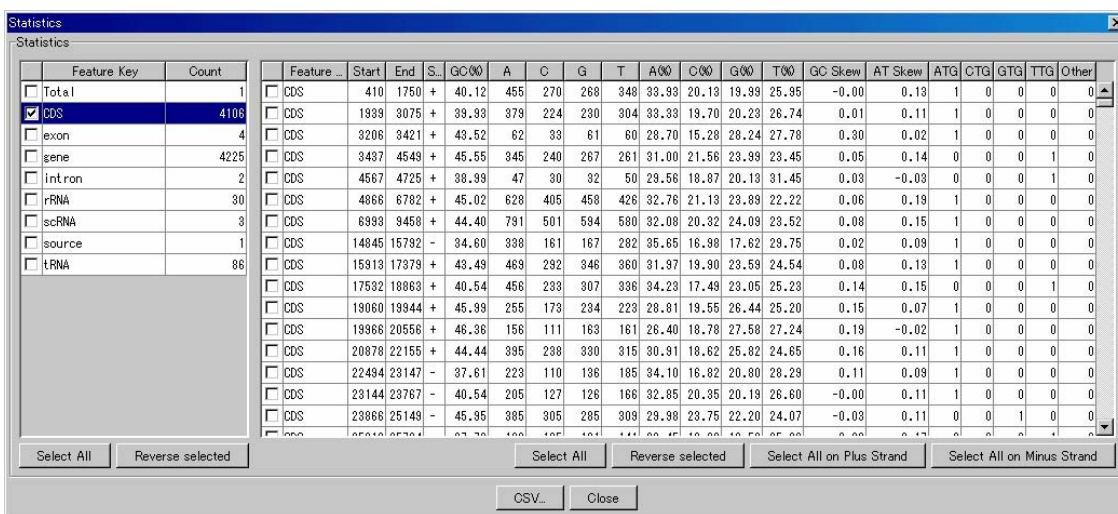


Figure 3.10 Feature statistics window. On the left, the existing features and their counts are listing. On the top of the list, A special non existing feature name Total is shown. This is virtually introduced feature to describe statistics about whole sequence. On the right, details of each feature statistics are shown. Among them, initiation codons count is also visible.

(4) Coloring methods on features

If CDS feature has an EC Number, the CDS can be shown in the classified color for the EC number. There is also mosaic coloring for three digits of EC number. The different colors can be assigned for each of the first three digits of EC number.

CDS features can be colored by the optionally assigned code when it is recorded on the qualifier */Classification=*, IMC draws color features according to its assigned color on the feature map.

(5) Keeping the modification history on features

Any touch on any feature is recorded with a time stamp and searched by keywords later on. From the result list of the searching, only one click moves the feature map into exact position of the modified feature.

3.3.5 Structure analysis tools

(1) Extraction of ORF candidates from an unknown microbial sequence

IMC extracts ORF candidates from newly sequenced data with listing all the stop-to-stop codon regions of a given length. Then find the first start codon downstream of an ORF candidate, and translate the frames into amino acid sequence, simultaneously generate its own databases for homology search.

(2) Codon usage table

A Codon usage table is widely used in basic analysis of protein coding regions. IMC provides this function in three different manners, namely per CDS, per regions and per genome. One click on a CDS leads to a pop up of the codon usage table. One sweep of mouse drag through any region on the currently shown feature map, and one click is enough to show the total and average codon usage of CDSs included in the region. One push of the codon usage button will show the codon usage of total or average among all the CDSs in the genome. An example is shown in **Figure 3.11**.

		Bacillus_subtilis_subtilis_168.gbk CDS's : 4,106 bases in CDS : 3,677,168 codons : 1,225,722						
		codon type: <input checked="" type="radio"/> mRNA <input type="radio"/> DNA		data type: <input type="text" value="Composition"/> option: <input type="checkbox"/> Amino Acid Composition				
2nd	1st	U	C	A	G			
U	UUU	0.0307	UCU	0.0128	UAU	0.0226	UGU	0.0036
	UUC	0.0142	UCC	0.0080	UAC	0.0120	UGC	0.0044
	UUA	0.0191	UCA	0.0148	UAA	0.0021	UGA	0.0009
	UUG	0.0154	UCG	0.0064	UAG	0.0005	UGG	0.0103
C	CUU	0.0231	CCU	0.0105	CAU	0.0153	CGU	0.0075
	CUC	0.0109	CCC	0.0033	CAC	0.0074	CGC	0.0085
	CUA	0.0049	CCA	0.0070	CAA	0.0196	CGA	0.0041
	CUG	0.0232	CCG	0.0159	CAG	0.0187	CGG	0.0065
A	AUU	0.0370	ACU	0.0087	AAU	0.0222	AGU	0.0066
	AUC	0.0269	ACC	0.0086	AAC	0.0171	AGC	0.0142
	AUA	0.0094	ACA	0.0222	AAA	0.0493	AGA	0.0107
	AUG	0.0270	ACG	0.0145	AAG	0.0211	AGG	0.0040
G	GUU	0.0192	GCU	0.0189	GAU	0.0330	GGU	0.0126
	GUC	0.0173	GCC	0.0158	GAC	0.0186	GGC	0.0233
	GUA	0.0133	GCA	0.0216	GAA	0.0489	GGA	0.0216
	GUG	0.0177	GCG	0.0201	GAG	0.0231	GGG	0.0112

Figure 3.11 Codon usage table of *Bacillus subtilis*. The notations of codons are changeable by pushing the 'mRNA' button or 'DNA' button on the top. The 'Composition' or 'Frequency' is selected.

3.3.6 Homology search functions

CDSs in any region are homology searched automatically and the annotation of subjects is copied. After selection of CDSs and the database for homology search,

automated search is started. The search results are stored on each feature and referred afterwards permanently. Color classification by homology and overlap score can be applied. By using annotated amino acid databases, annotation of hit subject features can be automatically copied to the query feature. Alignment and annotation information are also shown and related list can be saved as text files.

(1) Homology search with BLAST

When imported into IMC, any annotated sequence data is automatically converted into a BLAST database, so that it can be searched by BLAST program later. In addition to that, BLAST is used in other ways in IMC. When it comes to find out repetitive sequences among a genome sequence, BLAST is also activated and the results are parsed to be reported.

(2) Comparing related DNA sequences on the reference map

CDS or RNA features can be searched by its homology against the reference DNA sequence. If there are homologous elements on the reference DNA, the results are listed in a window. One click on any line of window makes the reference map to jump and show the exact feature on the reference map. Homologous region of CDSs or RNAs are shaded. Homology search can be done with both nucleic acid and amino acid translation of CDS. If previously created, homology search against amino acid database is also done.

3.3.7 Multiple alignment tools

After finishing the homology search, a multiple alignment of the found homologous sequences is usually performed followed by a drawing of phylogenetic tree finally. This is the main stream of function analysis.

(1) ClustalW

IMC uses ClustalW function for the multiple alignment analysis. Actually, it activates ClustalW with a set of homologous sequences and obtains results.

(2) Phylogenetic tree viewer

On the contrary to the fact of using of ClustalW for a multiple alignment, as for the phylogenetic tree drawing IMC provides its own function of drawing such trees. When drawing a tree, aligned file from ClustalW is used. IMC can draw one of three types of the trees as shown in **Figure 3.12**.

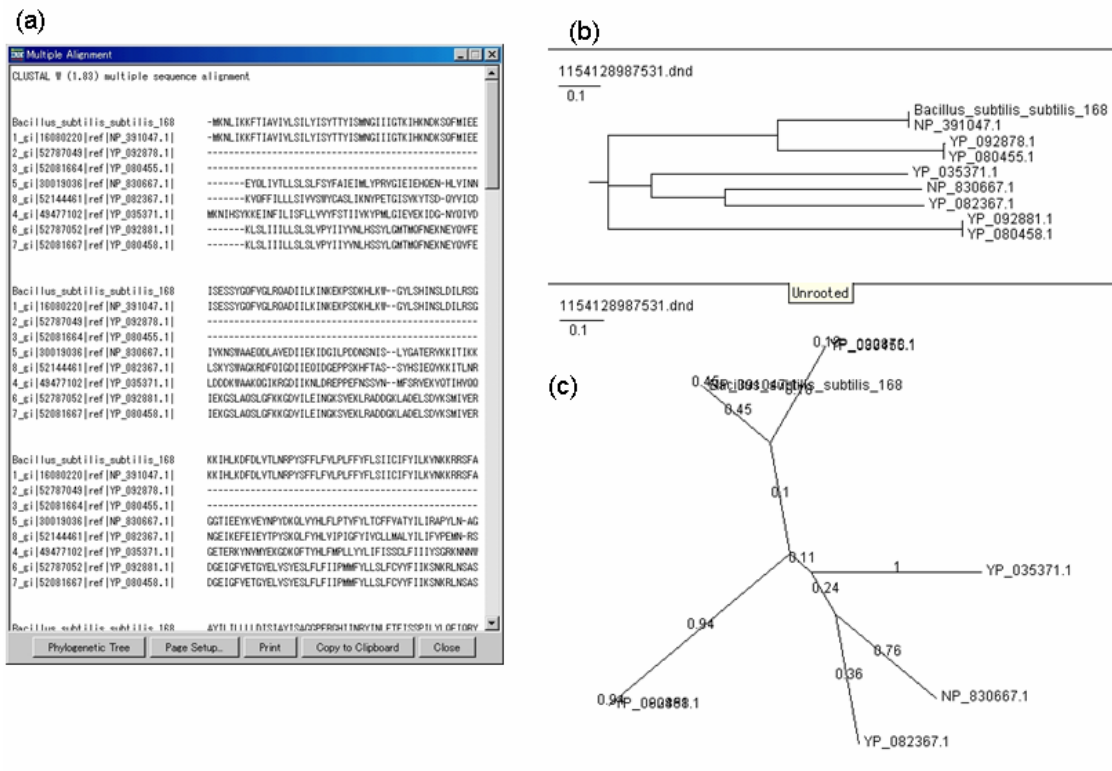


Figure 3.12 Multiple alignment and phylogenetic tree. (a) The multiple alignment of homologous amino acids sequences are shown on the left. (b), (c) The phylogenetic trees of the alignment are shown on the right.

(3) Import of cDNA and incorporate them on the map

cDNA sequences can be imported and pasted on the feature map as new features, with identifying exon and intron structure. Even after finishing the genome sequencing projects like the Human genome project, there still remain a lot of fragments which are not united yet. On the other hand, there are also several mRNA or cDNA sequencing projects that have a lot of mRNA or cDNA sequences. It is useful to paste cDNA onto its own genomic sequences. However, this is a clumsy work, because we do

not know on which chromosomes each cDNA should be located. IMC provides complex searching and pasting tool to locate the exact position on chromosomes for each cDNA sequence to be pasted. After completion of pasting, IMC automatically loads all the genomic fragments on feature map. Sometimes, cDNA can not be pasted along its whole length, both ends are often not homologous, IMC assigns such cDNA as partially pasted on genome, and classified it as a misc_RNA feature, instead of mRNA feature for exactly pasted cDNA. A click on mRNA on genomic sequence activates the alignment between genome and cDNA to examine the completeness of pasting.

3.3.8 Comparative genome tools

(1) Comparative genome map

Multiple genome alignment is obtained in simple operation. *in silico* MolecularCloning Genomics Edition (IMCGE) provides multiple genome alignment function with very simple and comprehensive operation. As far as platform's memory allows, a number of whole genomes can be compared simultaneously as shown in **Figure 3.3**. Alignment map would be aligned by clicking the concerned feature. This aligned map is directly printable on printer or created as a PDF file to be edited further by drawing software like Adobe Illustrator.

(2) Dot plot between two genomes

Dot plotting of two genomes is another application of the BLAST algorithm. IMC uses the one of BLAST algorithms, named megaBLAST which is implemented with fast processing of large size sequences. Such a dot plot between *B. halodurans* and *B. subtilis* is shown in **Figure 3.13**.

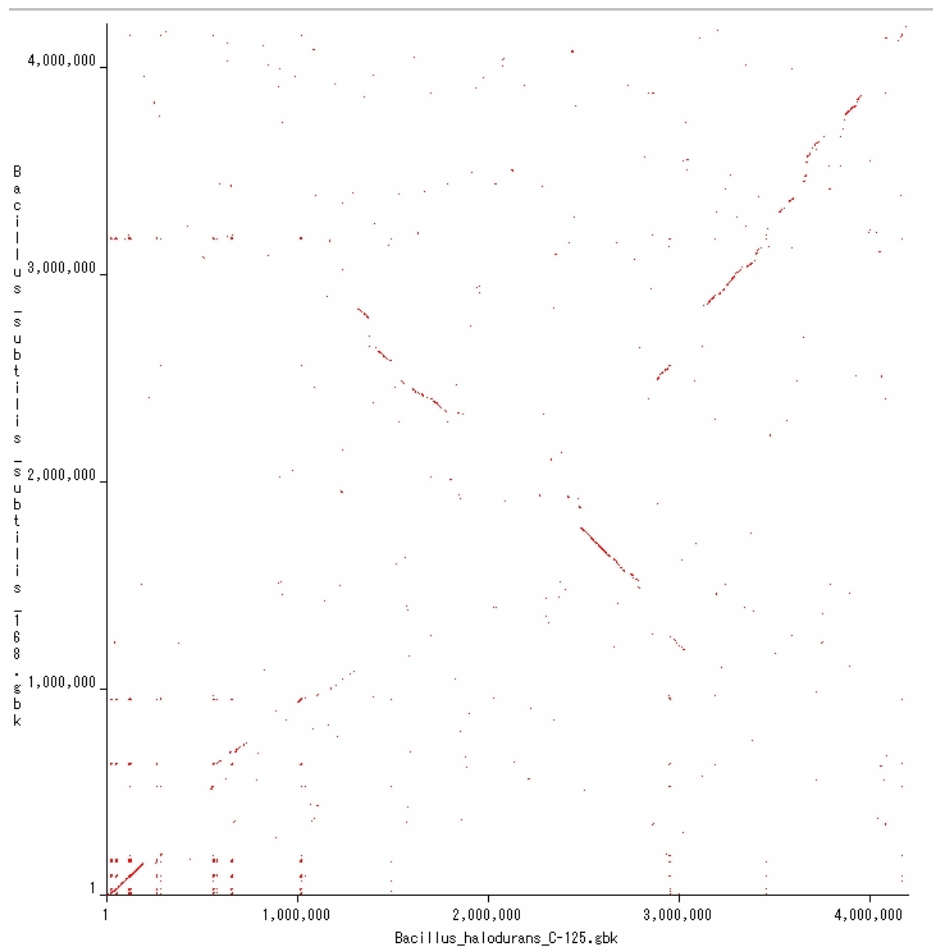


Figure 3.13 Dot plot between *Bacillus halodurans* and *B.subtilis*. The Y axis denotes the position of DNA sequence from *Bacillus halodurans*. The Y axis denotes the position of DNA sequence from *Bacillus subtilis*. The red points denote the homologous regions between the two genome sequences.

3.4 Discussion

IMC software has many advantages of the existing software tools. Especially, it has high speed import performance and handling of sub-millions order of features, as well as other unique functionalities.

(1) Expanding usability of features and qualifiers

The most contribution of IMC to molecular biology is that IMC expanded the usages of feature keys and qualifiers of the GenBank/EMBL convention. Previously, the convention is used only for the permanent recording purposes, however I proved that further dynamic and wider usages are possible (see also **Chapter 4** and **Chapter 5**).

(2) High speed importing of whole genome GenBank/EMBL files

IMC imports typical GenBank/EMBL format genome annotation files with higher speed compared to existing programs. In the **Table 3.1**, importing times for large genome sequences with annotations are listed. Even a largest of the human genome chromosomes can be imported within minutes.

Table 3.1 Importing performance of IMC. Six different sizes of genome sequences with annotations are imported and measured the speed until the feature map is first shown in the map.

File Name	Sizes in bps	File Import Time in seconds
Buchnera aphidicola APS	640,681	3.92
Bacillus subtilis subtilis 168	4,214,630	7.46
Streptomyces coelicolor A3(2)	8,667,507	11.55
Arabidopsis thaliana chr.I	30,432,563	27.34
Homo sapiens chr.2 frag.17	84,213,156	64.16

(3) Accommodation of sub-millions order features

IMC can handle annotated files with a huge number of features on them.

Actually, it affords to accommodate them in large sized single files. As described later in **Chapter 5**, a huge number of features, such as those of the tiling microarrays, are also stored in a single GenBank/EMBL format file. Even if handling of small prokaryotic genome, its file size would be around 50-100Mbytes. However, operation performance is not affected much. This good performance has been achieved by the improved technology of Java programming.

(4) High performance scrolling or zooming of maps

The good performance of scrolling and zooming the feature map with profile drawing or array map drawing has also been achieved. The measured speeds of feature map scrolling are about 19 seconds per million base (Mbp) and 58 seconds per Mbp, without drawing the profile of GC content and with drawing it, respectively. The measured speed of zooming from maximum scale to minimum scale in the case of *B. subtilis* (4,214,630bps), are about 4.5 seconds, 91 seconds and 102 seconds for without GC content profile drawing, with GC profile drawing of a relative window size and with GC profile drawing of an absolute window size, respectively.

(5) Variety of editing functions

IMC is implemented with a variety of editing functions. First of all, on the feature map, every feature on it is editable. Thousands of insertion of new features on the map require only a second, nevertheless if there are already a lot of features registered on the map, it will be much slower than that. In the case of a tiling microarray, the total number of features to be registered in a GenBank/EMBL file amounts to 100-300 thousands. The scrolling speed of the feature map with a tiling microarray is 21 seconds per Mbps and 35 seconds per Mbps, without drawing profiles and with profiles, respectively.

(6) Linked map with search results

Most of the searching results lists in IMC are directly linked to the locations of the found features. Consequently, users can jump to the desired site on the map immediately. Such cases are, that of homology search, keywords search, feature key search, lab note entry search, update entry search etc. In addition, any line of the GenBank/EMBL text file whose features are currently shown on the feature map is directly linked to the corresponding feature or nucleotides.

(7) Multiple alignments between entire genomes

IMC is implemented with a unique alignment method of genome level. Tens of whole genomes can be aligned according to following procedure. The genomes are not aligned just after importing, rather the result of homology search is first obtained and then genomes are aligned according to the homologous features of genes. Exchange of orders of genomes is easily handled by only dragging the icon designated for one genome.

Chapter 4

***in silico* experiments of molecular cloning**

4.1 Introduction

Cloning is the most popular experiment routinely performed in the molecular biology field. The experiments consist of basic routines such as DNA digestion by restriction enzyme, ligation and PCR, and so forth and routines for detection, such as gel electrophoresis (**Figure 4.1**). A wide variety of combination of these basic routines, are repeated in laboratories.

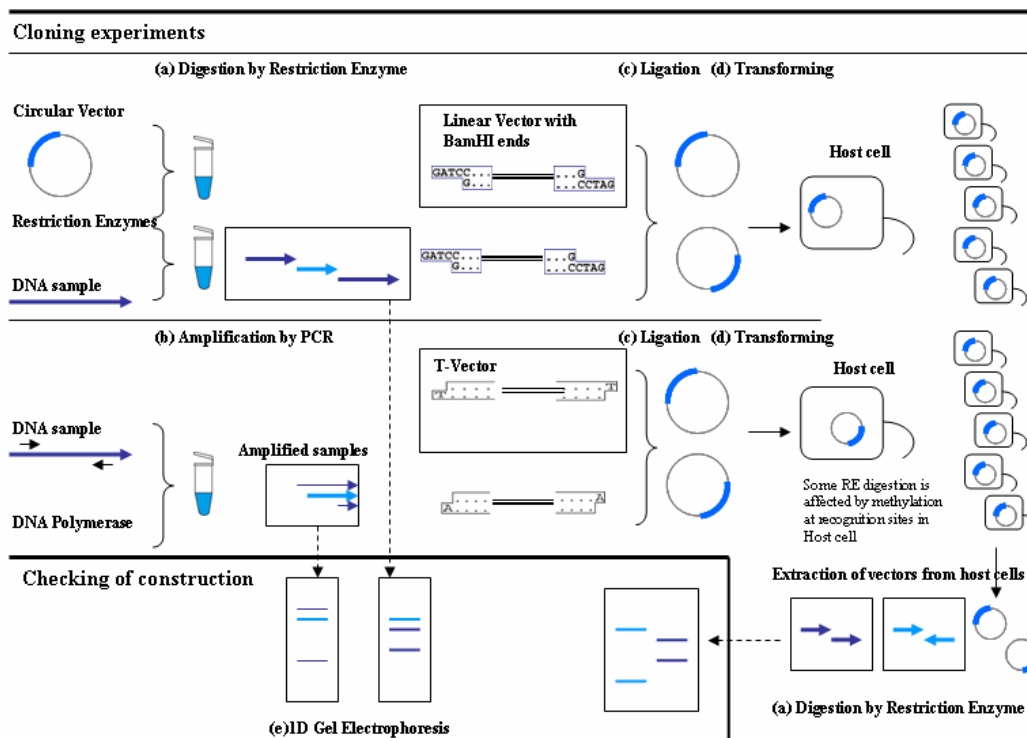


Figure 4.1 Flow of cloning experiments. (a) Digestion of DNA by a restriction enzyme BamHI produces linear DNA fragments with 5' sticky ends. A circular vector is linearized by digestion. (b) Amplification of DNA by PCR produces DNA fragments between two priming sites recognized by the given PCR primers. Occasionally, ambiguous priming produces different sized PCR products which can be seen by 1D gel electrophoresis. (c) The linear vector and a DNA fragment are ligated together with the presence of an enzyme ligase if the end shapes of the both ends are matched. This reaction produces two different circular vector with the insert is reversely ligated to the vector. (d) The vector with the insert is transformed into a host cell. The host cells are cultured until a number of clones are obtained. (e) 1D gel electrophoresis separates many DNA fragments by size.

Experiments in molecular biology are defined as the series of basic routines starting from the sample DNA molecules, after receiving various chemical or physical changes such as purification, amplification, modification or decomposition, until obtaining final target materials which are regarded as more feasible to be detected in amount or in property as the proof of the chemical reactions. For example, when DNA samples are applied into a test tube with solution of a restriction enzyme, the DNA molecules in the tube undergo chemical changes of digestion at the particular sites of the molecules. The solution is applied into a well of agarose gel for electrophoresis. Then solution of ethidium bromide is applied on the agarose gel. Finally, the gel is taken by a photograph under the UV lighting. It is suggested that a variety of consecutive experiments would be performed until the final target molecules which are detectable with certain experimental devices, are obtained.

In this work, I attempt to perform the molecular biological experiments *in silico*, in other words, in computer, as one of simulation tools of performing the experiments *in vitro*. The word, experiment, has wide meaning. In this thesis, I treat only experiments in the field of molecular cloning in which experiments are handled with mainly on DNA molecules. Therefore, I had focused on *in silico* molecular cloning and describe feasibility and effectiveness of *in silico* cloning and provide formulation of each component of the virtual experiments, such as DNA samples, reagents or reactions.

Prior attempts to perform *in silico* cloning were reported. A computer program named VIRTLAB was developed (Iazzetti 1998). VIRTLAB simulates plasmid preparation, digestion with restriction enzymes, agarose gel electrophoresis, DNA cloning and DNA sequencing. However, these simulated experiments in VIRTLAB are performed without accompanying DNA sequences processing. Visual Cloning 2000 is another software package providing restriction enzyme analysis,

Most software, when handling DNA, defines it as a single sequence. In nature, the DNA is, actually, consisting of two separate sequences. Most of the regions in DNA have bases complementary to each other. Accordingly, it is natural for such software to adopt a single sequence for the expression of DNA. However, as for the both ends of the molecule, it is quite different. They likely take single strand shapes at the end of DNA which takes an important role especially in ligation against other DNA end points. Most of such software tools do not have an expression about the end shapes of DNA molecules. Without information on the ends of DNA molecules, *in silico* cloning experiments are not feasible. Here, I emphasize that it is necessary to express the end shapes of DNA before developing *in silico* cloning software. Therefore, I propose extra

expression about the end forms of DNA molecules at first. Then, I discuss the *in silico* cloning software.

Genome projects such as the human genome, or many microbial genomes, have targeted the DNA sequencing across whole genomes, and most of these genomic DNA sequences are publicly available (Benson et al. 2006, Cochrane et al. 2006). DNA sequences of artificial vectors and amino acid sequences of restriction enzymes, where both materials are usually used in cloning experiments, are also available in nucleotide or amino acid databases (Roberts 2005). Thus, the infrastructure for launching *in silico* experiments has already been established by these preceding databases. For an organism, when its DNA sequence is determined, it becomes possible to make a logical plan for performing experiments, such as cloning of specific genes on the genome. In the present study, I introduce simulation of experiments for molecular biology referred to as *in silico* cloning focused on molecular cloning based on genome information.

4.2 Methods and Algorithms

I describe the definition of *in silico* cloning and requirements for the software. Typical functions of *in silico* cloning are implemented on this newly developed software on the basis of the required definitions and format for the DNA samples to describe state of these molecules. Then the feasibilities of performing such experiments *in silico* are discussed.

Although many software tools have been developed to implement cloning experiments, few of them have focused on this treatment of consecutive operations of cloning. A guideline to *in silico* experiments should be established before describing about the software itself. That is, such software would satisfy at least the minimum requirements such that consecutive experimental routines could be performed and could describe and visualize the states of molecules such as single stranded ends of DNA, circular double stranded DNA with staggered and blunt ends and so on as described later in this chapter. To satisfy these requirements, I will propose the formats for sample DNA.

4.2.1 Guideline to *in silico* cloning software

So as to make a guideline that leads to *in silico* experiments, I, hereafter, propose some requirements, although this is, in the same time, regarded as definitions on *in silico* experiments, for software to be implemented as follows:

- (1) A recording unit of a DNA molecule should be one-to-one correspondence to each molecule participating in a reaction. Namely, information concerning the molecule should be stored as a single corresponding recording unit.
- (2) After a reaction has finished, information on input molecules would be inherited to the resulting products. Namely, whenever DNA molecules are fragmented into several smaller and partial ones or amplified as many clones, or synthesized into fewer molecules, information on the corresponding regions of them must be exactly transferred to those of newly produced molecules.
- (3) Whenever a DNA molecule is synthesized together from two different DNA molecules by covalent bond, the combined molecule should have a single DNA sequence without trace of reaction left. That is, any evidence can not exist to show

where or when the bonding reaction occurred.

- (4) In a circular DNA, visualization, detection of recognition sites and reaction, which occurs on any sites or bases of the DNA, should be equivalently handled. Namely, this means that in circular DNA, any part of DNA should not be any gap between any bases.
- (5) An enzyme, as a protein, can be treated as a catalyst, *i. e.*, nothing of these molecules should be affected or changed after any reaction. However, this is a temporary requirement adaptable only for the time being, until adequate information on such enzymes would be formally annotated on the Genbank or EMBL files.
- (6) As for the selection of the viewing strand of DNA, there should not be any significant weight between the selected strand and its complementary. And moreover, there should be a function of switching to the reverse complementary strand from current strand, with all the features kept on it.

4.2.2 Samples and products of *in silico* experiments

```

LOCUS       AL009126       2163 bp       DNA                               CON 07-JUL-2003
DEFINITION  Bacillus subtilis complete genome.
ACCESSION  AL009126
VERSION    AL009126.2   GI:38680335
KEYWORDS   complete genome.
SOURCE     Bacillus subtilis subsp. subtilis str. 168
   ORGANISM Bacillus subtilis subsp. subtilis str. 168 Bacteria; Firmicutes;
             Bacillales; Bacillaceae; Bacillus.
REFERENCE  1 (bases 1 to 4214630)
.....
.....
.....
FEATURES             Location/Qualifiers
     end_type       1
                       /endtype=-2,2
                       /new=060504070352
     source           1..2163
                       /organism="Bacillus subtilis subsp. subtilis str. 168"
                       /mol_type="genomic DNA"
                       /strain="168"
                       /db_xref="taxon:224308"
     misc_feature     <1..623
                       /gene="yaaT"
                       /locus_tag="BSU00320"
     misc_feature     <1..623
                       /function="unknown"
                       /note="yaaT; similar to unknown proteins"
                       /codon_start=1
                       /transl_table=11
                       /protein_id="CAB11808.1"
                       /db_xref="GI:2632299"
                       /db_xref="UniProt/Swiss-Prot:P37541"
BASE COUNT       718 a   383 c   526 g   536 t
ORIGIN
1 atggcgatct tctcattgta gaagaaaata aacaggaagc actatcagca tttgatatct
61 gccaaaagaa agtgattgag catggcttgg atatgaagct ggtcgatggt gaattcacgt
121 ttgatcgcaa taaagtcatt ttttacttca ctgctgacgg ccgagtcgac tttagagagc
.....
//

```

Figure 4.2 Example of GenBank format file. The end shapes of the DNA is described by a feature key *endtype* with its qualifier *endtype=n, m*. The above example shows that the DNA has two single stranded bases at both 5' ends. The bases are recorded as the first two and the last two bases of its nucleotide sequence part.

In the international nucleotide database conventions, most of biological characteristics, such as genes, are expressed as feature keys and their qualifiers as shown in **Figure 4.2**. Feature keys are defined as units which have functions or structures of biological meanings, while their qualifiers providing values or contents.

When implementing software for *in silico* experiments, it is desirable to use these descriptions as much as possible. The reason is due to the fact that a large amount of nucleotide sequence is submitted to the international databases and anyone can access such databases and obtain the requested data anytime. Therefore, utilization of these international nucleotide database format records increases the data availability in a great extent. However, current conventions of GenBank/EMBL do not always satisfy the requirements to perform *in silico* experiments. Therefore, some extension for the existing conventions would be necessary. Software implementation is discussed in the following examples.

4.2.3 Description of sample DNA molecules

Extension required for qualifiers, is description of the end shapes of DNA fragments after digestion by restriction enzymes or PCR amplification. The end shapes of DNA fragment after RE digestion or PCR are likely to take sticky ones, namely to be single stranded at the end. To describe these end shape, categories of feature keys and qualifiers must be extended as shown in **Figure 4.2**. The annotation based on endtype is introduced as a new feature key to describe this molecule has an expression about the end shapes, and new qualifier named *endtype=m,n* are introduced to describe the type of ends created after reaction (**Figure 4.3**). Actually, it is not necessary to describe about the cause of reaction for the new feature key, because we generally do not know the cause of reaction.

Restriction Enzymes	Fragments	Annotation
(a) BamHI	$ \begin{array}{c} 5' \text{GATCC} \dots \text{G} \text{ 3'} \\ \text{ 3'G} \dots \text{CCTAG} \text{ 5'} \end{array} $	<code>/endtype=-4, 4</code>
(b) BclI	$ \begin{array}{c} 5' \text{GATCA} \dots \text{T} \text{ 3'} \\ \text{ 3'T} \dots \text{ACTAG} \text{ 5'} \end{array} $	<code>/endtype=-4, 4</code>
(c) EcoRI	$ \begin{array}{c} 5' \text{AATTC} \dots \text{G} \text{ 3'} \\ \text{ 3'G} \dots \text{CTTAA} \text{ 5'} \end{array} $	<code>/endtype=-4, 4</code>
(d) AatII	$ \begin{array}{c} \text{ 5'C} \dots \text{GACAT} \text{ 3'} \\ \text{ 3'TACAG} \dots \text{C} \text{ 5'} \end{array} $	<code>/endtype=4, -4</code>
(e) PvuI	$ \begin{array}{c} \text{ 5'CG} \dots \text{CGAT} \text{ 3'} \\ \text{ 3'TAGC} \dots \text{GC} \text{ 5'} \end{array} $	<code>/endtype=2, -2</code>
(f) EcoRV	$ \begin{array}{c} 5' \text{ATC} \dots \text{GAT} \text{ 3'} \\ 3' \text{TAG} \dots \text{CTA} \text{ 5'} \end{array} $	<code>/endtype=0, 0</code>
(g) SmaI	$ \begin{array}{c} 5' \text{GGG} \dots \text{CCC} \text{ 3'} \\ 3' \text{CCC} \dots \text{GGG} \text{ 5'} \end{array} $	<code>/endtype=0, 0</code>

Figure 4.3 Examples of DNA fragments digested by restriction enzymes. (a) 5' sticky ends of a BamHI digested DNA fragment have four single stranded bases. The end shapes are annotated as `/endtype=-4, 4`. (b) 5' sticky ends of a BclI digested fragment. (c) 5' Sticky ends of an EcoRI digested fragment. (d) 3' sticky ends of an AatII digested fragment. (e) 3' sticky ends of a PvuI digested fragment with two single stranded bases. (f) Blunt ends of an EcoRV digested fragment have no single stranded base. The end shapes are annotated as `/endtype=0, 0`. (g) Blunt ends of a SmaI digested fragment.

In the case of ligation, it is possible to simulate a ligated DNA fragment derived by two DNA fragments if each ends of DNA fragments match in a complementary style. This is discriminated by using the `endtype=m, n`, where m means 5' single stranded base length and n means 3' single stranded base length as shown in **Figure 4.3**. In the case that ligation would occur, a pair of ends match in length along single stranded bases and also match in complementary sequences. Let's take the ligation between the end digested by BamHI, and the end digested by BclI, as an example (**Figure 4.4a**). The ends digested by BamHI, are expressed by `endtype=4,-4`, while the ends digested by BclI are also expressed by `endtype=4,-4`. Ligation occurs between 5' end of one DNA fragment and 3' end of another fragment, so the discriminating process goes on like this. (1) Among the four possible combinations (if

the reverse complementary fragments are considered), the first one is taken to be examined. (2) The first combination (4, -4), (m, n) means the testing of 5' end of a BamHI fragment and 3' end of a BclI fragment. (3) Simple sum is performed between the digits in the bracket, if the answer is zero this means shapes are matched for ligation between the two ends. (4) Then, reverse complementary sequence between the two single stranded bases are examined, and the BamHI product has single strand of GATC at the 5' end, while the BclI has single strand of CTAG at the 3' end. Thus, the combination is proved to be ligated. (5) Remaining three combinations are examined and all the combinations are proved to be ligated. Further examples are shown in **Figure 4.4**. This means two circular DNA would be produced by ligation reaction between the two DNA fragments. In addition, it is better to implement suppress of ligation if phosphate group is removed from one or both ends by some enzymes like phosphatases. Thus, the implementation of the feature key dephosphorylation is necessary in view of the higher visibility, integrity and compactness for describing the state of DNA molecule.

Examples	Fragment A	Fragment B	End shape Check	Complementary Check
(1) BamHI + BclI	BamHI	BclI		
	5' GATCC.....G 3' 3' G.....CCTAG	5' GATCA.....T 3' 3' T.....ACTAG 5'		
endtype	-4	4 -4	4	0 0
(2) BamHI + EcoRI	BamHI	EcoRI		
	5' GATCC.....G 3' 3' G.....CCTAG	5' AATTC.....G 3' 3' G.....CTTAA 5'		
endtype	-4	4 -4	4	0 0
(3) BamHI + AatII	BamHI	AatII		
	5' GATCC.....G 3' 3' G.....CCTAG	3' C.....GACAT 5' 5' TACAG.....C 3'		
endtype	-4	4 4	-4	8 -8

Figure 4.4 Examples of ligation reactions. Two fragments digested by BamHI and BclI can be ligated by End shape check and complementary check. Two fragments digested by BamHI and EcoRI can not ligated by complementary check. Two fragments digested by BamHI and AatII can not ligated by End shape check and complementary check.

Description of recognition site sequences, cleavage patterns, and affection by methylation of recognition sites

The information on recognition site sequences and cleavage pattern of each restriction enzyme is registered in REBASE (Roberts et al. 2005), a restriction enzyme database. However, according to the requirements of *in silico* experiments, entries from amino acid databases would be transferred as input data for restriction enzyme. Namely, it is desirable to use rather amino acid data entries which describe RE recognition site sequences, cleavage patterns and possibility of affection by methylation at the site.

4.3 Results

4.3.1 *in silico* vs. *in vitro* experiments

In comparison of *in silico* experiments with *in vitro* experiments, let us consider upon the digestion of a DNA sample by restriction enzymes. Information on DNA is collected in the international nucleotide database such as GenBank or EMBL. As each sequence of DNA is recorded as one individual entry, it is also appropriate to use it as a sample to *in silico* experiments. In *in vitro* experiments, DNA samples are applied to a micro tube, followed by applying solution of restriction enzymes, then a reaction is allowed to progress in the tube, then digested fragments are obtained.

In silico version of this experiment is processed as software programs read the corresponding data file which describes a DNA sample sequence and annotations, and stores these data into a folder designed as a simulated micro tube. As soon as the sample is read, the corresponding feature map is drawn with many features indicated by a variety of figures. Typical experiments can be done against this sample. All the implemented experiments are applied to the DNA sample. For example, digestion by restriction enzymes is performed as follows. In *in silico* version, the restriction enzyme database is searched for cutting appropriate sites of DNA sequence and target enzymes are selected to be applied to the tube. Then, the reaction would be started until completely digested fragments are obtained. In the first step of digestion, a list of restriction sites are shown, then selection of sites leads to final digested DNA fragments, each of them is described and saved as a GenBank or EMBL format file.

Actually, IMC consists of a set of experiment functions ranging from digestion by restriction enzyme, PCR, ligation, and so forth as shown in **Table 4.1**. An experimental operator is defined as *in silico* process corresponding to the equivalent experiment *in vitro*. Thus, experiments *in silico* are regarded as computing of DNA sequences by some of the experimental operators to obtain the resulting sequences.

4.3.2 An example of confirmation viewer which shows the target molecule as it is

Most of plasmids and prokaryotic chromosomes have circular structure *in vitro* or *in vivo*. Such circular structure has actually no end point along its nucleotide

sequence. However, when it comes to record the nucleotide sequence in a text file, it can not be avoided to describe them as a string of characters with the start character and the end one. A seamless viewing and drawing of circular DNA sequence is thus required as well as for recognition or reaction across the point in *in silico* experiments.

4.3.3 DNA cloning *in silico*

DNA cloning means the extraction of a particular region of DNA (often that of a particular gene) from a genomic DNA or from other DNA sources, then the extracted DNA fragment is inserted into a plasmid vector. After transformed them into host cells, a lot of clones are obtained as the host cells are cultured. Cloning any designated segment of DNA from a genome is one of the most important techniques of recombinant DNA technology, as it is the starting point for understanding the function of any region of DNA within the genome. **Figure 4.1** shows protocol for DNA cloning. DNA cloning can be carried out by mainly three experimental protocols, digestion of DNA by restriction enzyme (RE digestion), polymerase chain reaction (PCR), ligation reaction, and transformation of the vector constructed to bacterial cells. In addition, to check the band patterns of gel electrophoresis are generally used to confirm whether or not DNA fragment of interest is obtained. In this chapter, we describe how efficiently IMC works in DNA cloning in combination of experimental units such as (1) RE digestion, (2) PCR, (3) ligation reaction, (4) transformation and (5) gel electrophoresis.

(1) RE digestion

The restriction enzymes used in cloning technology are derived mainly from bacteria, and their recognition sequences are too short to accidentally occur in any long DNA molecule. Thus restriction enzymes can be used to analyze DNA from any source. The main reason why they are useful is that a given enzyme will always cut a given DNA molecule at the same sites.

There are two types of ends of DNA fragments cleaved by restriction enzymes, **sticky** or **blunt ends**. For example, restriction enzyme BamHI recognizes the site 5'-GGATCC-3' and cleave it as two DNA fragments with 5'-G-3' and 5'-GATCC-3' which carry sticky ends (**Figure 4.3a**). On the other hand, **blunt ends** are produced by cleavage by restriction enzyme, such as EcoRV (**Figure 4.3f**). Note that sticky ends enable the ends of the two fragments to base-paired correctly with each other. This ligation also reconstructs the original restriction enzyme recognition site, which allows DNA fragments to be easily inserted or removed. Information on restriction enzymes has been accumulated in REBASE (Roberts 2005).

IMC software has functions concerning DNA fragmentation by restriction enzymes as shown in **Table 4.1**. One is to be able to recognize both two types of ends (staggered and blunt ends). This is important that the consecutive ligation is performed by using such information. Once a DNA fragment is digested by a restriction enzyme in IMC, the end type of the fragment is described with the qualifier.

Table 4.1 Experiment functions provided by IMC (Part). The functions in yellow colored lines are described in this paper. The columns are function identification code(1st column), Function corresponds to experiment in vitro (2nd), experiment name (3rd), substrates name (4th), enzymes (5th), reaction products name (5th), IMC function name (6th), IMC operator (7th), sample in silico (8th, 9th).

ID	C	Experiments in vivo	before reaction	Enzymes	after reaction	IMC internal functions	IMC Operator	Sample 1	Sample 2
9	C	Apply DNA sample(s) into the current tube				READ DNA sample file(s) into the current directory and show the feature map of one of the samples	Apply DNA	DNA file(s) in the current directory	
10	C	Dispose DNA sample(s) from the current tube				Delete DNA sample file(s) from the current directory	Remove DNA	DNA file(s) in the current directory	
11	C	Apply Restriction Enzyme(s)	DNA	Restriction Enzymes	DNA	Find recognition sites on the DNA sequence	Site Recognition	DNA file in the current directory	
12	C					Divide DNA sequence at the site with specified cleavage style	Digestion	DNA	
13	C	Apply primers	DNA, Primers	DNA Polymerase	DNA	Find priming sites on the template DNA sequence	Priming	DNA	
14	C	Start PCR	DNA, Primers	DNA Polymerase		Copy template DNA sequence between the two primers	Amplify DNA	DNA	
15	C	Apply ligase	DNA(s)	Ligase		Ligate DNA	Ligation	DNA	DNA
16	C	Transform				Get information about the host cell	Transformation	DNA	Host Cell
17	C			Methylase		Apply condition of methylation	Methylation	DNA	Host Cell
18	C	Apply Kinase		Kinase		Attach phosphate group at 5' ends of DNA sequence	Phosphorylation	DNA	ATP
19	C	Apply phosphatase		Phosphatase		Remove phosphate group from 5' ends of the DNA sequence	Dephosphorylation	DNA	
20	C	Apply Nuclease		Mung Bean Nuclease		Remove the single stranded bases from the DNA sequence	Burting by Mung Bean Nuclease	DNA	
21	C	Apply T4 DNA Polymerase		T4 DNA Polymerase		Make the single stranded bases of 5' overhang to double strand, while that of 3' strand overhang removed	Burting by T4 DNA polymerase	DNA	
22	C	Apply DNA sample(s) into a well of a lane in the agarose gel	DNA	Ethidium Bromide, Dye		Read the DNA files in the current directory, and check their sizes, and draw the gel image	Electrophoresis	DNA	DNA(s)
23	C	Start Electrophoresis							
24	C	cDNA hybridization	DNA, cDNA(s)			Read cDNA sequence files one by one, and make homology search against the specified DNA sequence, if homologous, registerate as a mRNA		DNA	cDMA(s)
25	C	Finish experiment				Put time stamp on record			
26	I	Prepare T-Vector				Attach T-base	T-base	DNA with blunt ends	
27	O	N.A.				Show the reverse complementary strand	Reverse Complementary	DNA	
28	N.A.					Draw plasmid map of the current DNA sequence		DNA	
29	N.A.					Draw feature map of the current DNA sequence		DNA	
30	N.A.					Draw genome map of current DNA sequence		DNA	

Once, a DNA sample has been digested by restriction enzymes, an extra description is added or re-written so as to identify the exact shape of its end. When the end has a single strand or protruding bases (sticky end), the end shapes of a digested DNA are expressed as two integers with sign depending on the direction of protruding bases and with comma as the delimiter which separates 5' and 3' end digits. A DNA sequence with this description will be examined whether the ligation between two ends of any fragments is possible or not, when these fragments are specified as target molecules applied with ligase enzyme.

As mentioned above, the single stranded bases in the double stranded DNA can be recorded upon the information about each end shape. Along most of DNA sequence, a DNA molecule has double stranded shape so only one strand sequence is adequate to be recorded. In addition, one end with actually complementary to the recorded strand must be converted into complementary sequence before viewed. Therefore IMC keeps the description of the extra single stranded bases after digestion. The bases on the sticky end of the complementary strand to the current sequence, are immediately converted as the bases on current strand, therefore it is always enough to record only one strand sequence.

Digestion by a restriction enzyme sometimes occurs just on any feature of the DNA sequence. After digestion, the feature is divided into two separate fragments. Namely, the each fragment has one incomplete feature on it each other. IMC performs this operation precisely.

(2) PCR

Using PCR technique, a given nucleotide sequence can be selectively and rapidly replicated in large amounts from any DNA sample that contains it. PCR is iterative reactions consisting of three steps by starting with a double-stranded DNA; separation of two strands (**Step 1**), hybridization of two primers to complementary sequences in the two DNA strands (**Step 2**), and synthesis of DNA from the two primers (**Step 3**). In the case of N th iteration, DNA fragments specified by two primers are produced to 2^N times of them.

Like as digestion by a restriction enzyme, priming in PCR sometimes occurs just on any feature of the DNA sequence. After PCR, the feature on the PCR product becomes incomplete one with either side of the feature missing. As for the primers themselves, they are usually modified by insertion or substitution of a few bases. These modifications are inherited to the PCR product. IMC performs this operation precisely. Occasionally, different sizes of multiple DNA fragments are obtained in PCR because of a few base-mismatched priming to complementary DNA. These multiple DNA fragments can be viewed by 1D gel electrophoresis in IMC.

(3) Ligation

The enzyme DNA ligase reseals also the nicks in the DNA backbone that arise during DNA replication and DNA repair, and has become one of the most commonly used tools of recombinant DNA technology, as it allows to combine any two DNA

fragments. Isolated DNA fragments can be recombined in the test tube to produce DNA molecules.

IMC decides whether two DNA fragments are ligated or not. This is important for constructing a vector including DNA fragment of interest. The results of *in silico* digestion by restriction enzymes or *in silico* PCR performed by IMC, are recorded as to reconstruct the shape of each products. When some of these products are the targets of *in silico* ligation, a test is performed if the end shapes are matched to be ligated. If not, ligation does not occur. The results are easily verified by drawing the plasmid map or one dimensional gel electrophoresis.

The exact bonding site of ligation should not be located after the reaction. Namely, there is no trace of ligation on the DNA ligated in experiment *in vitro*. However, identification of the ligation site sometimes would be convenient because IMC users would like to draw the plasmid map with the insert sequence ballooned outside the map. In this case, IMC's answer is that it records both the DNA fragments with different sources, instead of recording the ligation site. It has been a custom to describe that the ligated DNA sequence with various inserts should be recorded with the feature key source. Self-ligation could be described in the same manner. A single DNA fragment would be ligated with both ends of its own if matched for ligation. This ligation produces one circular DNA molecule, and the ligation site would not be identified after reaction. In this case, it is difficult to hide the trace of ligation because the sequence itself is actually recorded as a string of characters with the start character and the last one.

IMC realize a circular DNA map by examining the definition line information, linear or circular of the international nucleotide databases. If the definition line is written as circular, IMC would draw one circular DNA feature map without identifying the start base or last base. That is, the map can be eternally scrolled to one direction or another without stop.

If a DNA fragment is removed by a phosphate group at its 5' end by enzyme phosphatases, this end could not be ligated with any end of other DNA any more after the reaction. I have introduced a new feature key dephosphorylation for this purpose. If the DNA fragment has the feature key dephosphorylation as the expression according to the international nucleotide databases, ligation to this DNA fragment does not occur.

Some features may be specially processed when the bases on the features are the target of RE digestion, PCR amplification and ligation. Taking CDS feature for an

example, when a CDS is RE digested or PCR amplified and only a upstream half of the CDS is ligated into a plasmid vector. This means that the CDS has lost its downstream nucleotides and they are replaced by the nucleotides of the vector. In this case, the CDS has lost former termination codon, therefore if the initiation codon is maintained, only thing to do is to locate first downstream termination codon on the frame. Accordingly, the location of the CDS and the last half of the nucleotides might be updated. In contrary, when the CDS has lost its upstream half of the nucleotide, it means that the CDS has lost the initiation codon and considering that no initiation codon may identified on the vector sequence of the upstream of the CDS, the next possible initiation codon should be located and if found the CDS length becomes shorter while if not found, the CDS annotation must be deleted.

(4) Other experiments

DNA can be introduced into bacteria by a mechanism called transformation. DNA fragment does not change in transformation. A DNA fragment transformed with a vector plasmid, could be methylated in certain sequences if the host is that of *Escherichia coli*. If methylated, the sequence might not be digested by some of the restriction enzymes which are affected not to digest their recognition sites on methylated DNA. IMC recognizes these affections by methylases in such cells. If some recognition sites are methylated and the restriction enzymes are affected by methylation, IMC does not digest at these sites.

4.3.4 Examples of consecutive routines in cloning experiments

IMC implements fundamental *in silico* experiments concerning DNA cloning, that is, RE digestion, PCR and ligation. I explain how IMC performs the consecutive routines by giving an example of TA cloning (Holton and Graham 1991, Marchuk et al. 1991, Mead et al. 1991). In TA cloning, a DNA fragment with sticky end of a deoxyriboadenosine obtained in PCR procedure is cloned into the linear plasmid called the T-plasmid-vector with a deoxyribothymine (dT) addition at the 3' end, which is the complementary to the dA of the PCR product.

This experiment is performed *in silico* with IMC, according to the procedure described below.

(1) Applying a sample of DNA: Apply a sample of DNA, namely, read a GenBank format file of a DNA sequence. This is carried out by clicking ***Read Sequence File*** button. Then, ***Read Sequence File(s) into Feature Map*** dialog window is popped up,

where one or more sequence files are selectable to be read and feature map of one of them is shown(**Figure 4.5(1)**).

- (2) **Specify the region to be amplified on the map:** After showing the region to be amplified by manipulating zoom or scroll buttons, dragging mouse across the region on the feature map results in change of the background color in red(**Figure 4.5(2)**).
- (3) **Design of optimal primer sets:** Design optimal primer sets to amplify the declared region of the DNA sequence. On the colored region, clicking of the right button of mouse pops up a menu. On the top of the menu, there is a submenu *Design PCR Primer*, then clicking of the submenu pops up *PCR Primer Design* window where a set of parameters for designing PCR primers are listed and can be changed. PCR runs immediately after clicking *set* button at the bottom of the window(**Figure 4.5(3)**).
- (4) **PCR:** Let PCR be performed after selecting one of the just designed primer sets. This is initiated by starting PCR with clicking *PCR* button after selection of the primer, then *Priming site search* window is popped up where one set of primers can be selected with allowance of mismatches. If one base mismatch is allowed, select the radio button of *1 base mismatch*, then clicking of *Reaction* button starts PCR. All or specified portion of the DNA sample sequence are searched for priming sites, and a list of PCR products is shown (**Figure 4.5(4)**).
- (5) **Selection of PCR:** After selection of PCR products from the list of PCR products, a click on *Reaction* button starts PCR. Perform one dimensional gel electrophoresis for all the above PCR products. One clear band is shown on the lane with many dark bands resulting from mismatched priming. On the bottom of *Priming site* window, there is a button for one dimensional gel electrophoresis which is used to start gel electrophoresis about all or selected PCR products (**Figure 4.5(5)**).
- (6) **Registration of PCR products in DNA sequence file:** Any of products is registered in a new GenBank or EMBL format file. The shape of the both ends of a PCR product can be changed dependent on the kind of PCR enzymes. IMC lets users select type of shapes of one adenine base sticky or blunting end. The end shapes of the PCR product can be verified by scrolling the map until the both ends of DNA sequence are shown (this could be done one by one), after making the PCR product DNA as the current one. Each of the PCR products is recorded as a single entry of GenBank or EMBL format file. The contents of the file can be verified by a click on the *GenBank/EMBL Viewer* button (**Figure 4.5(6)**).

- (7) **Selection of Restriction enzyme:** Let us take a vector sequence digested by the restriction enzyme EcoRV which generates blunting ends. This is done by clicking **RE Recognition** button after the vector sequence is selected as current one. Then, **Enzyme Selection Window** is popped up and checking on EcoRV and clicking of **Show Recognition Site** button makes the **Recognition site** window to pop up on which a list of the recognition sites by the enzyme is shown. Clicking **Digestion** button starts the digestion process (**Figure 4.5(7)**).
- (8) **Addition of thymine to 3' end:** Let one thymine be added at 3' end of the above vector. This is a virtual reaction for IMC user's convenience and could produce a T-vector for TA-cloning. This is carried out by clicking **Add T-base at both 3'ends** button. The end shapes of the digested product can be examined by scrolling the map until the both ends of sequence are shown, after making one of the digestion fragments as the current one (**Figure 4.5(8)**).
- (9) **Ligation:** Let one of the PCR products and a T-vector be ligated. This reaction generates a circular plasmid vector with the PCR product inserted. This is carried out by clicking **Ligation** button. On the pop up window for ligation, two DNA sequences can be specified. After setting the T-Vector sequence as the first fragment and setting the PCR product as the second, clicking of **Ligation** button starts ligation and produces two circular DNA after reaction (**Figure 4.5(9)**).
- (10) **Plasmid map:** Before drawing of the plasmid map of the vector, list of DNA sources which consist of the circular DNA, is shown for the selection of inserted DNA. The inserted DNA is clearly identified on the plasmid map. This is done by simply clicking the **Plasmid Map** button. On the popup, a list of sources on the ligation product is shown. After selecting one for the source of the inserted PCR product, clicking of **Set** button leads to draw the plasmid map with the insert of PCR product. Map sizes can be changed by modifying the parameters (**Figure 4.5(10)**).



Figure 4.5 IMC Operation in consecutive routines. (1) After importing a GenBank format file with annotation, its feature map is drawn. (2) Dragging mouse over a region on the feature map to specify the region to be amplified by PCR. (3) By clicking the mouse right button, a menu is popped up. Selection of 'primer design' sub menu will start the PCR primer designing. A list of primer sets are shown in a dialog and PCR will be directly performed from this dialog, too. (4) After starting PCR, a confirmation message is appeared. (5) A list of the PCR products are shown and can be save as a GenBank/EMBL forma file. (6) At both ends of the PCR product, additions of adenines are optionally selected to be ligated with T vector. (7) A circular vector is imported and digested by a restriction enzyme. (8) Attachment of a single strand thymine base at the both ends of the linear vector can be performed to produce a T vector. (9) Ligation between T vector and the PCR product is performed. This reaction produces two ligation products, one has the reversely inserted PCR product. (10) Drawing of a plasmid map is accomplished by clicking one of buttons.

4.4 Discussion

4.4.1 Advantages

There are three merits in *in silico* experiments in view of (1) planning molecular biological experiments, (2) usage as lab notebook, (3) educational tool to learn molecular biological experiments.

(1) Clarification of procedure by predicting the results of actual experiments:

A large cost is required in *in vitro* experiments handling DNA samples because reagents or disposable materials such as plastic micro titer plates, micro tubes and pipette tips are consumed in large amount. Genome-wide projects require much higher rates in resource consumption. *in silico* experiments can provide low cost alternatives for verifying the effectiveness of *in vitro* experiments prior to starting them actually. For example, in a molecular cloning experiment, there is a case that insert DNA is actually inserted into vector in reverse way. To detect this event, an optimal selection of restriction enzyme is necessary, and leads to a good method to determine the direction of insert using 1D gel electrophoresis.

(2) Usage as lab notebook:

If there is a series of experiments that researchers are planning to perform, there are always some modifications against the standard protocols, so these procedures should be recorded just at the time the plan is ready. The history of these records could be utilized as a lab note, which is also important for researchers to verify the finished experiments. Recording such as comprehensive drawing of a plasmid map with DNA insert with features on it, will increase efficiency of the experiments.

(3) Educational tool to learn molecular biological experiments:

It might be difficult for beginners to understand what are going on in the micro tube in molecular biology experiments *in vitro* because molecules in the tube are only visible with indirect means like gel electrophoresis, *in silico* experiments, which can show any molecule at any time, help them to understand the micro phenomena and process of experiments.

4.4.2 Framework construction for *in silico* experiments

Performing *in silico* cloning requires recording of the end shapes of digested products by restriction enzymes or amplified products by PCR. For this purpose, I introduce a new feature key *endtype* and its qualifier *endtype*, and incorporate them into GenBank/EMBL database annotation convention. Some features on a DNA sequence might be truncated by PCR or digestion by restriction enzymes, therefore the annotations on the truncated features should also be modified. The ambivalent nature of DNA also requires occasional switching to the interested strand from one to another. In addition, I redefine information about the RE recognition sequences, end shapes after digestion and affection of methylation at the sites, to a new feature key and qualifiers. According to these definitions or data descriptions, I have developed a software for *in silico* experiments, and perform a few of typical molecular cloning experiments on computer, and verified that this approach would be effective as a recording tool of a series of experiments as a lab notebook, training tools for beginners to molecular biology, prior simulating tool for time or cost consuming experiments.

In molecular biology experiments, it is important how to describe the functionalities or activities of enzymes and how to use such description. According to the requirements of an *in silico* experiment, one data entry to one enzyme seems to be the best way. Descriptions on the enzymatic functional sites, are still poor in case of most of protein database entries. Therefore, in this stage, we assume that enzyme proteins act as only catalyst instead of multi-functional protein which has residue-specific activities around its amino acid sequence.

Chapter 5

A viewer for tiling microarray data

5.1 Introduction

A DNA microarray is defined as a miniature plate made of glass, plastic or silicon wafer attached with densely planted numerous DNA probes (Fodor et al. 1993, Chee et al. 1996, DeRisi et al. 1996). The probes are allocated on the plate with matrix shape, consist of single-stranded cDNA or oligonucleotides and are hybridized with complementary single-stranded DNA or RNA samples. If the probes are uniquely designed among the genome or the transcriptome, the DNA microarray can be used as a high throughput expressed molecular detector which measures numerous differently expressed genes in a single experiment. Since all the probes are derived from cDNA or complementary to genic regions of the genome, these kinds of microarray are called gene level expression microarrays.

A tiling microarray (Kapranov et al. 2002, Cawley S. et al. 2004, Kampa et al. 2004) is defined as high density DNA microarray whose probes cover a whole genome without gaps, namely the microarray probes are arranged to cover the genome entirely such as actual tiles which cover whole wall of a building as shown in **Figure 5.1**. The probes can be overlapped with each other, the extreme type of the tiling microarray is such that one probe sequence is only one base shifting apart from the previous probe sequence. Thus, the total number of probes which cover the entire genome, amounts to the same number of nucleotides in the genome. A tiling microarray provides an extremely fine measurement tool for transcriptome and genome analysis, such as novel gene discovery, gene expression, alternative splicing, binding of transcription regulators, chromatin immunoprecipitation-chip, ChIP-chip (Ren 2000, Iyer 2001), DNA methylation, polymorphism discovery and genotyping, comparative genome hybridization, CGH, and re-sequencing (Mockler & Ecker 2005). In contrast to a gene level microarray which provides gene level analysis after obtaining means of 11 to 20 values to enforce the robustness of measurement, a tiling microarray is much dependent upon the individual probe intensities and an intensity measured on a probe is required to be more significant and used independently from neighboring probes. Therefore, robust estimators for probe level expression intensities are much required. Thus, the tiling microarray is also called a probe level expression microarray.

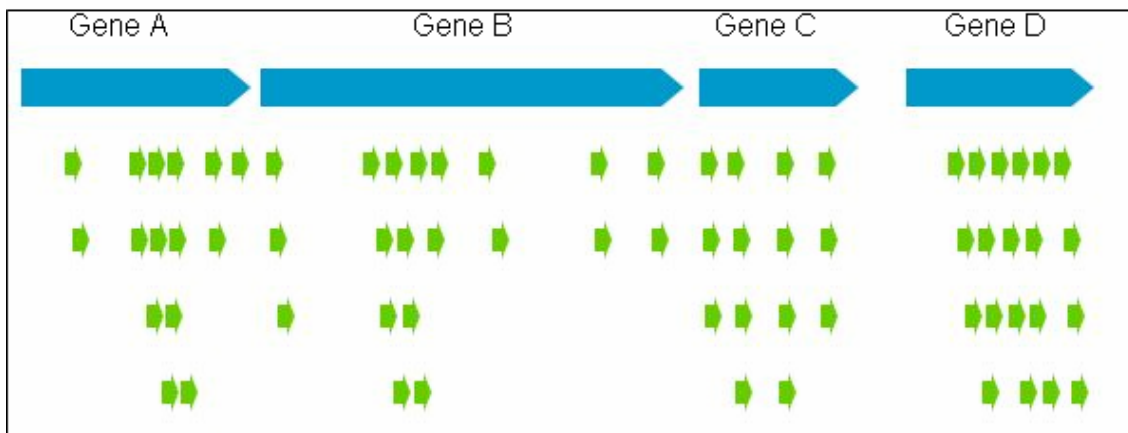


Figure 5.1 Schematic drawing of a quasi-tiling microarray (part), since the probes do not cover the genome entirely. Arrows painted in blue denote the CDS features. Smaller arrows in pale green denote probes of the tiling microarray.

Tiling microarrays are commercially available from *Affymetrix, inc.* or *Agilent Technologies, inc.* and other manufacturers. As for the *Affymetrix* microarray, its probes are made of 25-mer oligonucleotides while *Agilent* uses 60 or 70-mer. Even for a microbial tiling microarray, the number of the probes amounts to sub millions order, if the genome size is 5Mbp and the probes cover the entire genome even without any overlap.

Quasi-tiling microarrays with unevenly distributed probes with certain gaps between neighboring probes, are also designed (Mockler and Ecker 2005). Exon arrays are one of the typical examples. On the contrary, tiling microarrays whose probes on the intergenic regions are densely allocated while probes on the genic regions are sparsely planted (**Figure 5.2**), can be also designed. This kind of tiling microarray is used for detection of promoters or unknown smaller genes on the intergenic regions.

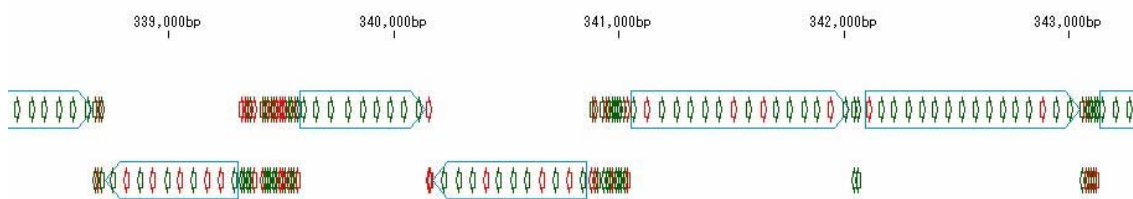


Figure 5.2: Probes on the *Bacillus subtilis* whole genome map. Narrow arrows with wings denote features of tiling microarray probes, while wide arrows without wings denote CDSs on the genome.

Currently, there are few software tools available to view the entire expression pattern on a whole genome using a tiling microarray. Without such tools, the tiling microarray analysis is not adequately performed. Therefore a software tool to visualize the tiling microarray data is urgently required. Even for a quasi-tiling microarray with unevenly distributed probes, the number of probes amounts to 120-160 thousands, whereas the number of genes in *Escherichia. coli* and *B. subtilis* amounts to about 4,200 and 4,100, respectively. Thus, such a software tool is required to handle these huge amounts of data without frustration. Unlike the gene level microarray software tools, a software tool for tiling microarray data analysis is required to be implemented with site-specific viewing facilities because the results are closely related with the position on the genome sequence and each probe is required to be more significance than a gene level analysis. Therefore, the existing annotation about genes, promoters and so forth, should be displayed in parallel with intensity indicators on each probe.

An efficient and effective structure for the above-mentioned tiling microarray data should be introduced and smooth viewing and editing should be also accomplished. In addition, when handling microarray data, certain problems are commonly encountered, including the existence of background and cross-hybridization, array

manufacturing bias, and so forth. Therefore, software tools are also required to solve these problems altogether.

If uniformly distributed similar sized DNA fragments which are derived from random fragmentation of the genome, are obtained, their hybridization to the microarray probes can be used to normalize the corresponding expression level of mRNA or other hybridizing molecules, due to the assumption of the hybridization intensity which is interpreted as hybridization efficiency of same oligonucleotides. Therefore, this concept requests that such a software tool provides arithmetic operation between RNA expression microarrays and genome fragments hybridization microarrays. Furthermore, as the still low level of microarray reproducibility leads to repetitive sample measurements or replicated experiments, average calculation between microarrays, is also necessary to be implemented on the software tool.

I developed a software system, named *in silico MolecularCloning Array Edition* (hereafter, referred to as IMCAE), to provide researchers with a visualization tool of the tiling microarray data with data correction functions as well as gene level analysis such as clustering and ranking of the expressed genes. In IMCAE, since the probes of the tiling microarray are assigned as features on a GenBank/EMBL format file, they are handled in a same way with other features such as CDSs, mRNAs, rRNAs and promoters. In addition, expression intensity values are also incorporated as the qualifiers to the corresponding probe features. The design requires only one file to record probes, expression data and features on a genome. Thus, simple operations and visualization of the tiling array data, is accomplished. This design increased the portability of the complicated microarray data between the researchers and their collaborators.

5.2 Methods and Algorithms

5.2.1 Mapping of tiling microarray data on a annotated genome sequence

(1) Structure of the probes and expression data

For describing an *Affymetrix* custom tiling microarray data, three types of data file, probe sequence file, channel definition file and microarray expression data files, are necessary. The channel or chip definition format file (CDF) is a conversion table between the array matrix position of each probe and corresponding probe sequence. The expression data file is named as CEL file, derived from microarray with one-to-one correspondence as the hybridization intensity on each probe position on the microarray. A probe file contains all the probe sequences and atom numbers and probe unit numbers. The probe sequences file contains these of perfect match only. And the atom number and the unit number are used to link a probe sequence and its hybridization intensity on microarrays. By using the probe sequence file and the CDF file, probe sequences are linked to its microarray position of the CEL files. The CEL files have only microarray positions for corresponding probe intensities. Thus, these expression intensities are connected to probe sequence to show where in the genome the expression is detected. The schematic diagram on the relationship between these files and the formats are described in **Figure 5.3**.

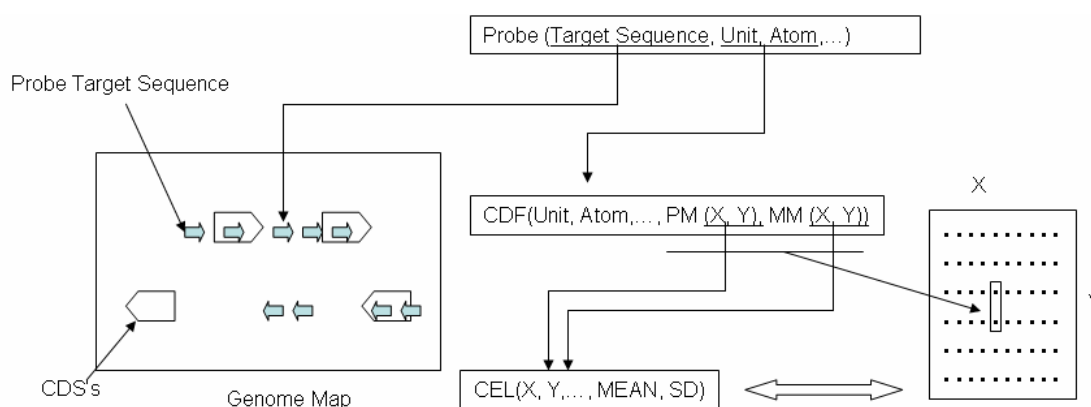


Figure 5.3 Diagram of relationship between Probe, CDF, CEL and Genome Map. The CDF file acts as a link table between probe sequences and their corresponding microarray positions.

There are detailed data formats of a probe and an expression microarray as shown in **Figure 5.4**. The probe file consists of the target sequences of each probe.

(a)

EXPOS	LOCAL_TARGET	NAME	ID	UNIT	ATOM
34	tatccacagaggttatcgacaacat	BsubGenome-1_x	!	1100	0
64	cattaccaacccctgtggacaaggt	BsubGenome-1_x	!	1100	1
97	caggttggtccgctttgtggataaga	BsubGenome-1_x	!	1100	2
126	gacaaccattgcaagctctcgttta	BsubGenome-1_x	!	1100	3
162	tatttggttttaactcttgattac	BsubGenome-1_x	!	1100	4
197	tttcctctttatccacaagtggtg	BsubGenome-1_x	!	1100	5
228	tgtggattgattccacacagcttgt	BsubGenome-1_x	!	1100	6
254	tagaaggttgccacaagttgtgaa	BsubGenome-1_x	!	1100	7
283	gtcgaagctatttactata	BsubGenome-1_x	!	1100	8
318	tcaacattaatgtgtacgaatggt	BsubGenome-1_x	!	1100	9
354	gctcttttttgtgtctataacag	BsubGenome-1_x	!	1100	10
383	agacgccatttttaagaaaaggag	BsubGenome-1_x	!	1100	11
411	cgtgccggaagatggaatatatt	BsubGenome-1_x	!	1100	12
443	tggaaccaagcccttgctcaaatcg	BsubGenome-1_x	!	1100	13
477	tgagcaaacaggagtttgagacttg	BsubGenome-1_x	!	1100	14

(b)

```
[CDF]
Version=GC3.0

[Chip]
Name=BSUBTILEb530197N
Rows=754
Cols=754
NumberOfUnits=8488
MaxUnit=9512
NumQCUnits=9
ChipReference=

[QC1]
Type=15
NumberCells=1084
CellHeader=X Y PROBE PLEN ATOM INDEX
Cell1=13 0 N 25 0 13
Cell2=17 0 N 25 0 17
Cell3=21 0 N 25 0 21
Cell4=25 0 N 25 0 25
Cell5=29 0 N 25 0 29
Cell6=33 0 N 25 0 33
```


number (Unit=3127) and 'ATOM's are used to link to the probe file. Namely, if a probe is given by Unit=3127, Atom=2, it is linked to the line written in red. Actually there are two of the lines which link to the probe. One is for the perfect match probe, and the other is for one-base mismatch probe. Mismatch probe is clearly identified with the same three bases such as TTT while the perfect match probe line is identified by the difference of the central base of the codon. 'CellHeaders' (X, Y) can be used to link to the CEL files which are the actual records of array expression intensities. X and Y mean the coordinates of the probe location on the array. (d) This is the contents of a CEL file in a text format produced by GCOS. In the [Intensity] field, X and Y of 'CellHeader' are used to locate its probe position. A perfect match probe's expression intensity is shown here in blue characters, while its corresponding mismatch probe's expression intensity is shown in red characters.

To accommodate millions order of these numerical data efficiently, a new data structure for the tiling microarray data is introduced. Prior to this, high throughput data handling method is already established on the software tool, *in silico MolecularCloning* (IMC: see **Chapter 3**). IMC handles almost all the biological annotation as features and qualifiers on a DNA sequence. In the same manner, when handling the tiling microarray data, most of information about the microarray data can be mapped as features on the genomic DNA sequence. Currently there are two feature keys, 'TilingInfo' and 'TilingArray', to describe tiling microarray data. General information about the microarray is also recorded as 'TilingInfo' feature. A 'TilingInfo' feature consists of data about the probe file, the CDF file and CEL files. Each file location on the PC is recorded on 'probe_valuen' qualifiers, where n denote the n th CEL file to be stored on the file. The probes on a tiling microarray are defined as 'TilingArray' features on the annotated genome sequence with GenBank or EMBL format. Every 'TilingArray' feature has a couple of qualifiers to describe the probe name, the probe position, and the means and SDs of measured intensities from PM and MM probes. Therefore, the computation between stored microarray data is immediately performed without referring any other file.

(2) Mapping of the probes on the genome sequence

Since the tiling microarray probes entirely cover a genome, it is always required to draw a genome map with probes, similar to the way of drawing map with array CGH experiments. Therefore, I describe how to draw such genome map with tiling microarray probe intensities. First, probe sequences are searched against the

whole genome nucleotide sequence. A perfect matched search is required to be performed for probe mapping on the genome. Most of the probe sequences are unique among the genome sequence, while some of the probe sequences, such as probes on rRNAs or tRNAs and other highly homologous elements among the genome, may have multiple perfect match sites on the genome sequence. This can be interpreted as one single probe represents two or more sites on the genome. All the different sites of a multiply allocated probe sequence are registered as single probes.

This kind of mapping process is usually performed only once and each probe is registered as a feature on the genome sequence. Therefore, the process time to complete the mapping is not critical and the probe file is not necessary to be referred later. At the same time, the two-dimensional coordinates of the probes on the array are also recorded as the qualifiers to the corresponding probe feature. Afterwards, by only using the coordinates, corresponding intensity of samples hybridized to the probe is directly obtained. Therefore, neither CDF file nor CEL files are necessary after the creation of probe features. Thus, four kinds of data files are used to describe the results from *Affymetrix* oligonucleotide microarrays. **Figure 5.3** shows the diagram to explain the relationship between the four data formats. The target sequence is a reverse complementary sequence to the actual probe sequence.

The other two parameters named 'unit' and 'atom' are used to link to the corresponding CDF file which has the planted positions of all the probes on the microarray. The probe positions, including these of perfect match probes and mismatch probes, are designated by the coordinate X and Y . The mismatch probe's location is always just below (plus 1 row) the location of its corresponding perfect match probe. As for all the probes on the microarray, a mean value and a standard deviation (SD) value of hybridization intensity are accompanied with its probe coordinates X and Y on a CEL file.

5.2.2 Correction of tiling microarray data

As well as gene level microarrays, the tiling microarray data has also statistically interesting but problematic characteristics. To conquer these problems, a variety of methods or algorithms should be implemented.

(1) Evaluation and correction of array data reliability using SD

Affymetrix oligonucleotide microarrays consist of huge number of probes which hybridize to complementary RNA or DNA fragments. These probes on the

microarray are usually measured by an image scanner of typically 16-bit resolution and the image assigned for one probe, can be pixelated up to about 100 individual pixels (Affymetrix 2006). One pixel is given one single value of intensity, from zero to 65,535 if the bit is used at its full range. After analyzed by the image processing software, outliers among the measured intensities are screened out and therefore only the mean and the standard deviation (SD) of all the pixels except the outliers, are provided as the primary data. To remove the outliers with large SD value of probe intensity, a straight line through (0, 0) with slope A is drawn. All the points dropped below the line are classified as outliers. A common graphical notation of error bar to describe the reliability for each intensity, is also implemented.

(2) Background subtraction

The background measured with microarray is regarded as fluorescence intensity resulting from various factors, including non-specific binding of labeled target, stain, and other fluorescent materials contained in the sample solution (Liu 2002). The background also varies by regions on a microarray. Average of lowest 2% intensity values are used as the background intensity in the *Affymetrix* software, *MAS 5.0*. The background subtraction from detected intensity of probes, is performed before other data correction methods. This is global correction of background intensity. However, occasionally, the surface of microarray is unevenly stained. In such a case, local estimation of the background intensity is required. Currently, I used only a global background subtraction method for avoiding complicated and multiple correction to the raw data.

(3) Normalization methods of microarray data

It is reported that the efficiency of hybridization is different from one microarray by microarray. In addition, when manufactured, the microarray probes are not evenly planted and when hybridized, the fluid containing RNA samples are not equally extended on the chip surface (Holloway 2002, Liu 2002). These facts lead to some systematic biases existence in microarray experiments. Therefore, some normalization methods are necessary when handling microarray data.

When comparing two microarrays, most popular method to normalize data is to sum up globally over the probe intensities on the microarray surface, the entire genomic region, namely to take summation of all the intensities, then the each expression intensity is divided by the sum. In this calculation, the values of normalized intensities tend to be very small numbers, therefore the use of double precision type of data format

is required to avoid the loss of significant digits. A local normalization is also considered. Similar calculation of global normalization is applied in a given interval between i th and $(i+n)$ th intensities.

(4) Confirmation of intensity distribution

The distribution of raw probe intensities of microarrays, is in advance examined to decide the strategy how to handle the microarray expression data. In this process, screening of outlier intensities is also required. The common method to confirm the distribution pattern of probe intensities, is plotting of every expression intensities on a scattered plot. Because the dynamic range of intensities is wide and, in most case, ratios between two different arrays are concerned, logarithm scales are introduced to both axes of the plot graph. The simplest plot is that of two microarrays, x for one axis and y for another.

In some of microarray data, systematic biases appear. It is reported that the \log_2 (ratio) values can have a systematic dependence on intensity, which commonly appears as a deviation from zero for low-intensity regions. Locally weighted linear regression (LOWESS) analysis has been proposed as a normalization method that can remove such intensity-dependent effects in the \log_2 (ratio) values (Quackenbush 2002).

I implemented the function of drawing the distribution plots of probe intensities, with a robust normalization of trimmed means.

(5) Reduction of the effects from non-specific binding or cross-hybridization

In *Affymetrix* oligonucleotide microarray, two types of probes were introduced; one type has the perfect match (PM) complementary DNA sequence to the genome, while the other has one-base mismatch (MM) complementary DNA sequence to the genome (Lipshutz et al. 1999). These probes are allocated on the microarray in the neighboring rows each other. The differences of intensities between perfect match and mismatch probe are used to reduce the effects of non-specific binding or cross-hybridization. The perfect match expression intensity is called PM, the mismatch expression intensity MM and the ideal expression intensity (IM) that might be calculated by using PM and MM as IM. IM (Ideal Match) is represented by PM minus MM if PM is larger than MM, nevertheless if PM is smaller than MM, some consideration must be taken. On the gene level microarray analysis, the PM-MM set is used for determine the probability of each probe expression detection is positive or negative.

One solution is IM should be zero if $PM < MM$. Another solution is IM would be neglected if $PM < MM$. Other solution is IM would be the absolute value of PM minus MM, namely $IM = |PM - MM|$. This is a controversial problem so the software can handle either way. *Affymetrix* adopts more complicated calculation to derive IM. An example of PM, MM, and IM profiles are shown later in **Figure 5.7**.

The usage and interpretation of PM and MM are still controversial, therefore I implemented functions to handle PM and MM independently each other. Namely, PM and MM are called, exhibited and computed separately.

(6) Trimmed mean estimator

Single measurement of expression intensity for the target DNA or RNA is not robust against background noise and cross-hybridization. To reduce the effects, redundantly arranged probes have been introduced. In the case of a tiling microarray, one problem is that a probe intensity has much more significance than that of a gene level microarray. Therefore, if it is used to represent as a single measurement, the robustness may be lost against noise.

To increase the robustness of tiling microarray data, a sliding window approach and trimmed mean is usually effective. The sliding window approach method is widely used in the sequence analysis of nucleotides or amino acids. In similar way, the single intensity on a probe is replaced with the average of n intensities within the window size interval of the genome. The trimmed mean is a robust estimator in statistical analysis. When calculating trimmed mean within the window, highest m values and lowest n values are omitted from calculation, where m and n are integers independently given each other. The outliers are usually removed before obtaining the trimmed means.

(7) Detection of transcript initiation sites using intensities on CDS regions

Unevenly distributed quasi-tiling microarray probes between CDSs and intergenic region of the genomes are designed (see **Figure 5.2**). In this case, the probes are allocated densely in the intergenic regions while these in the CDS regions are more sparsely allocated. The purpose of this imbalance is coming from the ideas to reduce the number of probes totally, and to investigate local structures and functions on the intergenic regions. The main targets are to obtain the exact site of transcription initiation and to find out possibilities of smaller CDSs between identified larger CDSs. As the expression values of the probes on the CDS regions are regarded as the baseline of the transcript, they can be used to prove the expression intensities of the intergenic probes

valid or not. If there is no probe on the CDS regions, it is difficult to prove the real expression level of the CDS.

5.2.3 Comparison between arrays

Most microarray experiments provide only relationship between related samples and the finding of differently expressed regions or genes are the simplest approach by using the relation between the two expression values.

(1) Scattered plot and R-I plot

The dynamic range of the array expression value is mostly wide to plot the intensities in a single real number scale. For such a wide dynamic range data, logarithm conversion is effective. In addition, in the two-channel microarray, the intensities are usually presented as ratios. The resulting reciprocal value is easy to understand when it is shown in logarithm scale. In the case of the RNA expression microarrays using a typical image scanner, the dynamic range is the order of 10^5 - 10^6 . This is also useful to detect systematic biases for drawing an R-I plot, (it is also referred to as M-A plot). An R-I plot is defined as the plot graph for each intensity of two arrays where X axis is presented by $X=\log_2(\text{ratio})$ and Y axis is presented by $Y=\log_{10}(\text{intensity})$. This plot indicates intensity-dependent bias of measured data.

(2) Comparative operators between different arrays

Due to the low confidence level of microarray expression intensity accuracy, repetitive measurements, or replication experiments, on the same sample are often performed in the actual microarray experiments. For this purpose, some arithmetic operators between raw or corrected intensities are implemented to calculate averages between the repeated measurements. In addition, genomic DNA fragments of same size are applied to the microarray to measure the sequence-dependent efficiency of probe hybridization. This requires for the software tool to be implemented with arithmetic operators between RNA hybridization or other expression and genomic DNA fragment hybridization. Five of such operators of additive, subtractive, multiplier and divider and mean operator are required to satisfy the minimum functions. These operators directly take the expression file (*CEL* file) as their arguments. Therefore, simple operation of A (*CEL*) / B (*CEL*) gives numerous multiple divider operations between each component of A and B .

The genomic fragments are regarded as evenly distributed, however each hybridization efficiency is different according to the GC content of the probe DNA sequence. For example, if a probe sequence is AT-rich one, it may be more likely to be hybridized to the corresponding genomic DNA fragment. The hybridization efficiency may be theoretically calculated however it is also useful to obtain the experimental data from the genomic fragments hybridization. In the case of RNA expression data normalization, division operator would be applied as follows (Eq. 5.1).

$$I_{c_i} = I_{m_i} / I_{g_i}, \quad (5.1)$$

where I_{c_i} is defined as i th calibrated intensity of an array, I_{m_i} as i th intensity of mRNA and I_{g_i} as i th intensity of genomic fragment. The reason why the division operator is used is that the genomic fragments hybridize to almost all of the probes of the tiling array. The proof of the validity of division in this case should be done with the other comparative experiments on the efficiency of hybridization by sequence.

5.2.4 Gene level analysis

(1) Gene level expression

A gene level microarray consists of approximately 11 to 20 probe pairs on each gene of a genome. The expression level of gene is estimated using all or selected intensities of the probe pairs. After clustering of co-expressed genes together, a gene expression matrix with a gradient color scale, is commonly presented.

As for the tiling microarray, probes pairs are further densely allocated on each gene. The major difference between a gene level microarray and tiling microarray is that tiling microarray is regarded as probe level microarray. The probe intensities of tiling microarray are much significant than those of a gene level microarray. In contrast, it is rather easy to interpret tiling microarray data into gene level one. The difference is that of number of probe pairs on each gene. Then, gene level expression is easily presented by the tiling microarray, too. The co-expressed genes derived from the tiling microarray, is also presented as a gene expression matrix.

(2) Trimmed mean estimator

The mean calculation after removing m and n values from the largest m intensities and smallest n intensities from all the measured intensities mapped on a gene, resulting from the order statistics, is defined as trimmed mean. Trimmed mean is

regarded as a robust estimator of gene expression intensity.

(3) Ranking and clustering of gene level expression using tiling microarray

By using above trimmed means of intensities on CDS, gene level expression analysis is performed. A ranking of genes by trimmed mean is easily obtained by sorting a set of microarrays. Gene clustering is also possible using the data. Various algorithms for clustering genes by similar expression pattern were reported, such as hierarchical algorithms, self-organized map (SOM) (Kohonen 1984), support vector machine (SVM) and others. Without prior knowledge about the gene set, unsupervised clustering is usually used. Among the hierarchical methods, UPGMA (Un-weighted Pair Group Method using Arithmetic Averages) (Sokal and Michener 1958) and Neighbor-joining method (Saitou & Nei 1987) are most popular. I implemented an algorithm to clustering gene level data by using UPGMA.

5.3 Results: Implementation of a software tool for the tiling microarray

I developed a software tool, named *in silico* Molecular Cloning Array Edition (IMCAE), for analyzing, viewing, and editing of the tiling microarray data. IMCAE is written in Java application language and compatible with Windows^{XP} and Mac OS X. The software is implemented with several unique features, especially its probe and microarray data recording method as described in the **Section 5.2**.

5.3.1 Mapping of probes and expression data

(1) Probe and microarray data importing

In IMCAE, a probe file with its corresponding CDF file, is imported after reading the GenBank/EMBL annotated sequence of the genome. This procedure requires several hours on a typical PC to complete the importing of the probes and mapping them on the genome sequences. Then, currently, up to 10 expression microarray data files can be imported and their intensity data is recorded as the qualifier of the probe features.

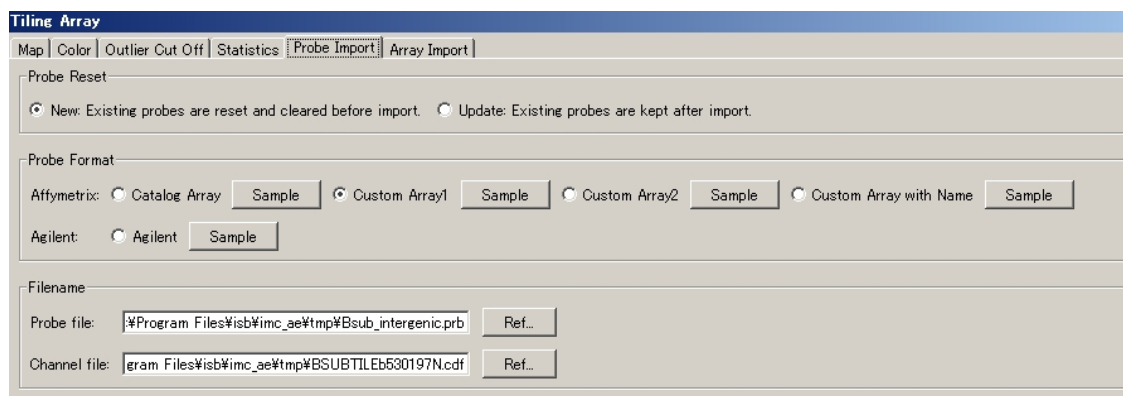


Figure 5.5 Parameter setting dialog for importing a probe and a CDF file. When ‘Update’ is selected, the probe mapping on the genome is additive without affecting the probes previously mapped. Currently, five types of tiling microarray is supported, four of *Affymetrix* microarrays and one for *Agilent* microarrays. The probe file and its CDF file must be specified simultaneously.

Multiple files of probes and CDF are also imported and mapped to a same genome using update function of the importing procedure. Before importing a new probe file, IMCAE usually clears the existing probe features on the GenBank/EMBL

format file (**Figure 5.5**). However, if the update function is selected, this initial clearance of probe features is not performed. In the current version of IMCAE, up to sub-millions of probes can be imported and mapped in a single genome sequence.

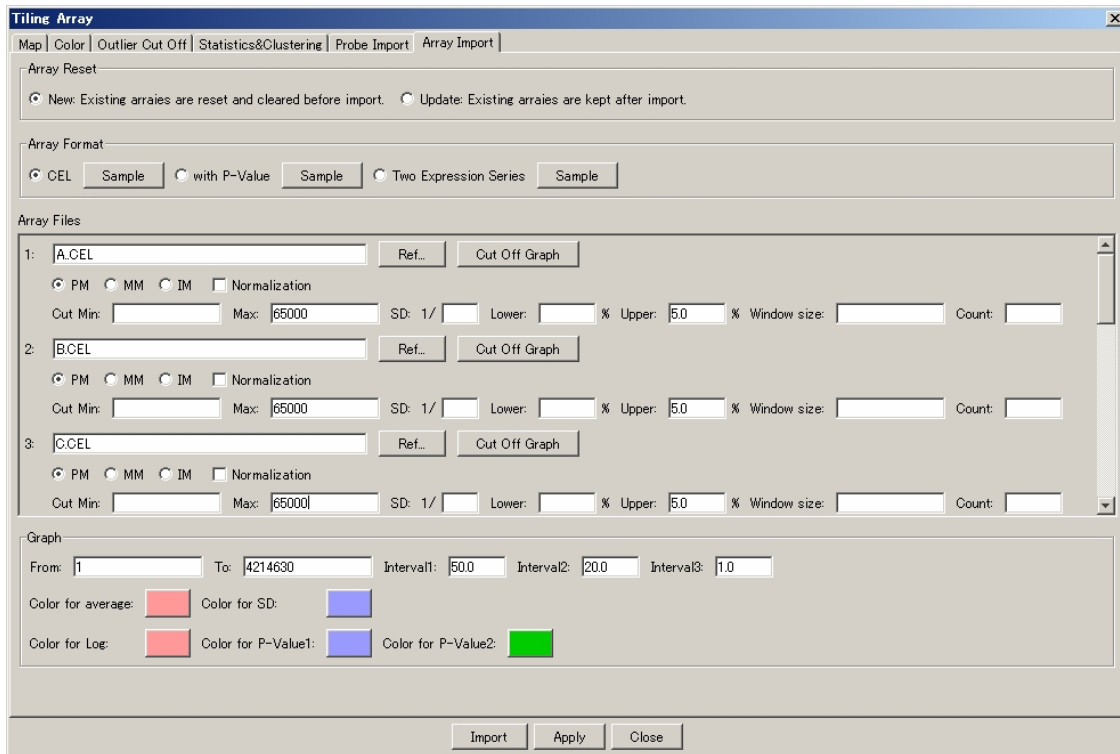


Figure 5.6 The CEL files importing window. Selection of importing methods, initial or additive import, is implemented. Two channel probe expression table format files can be also imported when the ‘Two Expression Series’ button is ON. The intensities are selectable from PM, MM or PM minus MM. ‘Cut Min:’ or ‘Cut Max:’ is used to cut off the intensities with lower than ‘Cut Min:’ or higher than ‘Cut Max’. ‘SD 1/’ is used to define the inclination of the cut off line. The probes with SD value larger than this value are cut off. Cut off is also performed on the rank of probe intensities, lower m % or larger n % of probe intensities are cut off.

After completion of probe mapping on the genome, up to 10 CEL files can be specified and imported. The CEL file locations are recorded as a description of feature key ‘TilingInfo’ on the GenBank/EMBL file. The probe intensities recorded on a CEL file are transferred to the exact corresponding probe features and also recorded as the qualifier. Namely, each probe feature entry, ‘TilingArray’, on a Genbank/EMBL format file has all the intensities of PM and MM probes.

```

Genbank file viewer
TilingArray    complement(371..395)
                /probe_name=int_BG10065_rev
                /probe_x_y=274_217<>275_218
                /probe_value=0,291.0,44.0,118.0,18.0,0,75.0,22.7,72.5,14.3
                ,0,82.0,17.0,77.0,18.2,0,118.0,20.1,122.0,17.6,0,469.0,83.
                2,82.0,12.2,0,1682.0,147.5,300.0,53.5,0,230.0,28.6,178.0,3
                6.3,0,176.0,32.5,197.0,28.5,0,107.0,23.2,139.0,25.5,0,156.
                0,37.5,146.3,28.1
                /update=060209163017
gene           410..1750
                /gene="dnaA"
                /locus_tag="BSU00010"
CDS           410..1750
                /function="initiation of chromosome replication (DNA synthe
                sis)"
                /note="dnaA; alternate gene name: dnaH, dnaJ, dnaK"
                /codon_start=1
                /transl_table=11
                /protein_id="CAB11777.1"
                /db_xref="GI:2632268"
                /db_xref="GOA:P05648"
                /db_xref="UniProt/Swiss-Prot:P05648"
                /translation="MENILDLIWQALAQIEKKLSKPSFETWMKSTKAHSLQGDTLTIT
                APNEFARDWLESRYLHLIADTIYELTGEELSIKFYIPQNDVEDFMPKPOVKKAVKED
                TSDFPQMLNPKYTFDTFVIGSGNRFAHAASLAYAEAPAKAYNPLFIYGGVGLGKTHL
                MHAIGHYVIDHNPSAKVYVLSSEKFTNEFINIRDNKAVDFRNRYRNVYLLIDDIQF
                LAGKEQTQEEFFHTFNTLHEESKQIVISSDRPKEIPTLEDRLRSFEWGLITDITPP
                DLETRIAILRKKAKAEGLDIPNEVMLYIANQIDSNIRELEGALIRVVAYSSLINKDIN
                ADLAAEALKDIIPSSKPKVITIKIQRVYVGGQFNKLEDFKAKKRTKSYAFPRQIAMY
                LSREMTDSSLPKIGEEFGGRDHTTYIHAHEKISKLLADDEQLQQHYKEIKEQLK"
TilingArray    413..437
                /probe_name=BG10065_x
                /probe_x_y=174_343<>175_344
                /probe_value=0,1000.0,143.2,113.0,36.1,0,1920.0,279.3,131.
                3,29.1,0,2502.0,328.7,162.3,56.7,0,2642.0,476.8,208.0,34.2
                ,0,676.3,100.1,57.0,12.9,0,2226.0,247.5,112.0,21.2,0,4417.
Close

```

Figure 5.7 GenBank format file with tiling microarray annotation. (1) Newly introduced feature key, ‘Tiling_array’, is shown with its position on the genome. (2) The two-dimensional position of each probe is given such as ‘/probe_x_y=174_343<>175_344’. The former pair values mean the x-y coordinates of the perfect match probe on the microarray while the latter pair values mean the x-y coordinates of the one base mismatch probe on the microarray. (3) The means and standard deviations of the probes, are shown.

Consequently, the EMBL/GenBank file is added by numerous entries of new features of tiling microarray along the genome DNA sequence. The file is saved and can be handled by most of ordinal sequence analysis software tools. Since the IMC series software is capable of *in silico* cloning experiments, with its PCR function, a specified region of the genomic sequence is duplicated with the features on it. Namely, any portion of genome with tiling microarray probes and expression intensities are reproduced easily. A drawing sample is presented in **Figure 5.8**.



Figure 5.8 (a) Microarray data format and profiles for RNA hybridization. The means, standard deviations and number of pixels measured by image scanner, are listed as well as the probe coordinates on the microarray. (b) Probe file format of the tiling microarray. Probe set names, coordinates on the microarray, and probe sequences are listed. Bases in red show mismatch base for MM. (c) The main viewing window of tiling microarray software. Rectangles painted in pale green are tiling microarray probes. Arrows in red and blue are CDSs and rRNAs. The vertical bars denote the corrected probe intensities.

5.3.2 Correction of tiling microarray data

(1) Hybridization quality indicators

Among the results from hybridization of sub-millions of probes to sample DNA, considerably large number of the intensities measured, may be outliers. There are several functions to cut off or eliminate them from analysis. The most basic of them is that of the cut off parameters of upper and/or lower limit to the means and standard deviations of probe intensities. There are also the cut off parameters by ratio, in these parameters the upper and/or lower limit are given by percentage ratios of the number of the to-be-eliminated results. When certain probe intensities are cut off by the above-mentioned methods, on the genome map drawn by IMCAE, these probes are certainly identified from those probes with missing intensities. IMCAE reports these states to indicate each probe with missing intensities or cutoff intensities in different colors each other (**Figure 5.9**).

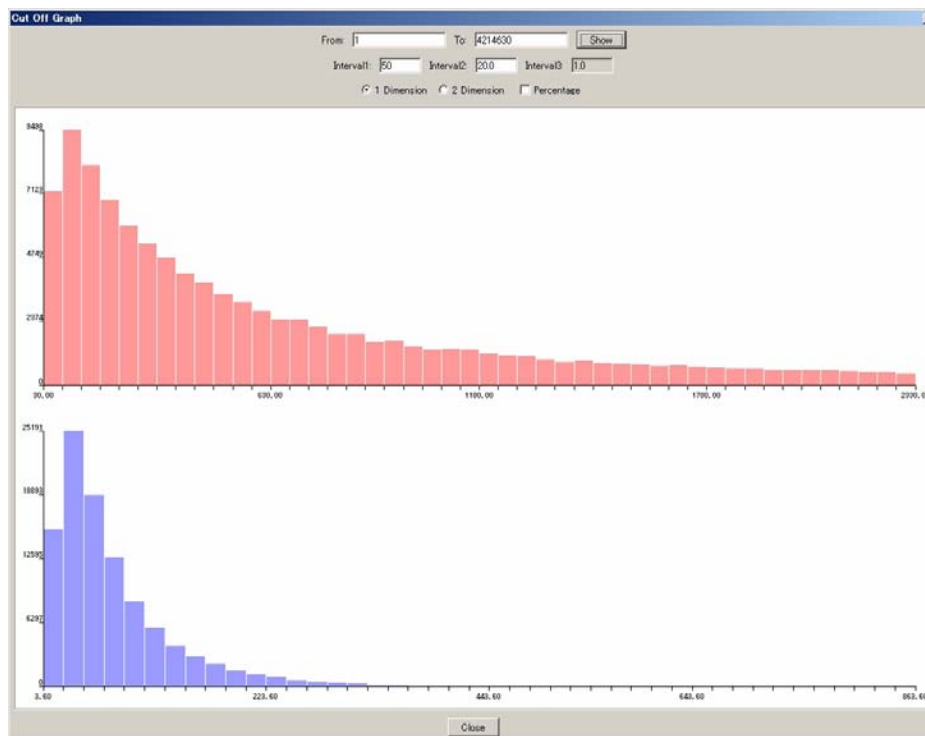


Figure 5.9. Histograms of means (upper) and standard deviations (lower) of all the probe intensities. Width and color of the bars can be changed. Probe intensities shown on the histograms, can be limited within a specified region of the genome. Two dimensional scattered graph is shown when '2 Dimension' button is *ON*. An example is shown **Figure 5.10**.

(2) Detection of corrupted spots

IMCAE provides a visual tool to draw expression intensity per probe with value of data reliability. A two dimensional plot, standard deviations of intensities for X axis and means for Y axis, can be drawn using all or part of intensities within a microarray cutoff line is selected. **Figure 5.10** shows that some of outliers exist on the other sides of cutoff lines.

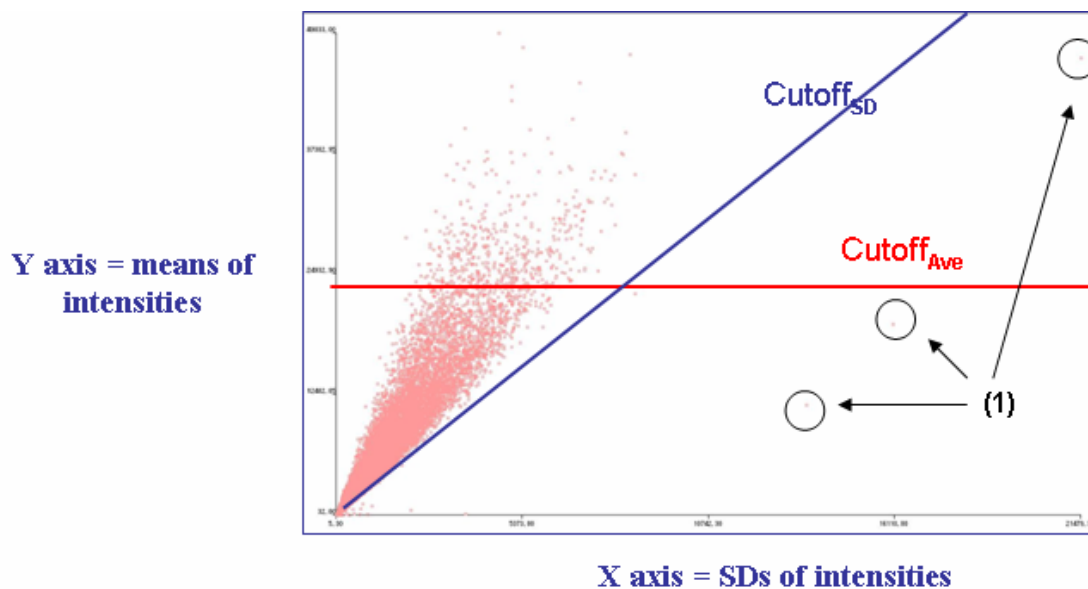


Figure 5.10 Cutting off of outliers. In actual measurements in tiling microarray of *Affymetrix*, one probe consists of up to 100 effective pixels at the highest resolution and only the means and standard deviations of them are provided as primary output. (1) The points circled are estimated as outliers.

(3) Scattered plot: viewer of measurement bias

In IMCAE, a scattered plot for any two sets of measured intensities can be drawn including a scattered plot between PM and MM intensities of the same microarray (**Figure 5.11**). A series of intensities, one of PM, MM or PM minus MM values from the imported microarrays, can be selected on X or Y axis. When 'R-I plot' is selected, an R-I plot is drawn to show the intensity-dependent bias of the microarrays.

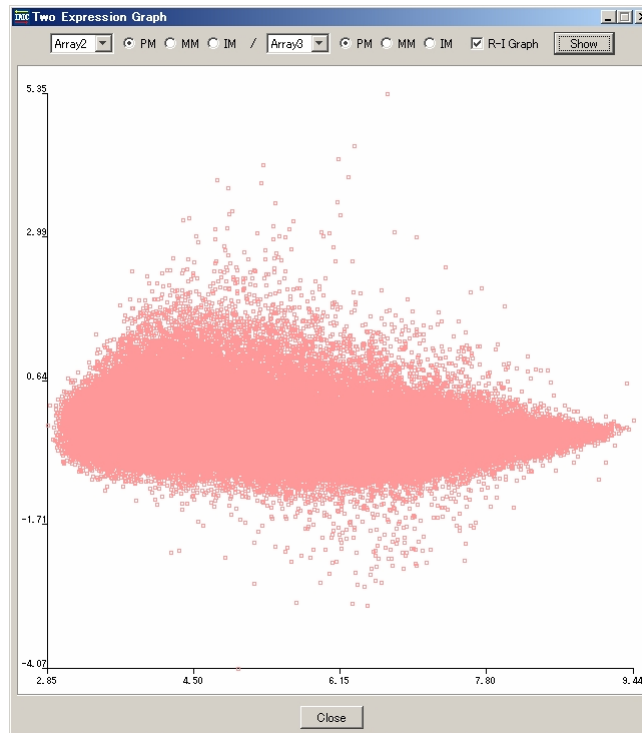


Figure 5.11 R-I plot between intensities comparing two microarrays. X axis denotes the intensity (I). Y axis denotes the ratio (R). The scale is automatically adjusted. This plot shows that there are large variances in the middle range of intensities.

(4) Reduction of effects caused by non-specific binding or cross-hybridization

By using the method described in **Section 5.2**, as for the intensity of each probe, one from PM, MM or PM minus MM, is selectable for each array although it is recommended to use PM minus MM. The expression profiles are also drawn with any of three values. In addition, it is also possible to draw three profiles of PM, MM and PM minus MM of the same array in parallel (**Figure 5.12**).

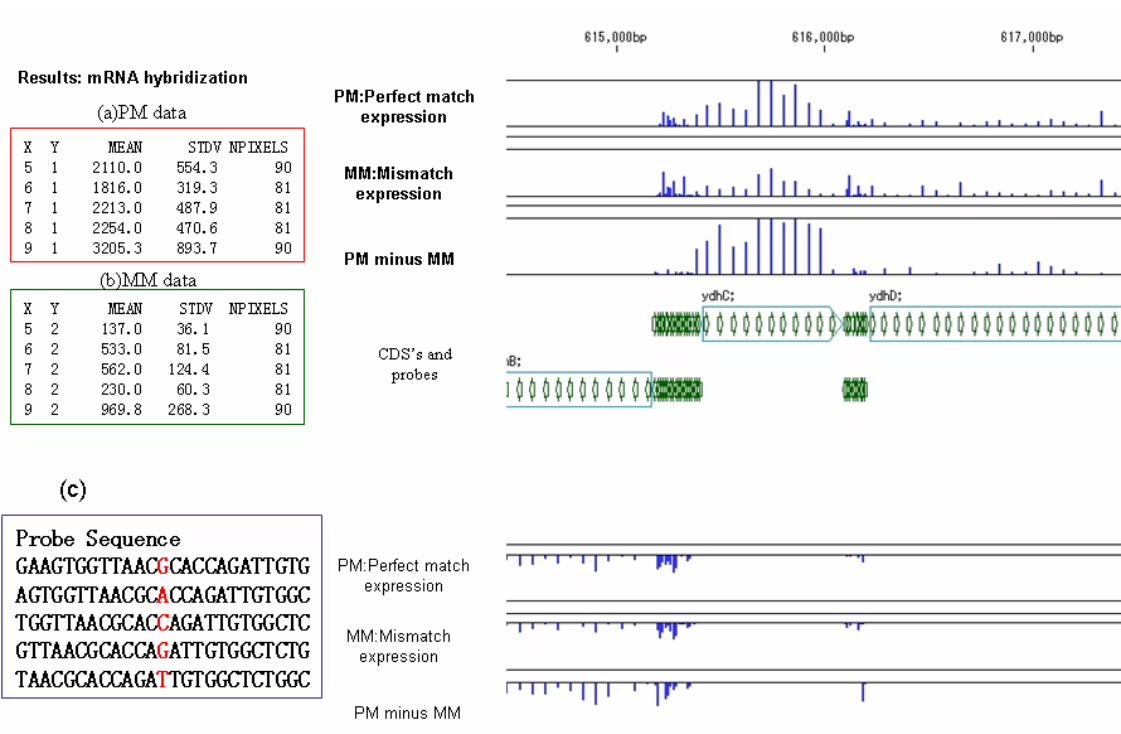


Figure 5.12 Perfect matched vs. one base mismatch probe intensity. (a) Perfect matched data list is shown. 1st and 2nd columns denote the probe location on the array, 3rd column denotes means and standard deviations of expression values on the probe. 5th column denotes NPIXELS which is the number of pixels on the probe image. (b) Same as (a), (c) Probe sequences list. The 25-mer probe has one mismatch base on the 13th base of the mismatch probes.

(4) Windows approach for increasing reliability of tiling array data

To increase the robustness of the estimated intensities, a window approach is also chosen when drawing the tiling microarray expression profile. The single intensity is replaced with the average intensity of all the probes within the window size reach of the genome positions. In addition, the upper and lower trimming settings to expel the outliers, are also chosen as parameters to increase the robustness.

(5) Normalization by total sum of expression intensities on entire microarray

IMCAE provides a normalization function for comparison between two microarrays. If checked on the microarray data, the normalization process has applied to all the expression intensities on the microarray. All the intensities, actually the mean of

all the probe pixel intensities, are summed up to obtain total sum of intensities over the microarray, then all the means are divided by the sum. The process leads to those divided numbers very small,

5.3.3 Comparison of tiling microarray data

Arithmetic operators between three different microarray expression experiments are implemented. As mentioned in the previous chapter, the hybridization intensities of genomic fragments which are fragmented in almost same size, is used as calibration factor to normalize the measured intensities of a tiling microarray. IMCAE is implemented with several operators, such as addition, subtraction, division, multiplication and average (**Figure 5.13**). The adding operator provides all the adding results of the two components derived from a same probe along the genome. A coefficient can be placed before each microarray data for a weighted calculation between two microarrays.

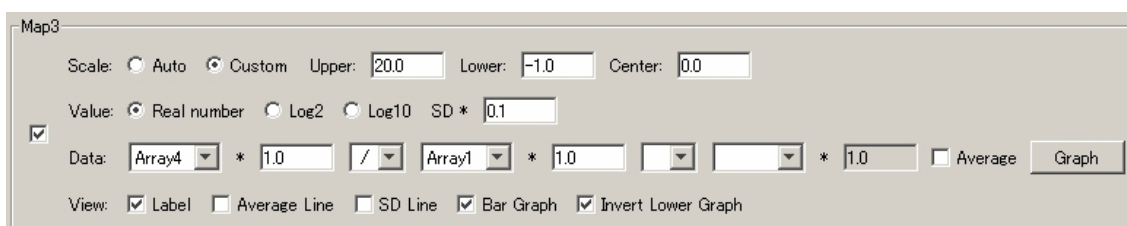


Figure 5.13 Settings of the arithmetic operations between microarrays. On the first row, the scale used is selected as ‘Custom’, which means that the scale is set as indicated by the three parameters on right. On the second row, two of logarithm presentations are selectable. On the third row, this indicates that Array4 is divided by Array1 with coefficients are 1.0. Four arithmetic operators between three microarrays are implemented, as well as an average operator on two or three microarrays. One click on the graph button activates drawing of comparison plot including R-I plot. On the fourth row, five options to plot the corresponding profile on map are shown.

The average operator is applied in the case of replicated experiments on the same sample. When the reproducibility of the microarray experiment is not adequate enough, a next best solution is to perform the experiment repeatedly and obtain replicated results on the same sample. In this case, the average operator is required. In the **Figure 5.14**, an example of normalization by genomic fragments is shown.

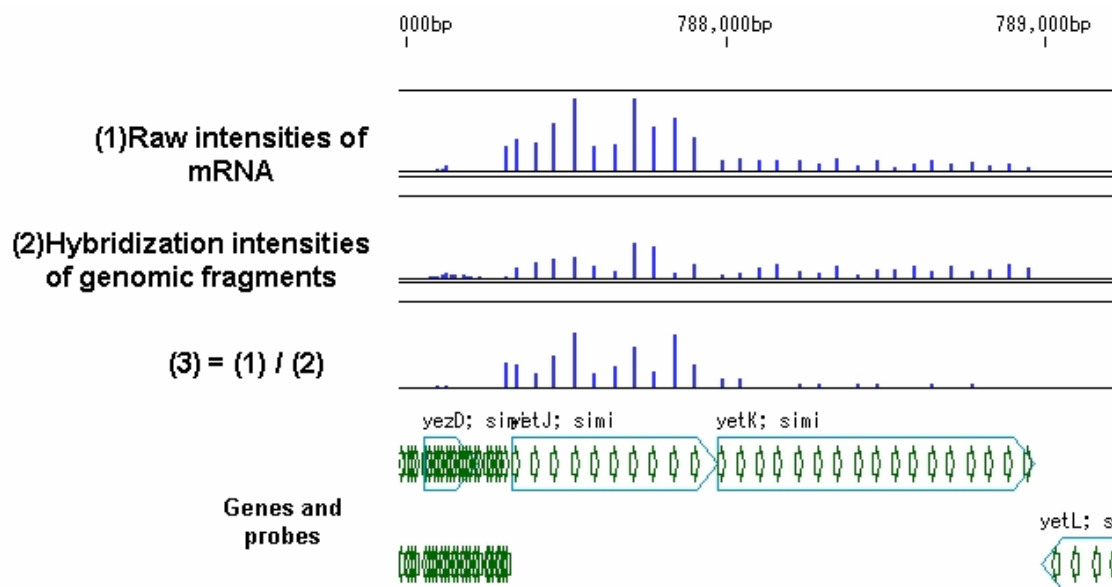


Figure 5.14 Normalization by expression of genomic fragments hybridization. The genomic fragments are evenly distributed on the genome, therefore it is regarded as index for site-specific hybridization efficiency of the genome. In the above profiles, the CDS *yetJ*'s expression intensities are still remarkable after subtraction of hybridization efficiency. However, the CDS *yetK*'s expression values are much reduced after subtraction.

5.3.4 Viewing of expression data

(1) Viewing of the expression results with features

IMCAE provides parallel viewing facility of expression profiles and genome annotation. Up to three microarray expression profiles can be displayed in parallel on the feature map (**Figure 5.15**). The expression profiles on each strand of DNA can be shown separately or in superimposed manner (**Figure 5.16**). On the reverse complementary strand, the profiles can be also plotted downwards instead of upward bars to indicate intensities of the probes. Profiles to be displayed on the plot, are ranging from the raw intensities such as PM or MM, to the resulting intensities from the between-microarray arithmetic operations. Three level lines can be changed by a parameter setting, or automatically arranged to the optimal setting.

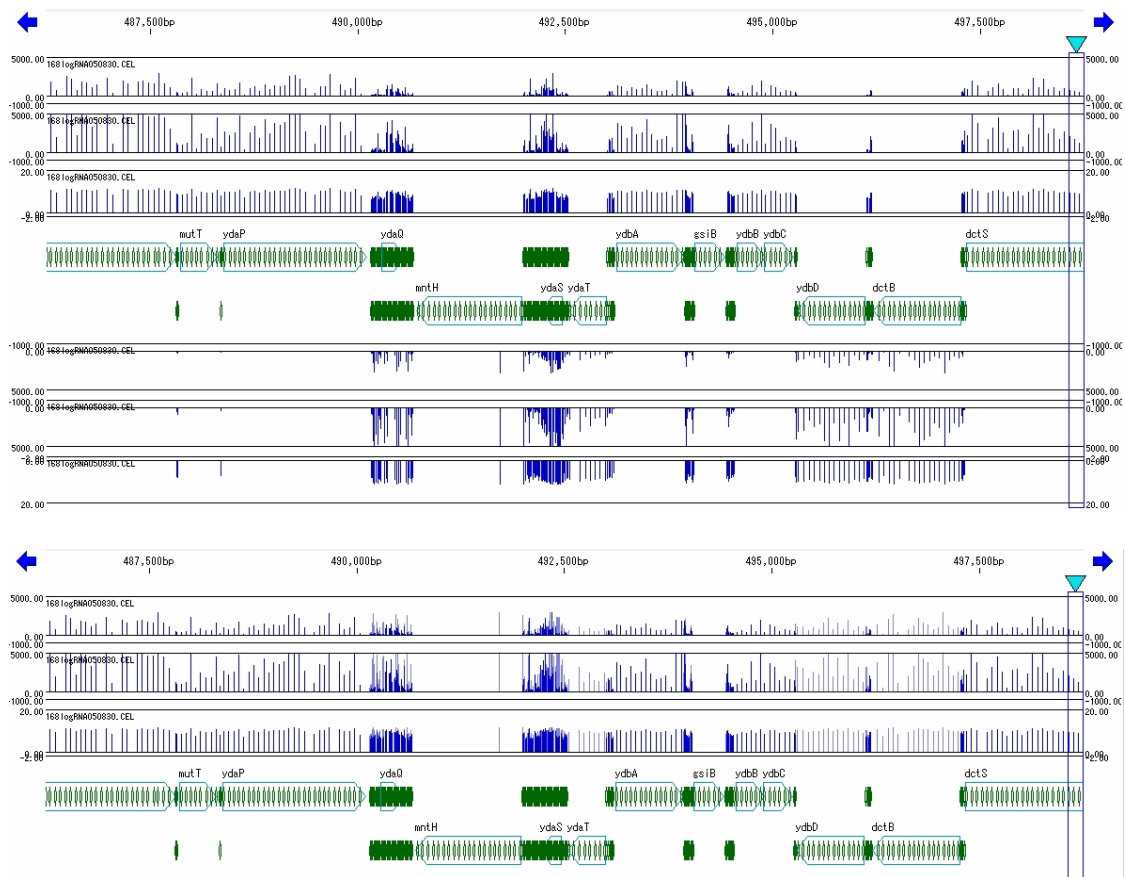


Figure 5.16 Superimpose of the profiles on the reverse complementary strand. The intensity plots on the reverse strands are overlain on profiles on the forward strand (bottom) by clicking the toggle button on the button tool box.

(2) Trimmed means of microarray expressions on CDSs or on intergenic regions

Another function which is implemented on IMCAE, gives trimmed means of all the probe's expression intensities which are planted on each CDS or intergenic region (**Figure 5.17**). If a series of probes are located on any CDS of the genome, this function provides that statistical results from the intensity values of every probe on it. Namely, the users of IMCAE can be provided by the average intensity of expression in the range of a CDS or an intergenic region. This function is consequently compatible to the gene level microarray analytical function which shows an ordered list of expression intensity per gene.

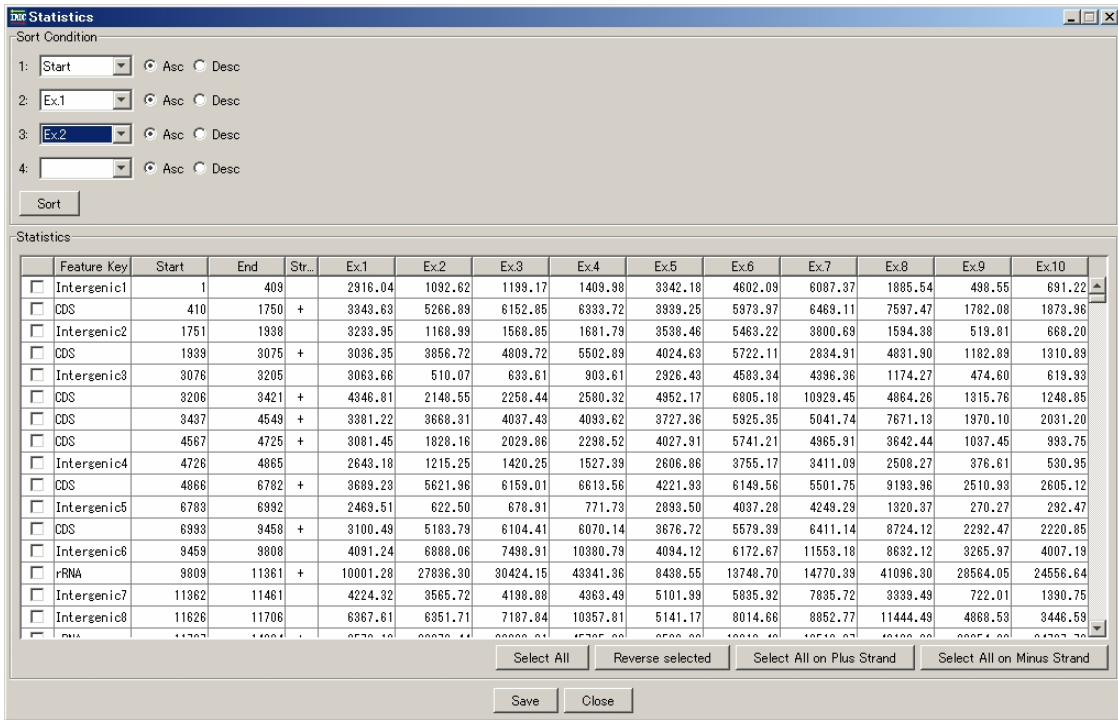


Figure 5.17 Gene level expression table. The Window shows the average intensities per CDS or per one intergenic region. This is compatible to the common presentation of gene level DNA microarray. (1) This list can be sorted by 4 different sorting keys. (1) ‘Feature Key’ denotes the type of regions, such as CDS, rRNA and Intergenic region. (3) ‘Start’ denotes the starting base number on the genome. (4) ‘End’ denotes the end base number on the genome. (5) ‘Strand’ denotes the strand on which each feature is located. (6) ‘Ex1’ to ‘Ex10’ denote the average intensities on the feature of different microarrays.

(3) Clustering of genes with similar expression pattern between microarrays

Gene level expression matrix for up to ten microarrays can be extracted from the above function of trimmed means of the probes on all the CDSs. IMCAE also provides common analysis of gene level expression clustering using UPGMA. A result is shown in **Figure 5.18** where 10 microarrays x 4,100 genes expression matrix is shown. The label for the genes is selectable among the all qualifiers given to the CDS feature. By using the position information about every gene, this matrix has many direct links to the corresponding features on the feature map and nucleotide sequences. The number of probes to be allocated on each gene is also indicated. Color assignment is changeable between two color codes which can be freely selectable, with also selectable gradient scale. Logarithm presentations of \log_2 and \log_{10} are also implemented as options.

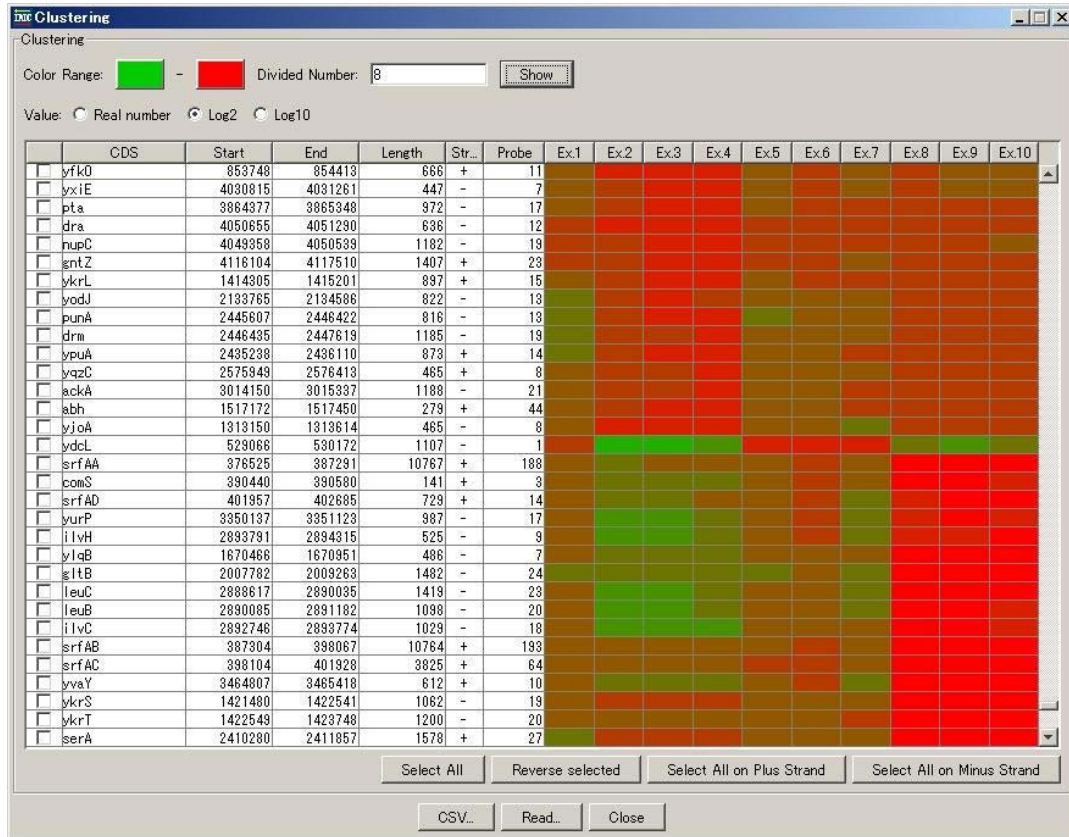


Figure 5.18 Clustering of a gene expression matrix. The Window graphically shows that a gene expression map of clustering genes with similar expression by using UPGMA method. The column, ‘CDS’, denotes gene name assigned to each gene region. The column, ‘Start’ and ‘End’ denote the start and end positions, respectively, of each CDS on the genome sequence. The column, ‘Length’ denotes the nucleotide length of each CDS. ‘Strand’ column denotes the direction of transcription for each CDS. The column ‘Probe’ denotes the number of probes which are mapped on the CDS. The colored columns display the trimmed mean intensity of each CDS and each microarray participated in the analysis. The colors are allocated in flexible manner. After selection of two colors, interpolating colors are generated between the two colors. Because of wide dynamic range of probe intensities, logarithm scales are also selectable. Any line is selectable and sub set of the gene expression map is easily generated after storing as a CSV format file.

A subset is derived from the whole set of genes by manual selection (**Figure 5.19**). Since a subset is also the same format as the whole genes set file, it can be handle in the same manner of the whole set.

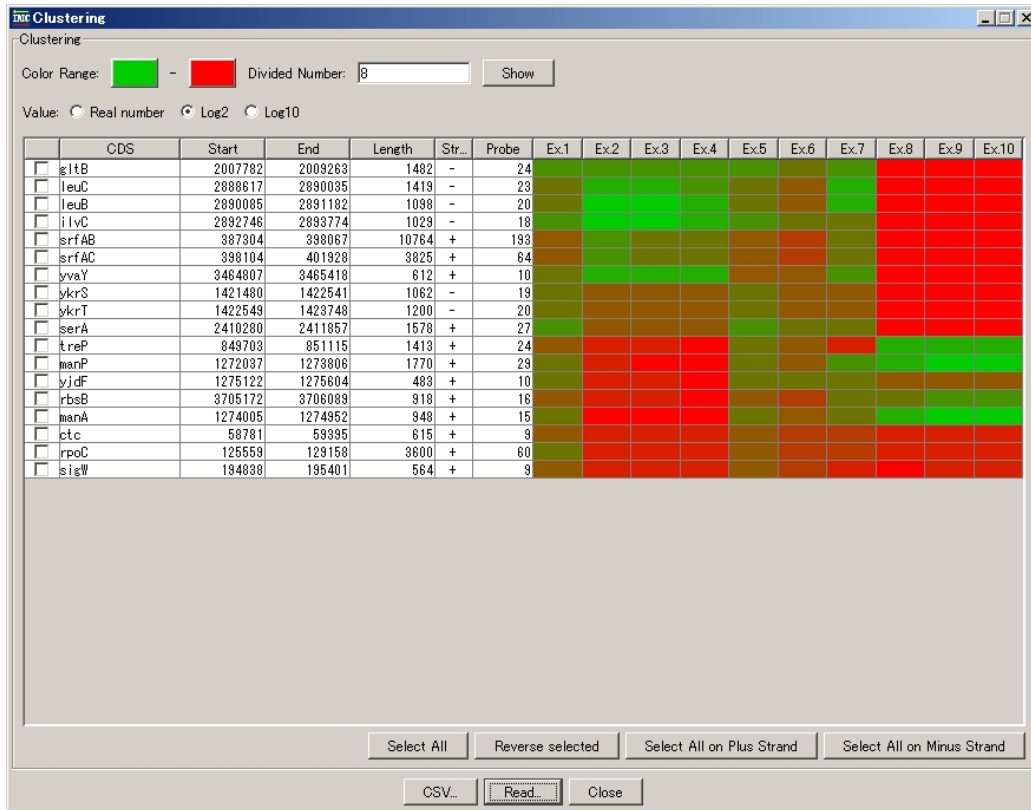


Figure 5.19 Manually selected set of genes with similar patterns of expression. This is derived from the whole set of genes displayed in the Figure 5.18.

5.4 Discussion

(1) Portability and visibility with GenBank/EMBL format files

In IMCAE, I implemented a storing function of the complicated microarray data into single portable GenBank/EMBL text format file which is commonly visible with any available viewer and editor software. In addition, another advantage of handling a probe as a feature is that any region of tiling arrayed genome could be copied or transferred without affecting remaining regions of the genome. For example, only a small part of the tiling arrayed genome can be duplicated by the PCR function of IMC without losing any information about the microarray experiment. Occasionally, the file of a whole genome tiling array amounts to several hundred mega bytes in size, therefore it is useful to divide it into several partial files without losing any corresponding array data and annotated features. In addition, this kind of portability and reproducibility is also useful to exchange the array data with the research collaborators.

(2) Multiply allocated probes on a genome

In IMCAE, multiply allocated probe sequences, such as probes on rRNAs, tRNAs or highly homologous genes, are recorded as multiple entries for different sites on the genome sequence. This handling should be considered to be different from handling of probes on the unique sequence genes. When the concentration of total RNA is low, most of the probe molecules are not hybridized and not saturated, then the intensity measured on the probe is regarded as total sum of the corresponding RNA molecules which have complementary sequence to the probe. Consequently, the intensities on the multiply allocated probes should be divided by the number of sites which share the same probe sequence. When the microarray probe is designed to have exact copy number of probes for each multiply allocated probe, the intensities measured on the probes are thought to be equal to each other and reflect actual intensities on the probes. On the other hand, when the concentration of total RNA is high, most of the probe molecules are hybridized and saturated, then the intensity measured on the probe can be underestimated.

(3) Saturated intensities

In the same sense, saturated intensity can be regarded as underestimated. When the dynamic range of the image scanner which was used to measure the microarray, is given, saturated intensity can be easily estimated. However, some of image scanner software may have corrected saturated intensities. In such cases, the algorithms taken by

these image analyzing software tools are necessary to estimate the saturated intensity.

However, further consideration is necessary when handling PM and MM intensities provided by Affymetrix GeneChip. When considering PM and MM intensities, there are three cases of saturated intensities, both PM and MM saturated, PM saturated and MM unsaturated, and PM unsaturated and MM saturated. IMCAE identifies three cases because these cases should be differently treated. When a PM intensity is saturated and the corresponding MM intensity is not saturated, this is easily speculated as the actual intensity on the probe is underestimated. Therefore, the intensity could be used with remarks such as 'Underestimated'. In contrast, when a PM intensity is not saturated and the corresponding MM intensity is saturated, the original scanned image of the MM probe might be in low quality. Therefore the PM intensity can be used carefully. However, most of microarray software tools treat these intensities as outliers and do not use for further analysis. When both intensities of a probe pair are saturated, a couple of explanations can be given, such as too heavy cross-hybridization, or corruption of both images of PM and MM, or others. In any case, the intensities should be handled as outliers.

(4) Microarray analysis with sequence analysis functions

The standard edition of IMC software provides a wide variety of sequence analysis functions about genomic sequences, and IMCAE inherits almost all the functions of IMC, as well as its original microarray analysis functions. Therefore, these combined functions of IMCAE can provide various useful operations when handling microarray data, while most of other microarray software tools were specially designed to analyze microarray data only. For example, when some novel features are identified on the genome after analysis on the microarray data, these features can be directly inserted on the same data file which describes the genome sequence and microarray results. Another example is that of searching functions of IMC ranging from keywords search, pattern matching, and homology search against reference genomes or databases. The combination of microarray analysis tools and information retrieved by the searching tools accelerates and widens the comprehensive analysis.

(5) Comparison analysis using arithmetic functions between arrays

A comparison analysis between two microarrays is performed in common. The results from genomic fragments hybridization are used to normalize the mRNA expression levels. In addition, the replicated microarray experiments are commonly performed to increase the reliability of the results. The arithmetic operators on IMCAE

between all the intensities of arrays together, provide several important functions, such as averaging replicated microarray experiments, normalization by genome fragment hybridization and raw data analysis of PM and MM intensities.

(6) Strand specific and strand non-specific viewer

The most of ChIP-chip experiments provides site-specific but not strand-specific data along a genome sequence, while RNA expression data is provided as strand-specific. Thus, software tools to handling the both types of the microarray experiments, are required with an overlay function of reverse complementary strand data onto the forward strand. IMCAE provides both viewing methods of strand-specific and strand-non-specific.

(7) Flexibility against the genome sequence update

Currently, most of the genome sequences, which were already published, are neither the final version nor fixed yet. These nucleotide sequences will be occasionally updated with additional annotations. The newly updated information can not be used until the microarray provider modifies the data and provides it to the users. IMCAE is implemented with the function of importing probes sequences and mapping them onto target genome sequence. By using this function, up-to-date analysis is possible.

(8) Requirements of handling raw data

Since the complicated characteristics of microarray data was discovered, thousands of papers about microarray data correction, were reported annually (Chen et al. 2004). Many novel methodologies were introduced, and the analysis on the microarray is said to become too sophisticated and complicated for the ordinary biomedical researchers (Chen et al. 2004). These complicated algorithms and processes are becoming black boxes for the most of researchers. IMCAE provides raw data handling for the researchers. If preferred, the simplest analysis can be performed, such as using only PM intensities without any correction to the data.

(9) Common handling of tiling and quasi- or non-tiling microarray data

The probe positions on the genome sequence are rarely concerned previously and the results from the gene level microarray are presented as a list or matrix of clusters which share similar expression patterns. In a gene level oligonucleotide microarray such as Affymetrix GeneChip, 11 to 20 probes are allocated on each genic region. Therefore, the tiling microarray is regarded as the extension from the gene level

expression microarrays. IMCAE can interpret the results from tiling microarrays into those of gene level analysis. Gene clustering is provided after extracting the intensities on the genic regions.

(10) Compatibility with MIAME format

In response to the widely acknowledged need for public repositories for microarray data, MIAME, the Minimum Information About a Microarray Experiment, was proposed (Brazma 2001). MIAME provides only a standard or guideline for the format and information about microarray results to be stored. Several actual databases were already established according to the MIAME standard (Brazma 2003, Parkinson 2005, Menten 2004, Ball 2005). In the concept of MIAME standard, three major kinds of information resulted from microarray experiments, gene expression matrix, gene annotation and sample annotation, are necessary. Among them, the gene annotation is recommended to be linked to public sequence databases such as GenBank or EMBL, while the sample annotation has currently no database like sequence database. MIAME proposed that some of gene annotation should be also included in microarray databases. However, this handling requires complicated many-to-many relationship between genes. Therefore, the MIAME implementation is still controversial.

My proposal of integrating microarray expression data with its annotated sequence will be one of the solutions for the above problem, although it requires a large size file to describe a eukaryotic microarray experiments. The links between genes with related or similar expression can be easily performed using the subset table from the gene expression matrix of IMCAE.

Chapter 6

**Asymmetry found in the local composition of GC
and its correlation to the transcript units and
directions in genomes**

6.1 Introduction

Today, over 400 complete microbial genomes are sequenced. This widens the chance to study the causes of the asymmetries of the base composition found in microbial genomes. The biased base composition in the local region of genomes have been a clue to predict horizontal transferred regions in bacterial genomes, isochores in eukaryotes (Bernardi 1993) and base skew information such as GC skew and AT skew is also important to determine replication origin and terminus in microbial genomes (Lobry 1996; Mackiewicz 1999). However, the causes of the asymmetries in the base composition are still controversial.

The cause of the GC asymmetries has been explained mainly by five mechanisms, two from mutation, one from repair error, one from natural selection and one from tRNA availability. The replication-coupled mutation hypothesis explains an excess of G over C in the leading strand (Lobry 1996). While DNA replication is occurred, the leading strand is staying in the single-stranded longer than the lagging strand. Therefore the leading strand sequence is more prone than the lagging strand sequence. The transcription-coupled mutation hypothesis can explain the similar mechanism as that of replication-coupled mutation. When RNA transcription is occurred, RNA is transcribed along the complementary strand against coding strand while the coding strand is not paired. Therefore the coding strand sequence is more prone to mutation during transcription. In addition, the transcription-coupled repair hypothesis explains an excess of purine over pyrimidine in the coding strand. In this mechanism, pyrimidine dimers are repaired by enzymes after transcription. This causes the bias of purine over pyrimidine (Francino et al. 1996). Natural selection on individual nucleotide also explains biased codon usages and sometimes amino acid compositions, such as arginine composition over lysine to obtain thermostability in a higher temperature environment (Trovato et al. 1999, Nishio et al. 2003). While Nishizawa and Nishizawa shows the existence of correlation between GC content at third codon positions and local bias in arginine-lysine usages (Nishizawa and Nishizawa 1998). The number of tRNA on a genome varies between the different genomes. This also affects the nucleotide composition of coding regions.

To collect the evidences of the proof about the causes of the asymmetries between the strands of genomes, local and regional analysis of the skews are necessary. Francino and Ochman examined the extraordinary long intergenic regions with the known transcription start point and terminator of *Escherichia coli* K-12 (Francino and Ochman 2001). Lobry and Sueoka examined the skews on overall intergenic regions

and the first, second and third position of the codons (Lobry and Soeoka 2002). None of the single hypothesis seems to directly prove the causes of the asymmetries found in genomes. In addition, other mutations and selections also affect the asymmetries.

IMC (*in silico* MolecularCloning; see **Chapter 3**) has many functions to visualize base compositions such as GC content, GC skew, AT skew along with the positions of individual nucleotides in genomes, which is used to examine genome structure based on base compositions. I illustrate species-specific and site-specific GC and AT compositional asymmetries for several genomes.

By examining those skews using IMC, I obtained a fact that transcription direction is coincided with local GC skew and AT skew for the genomes low GC content such as *Clostridium perfringens* strain 13 (Shimizu et al. 2002) and *Fusobacterium nucleatum nucleatum* ATCC 25586 (Kapatral et al. 2002). This fact strongly supports the hypothesis of transcription-coupled mutation is the major bias of the asymmetries found between both strands of genomes.

6.2 Methods and Algorithms

(1) Microbial nucleotide sequences

Nucleotide sequences are mainly obtained from the NCBI web site. NCBI provides two types of genomic nucleotide sequences with annotation, GenBank (www.ncbi.nih.gov/genbank/genomes) and RefSeq (www.ncbi.nih.gov/refseq/release/microbial). Since the RefSeq is a somewhat automatically annotated, the database is less biased.

(2) GC skew

GC skew is defined as the difference between G and C composition in one strand in a DNA sequence. The profile of GC skew in a genome is provided with a window approach. In the present system, GC skew for i th nucleotide in the plus strand of a genome was represented by **Eq.6.1**.

$$\text{GCSkew}_{w,i} = (G_{w,i} - C_{w,i}) / (G_{w,i} + C_{w,i}). \quad (6.1)$$

where, w is the window size, $G(w, i)$ and $C(w, i)$ are the numbers of guanine and cytosine nucleotides, respectively, with the i th nucleotide as the center of a window size with w nucleotides.

(3) Definition of genic and intergenic regions

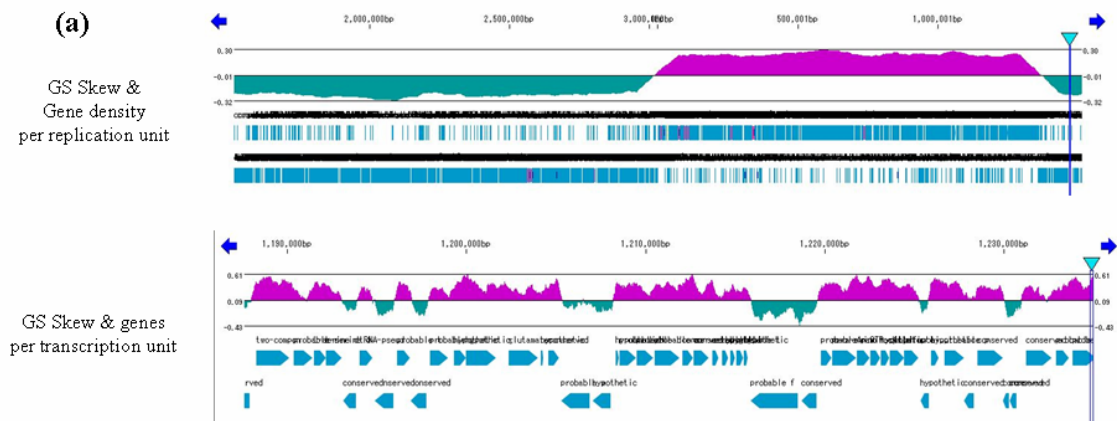
To identify genic and intergenic regions, CDSs boundaries annotated of the database entries are used. Although a very few cases of introns are annotated on some of the genomes, such as those found in *Bacillus subtilis* genome.

(4) Gene cluster

A gene cluster is defined as an uninterrupted series of genes or CDSs whose transcription directions are identical. Therefore, if the directions of transcription direction of CDSs are not change during a long stretch of genomic sequence, it is regarded as a single gene cluster.

6.3 Results

I arbitrary set window size (w) and step size (ss) can be arbitrarily set. The step size is a unit to calculate a base composition. When I set $ss = k$, the base composition is calculated every k bases in genomes. **Figure 6.1 (abc)** shows GC skew with $w = 10,000$ bases and $w = 500$ bases in *Clostridium perfringens*. The global asymmetry of GC skew is observed in *C. perfringens*, that is, GC skew (1000, i) switches from negative to positive at the center of this figure along with genome from left to right, and from positive to negative at the right side of the genome. In addition to the global GC skew observed in *C. perfringens*, a local GC skew is also observed (see lower in **Figure 6.1 (abc)**). The positive GC skew is seen in the regions including genes on the plus strand, on the other hand, the negative GC skew is seen in those including genes on the minus strand. This indicates that the transcription direction is also associated with the asymmetry of GC skew.



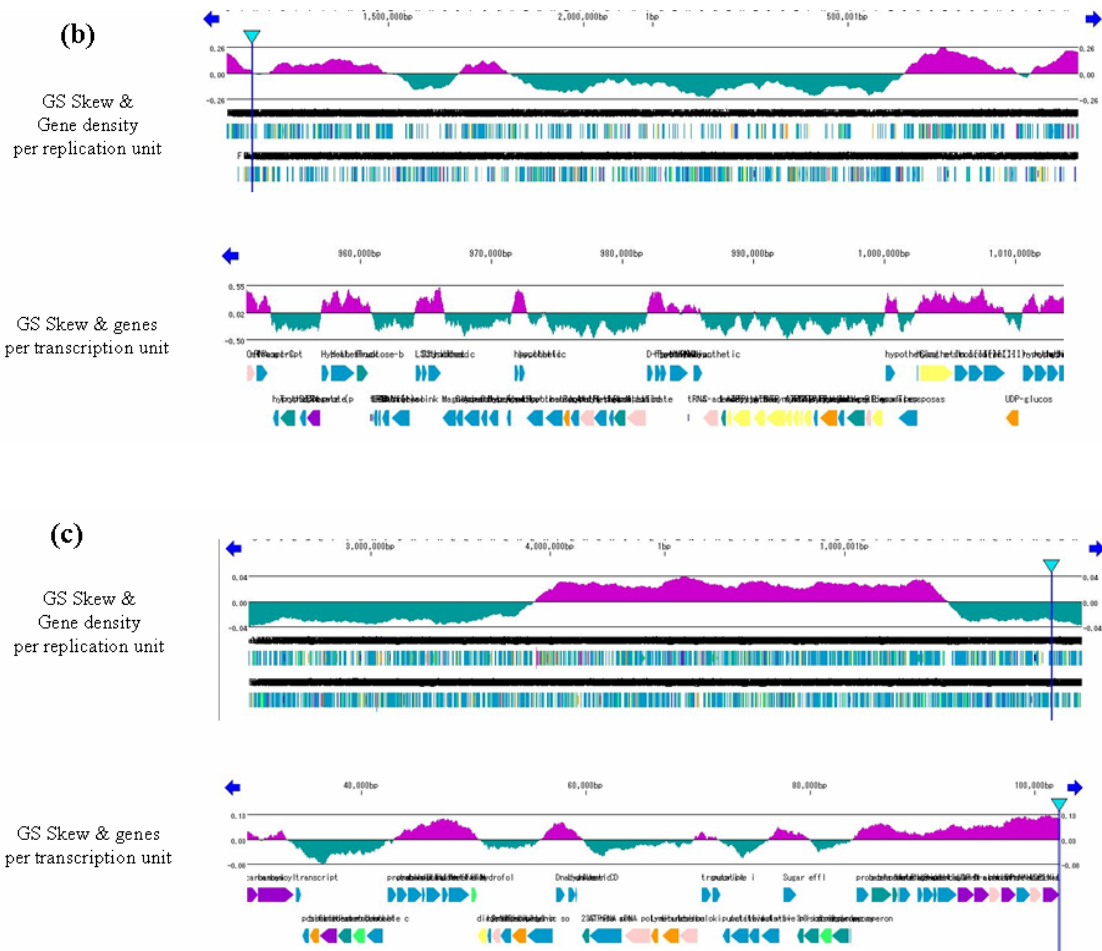
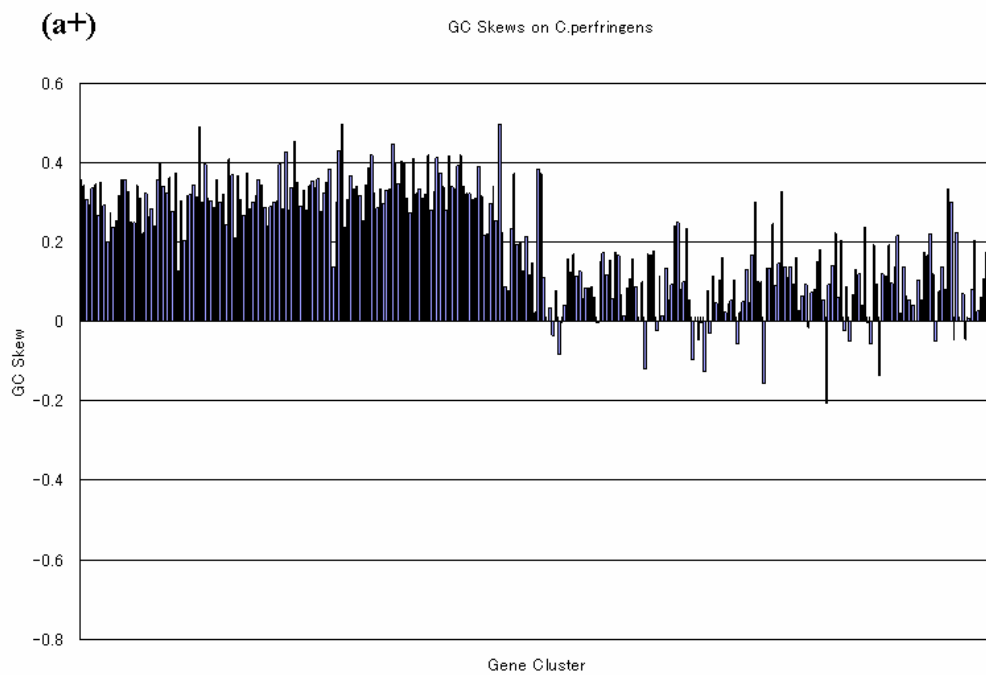


Figure 6.1 Local GC skew profiles. (a) Local GC skew profile in *Clostridium perfringens* Strain 13. GC skew per DNA replication unit are observed clearly (**top**). GC skew per transcription unit (**bottom**). The exceptions to this rule are transposons, some membrane proteins and transcription factors. This genome is classified as a gram negative bacteria, however the GC skew is clearly observed. (b) Local GC skew profile in *Fusobacterium nucleatum nucleatum* ATCC 25586. GC skew per DNA replication unit are observed clearly, however this is not mono-cycled such as seen in gram positive bacteria (**top**). GC skew per transcription unit are observed clearly (**bottom**). (c) Local GC skew profile in the *Escheirchia coli* K-12 genome.

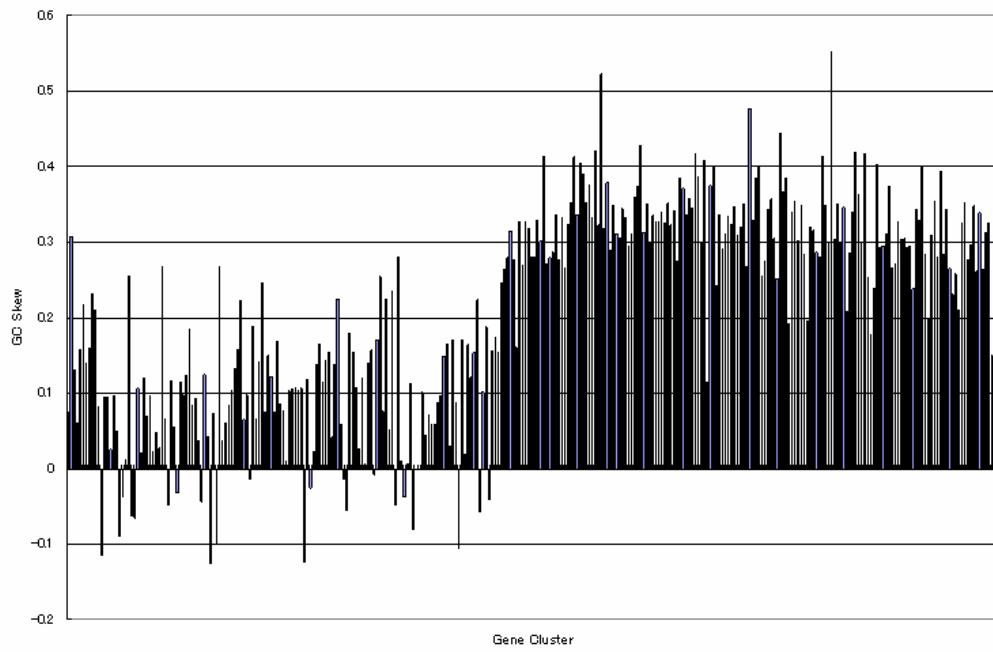
Asymmetry of the global GC skew is also observed in *F. nucleatum*, but the situation is somewhat different from *C. perfringens*, that is, there are four switching points in the global GC skew, but transcription coupled asymmetry of the local GC skew is again observed in *F. nucleatum*.

Figure 6.2 (abc) shows that majority of gene clusters have positive GC skew. For examples, *C.perfringens* has average 0.206 GC skew per gene cluster. On the plus strand, GC skew is higher on the leading strand, which is the left half of the bar chart while lower on the lagging strand, which is the right half of the bar chart (**Figure 6.2 (a+)**). On the minus strand of *C.perfringens*, GC skew is higher on the leading strand, which is the right half of the bar chart while lower on the lagging strand which is the left half of the chart(**Figure 6.2 (a-)**). Even on the lagging strand, the GC skews are almost positive.



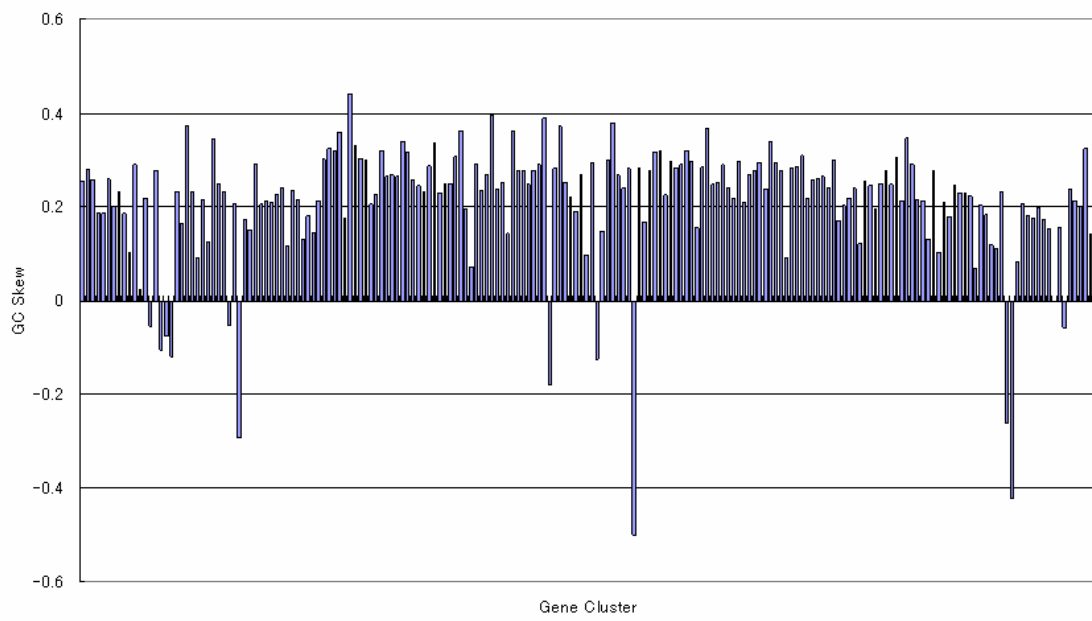
(a-)

GC Skew on minus strand in *C.perfringens*



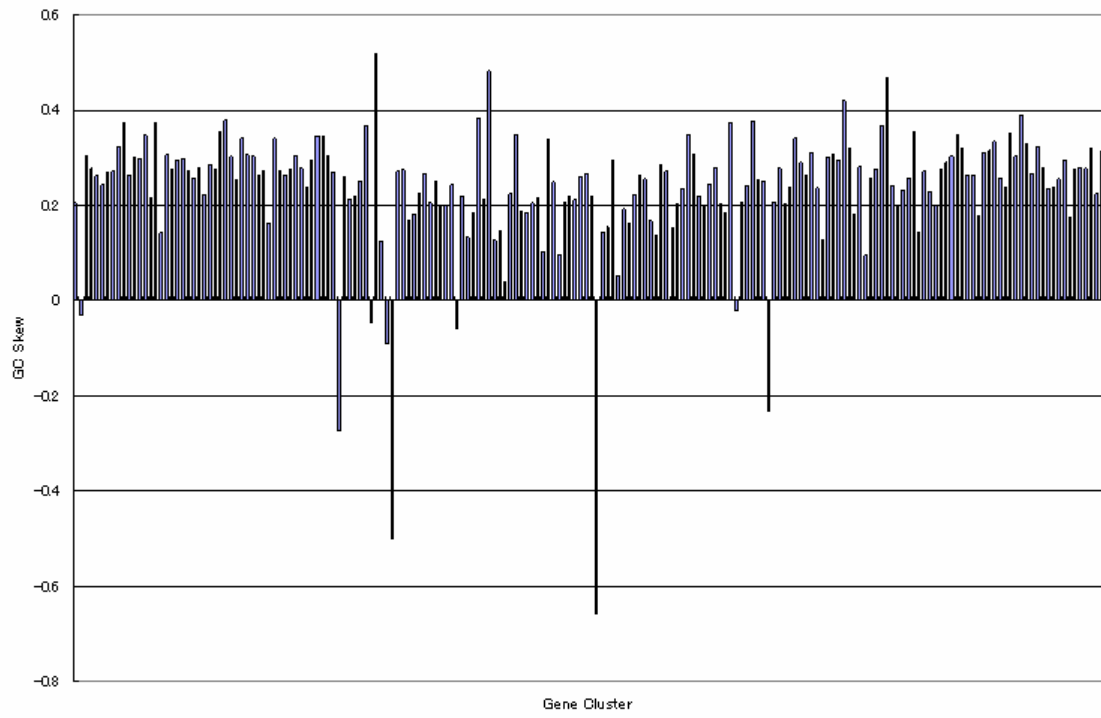
(b+)

GC Skew in *F.nucleatum*



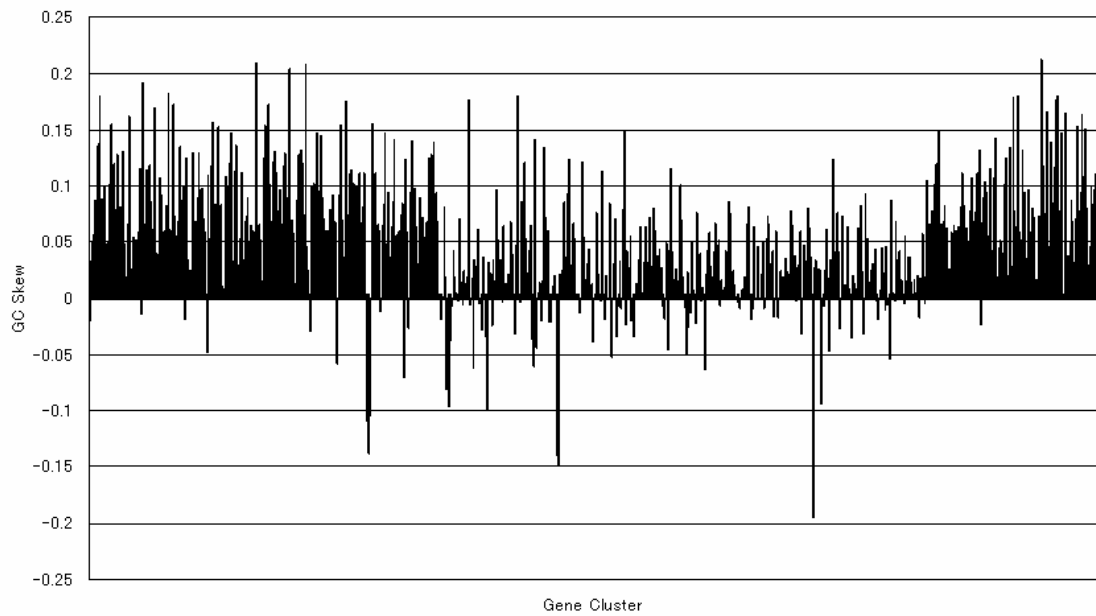
(b-)

GC Skew in *F.nucleatum* on minus strand



(c+)

GC Skew on *E.coli* genome



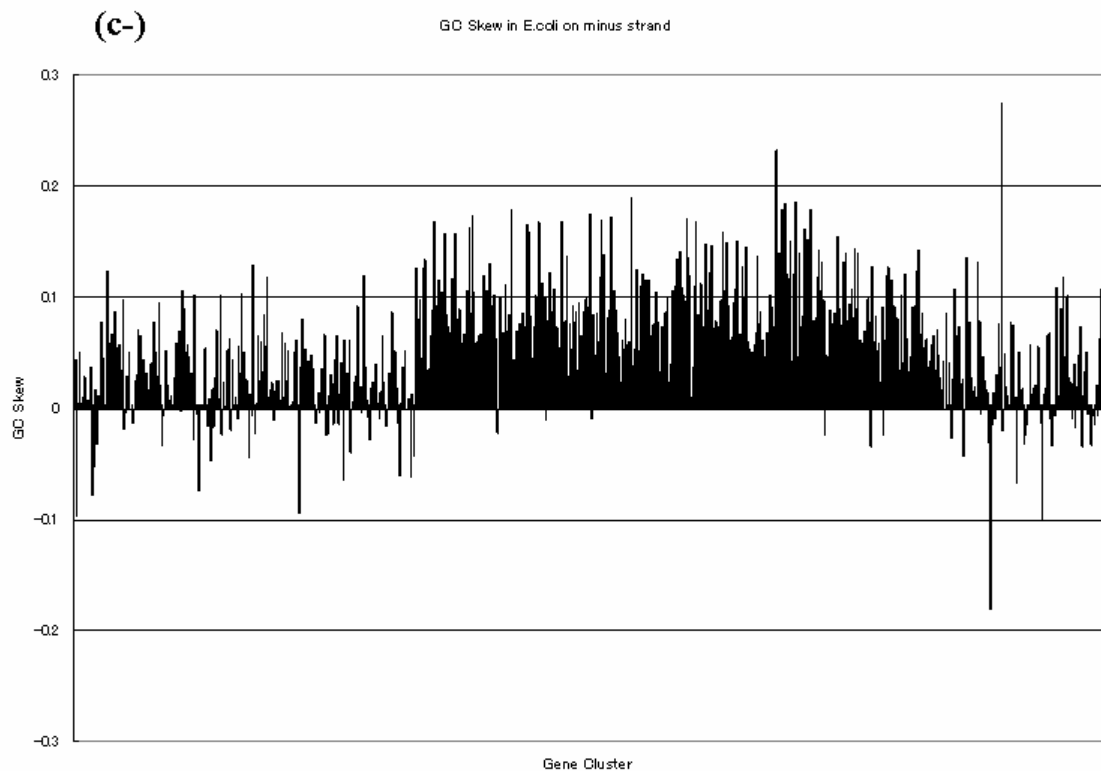


Figure 6.2 GC skews per gene cluster. X axis denotes CDS order from first CDS to the last CDS. Y axis denotes GC skew value in ratio ($-1 < Y < 1$). The skews are measured in the direction of transcription. (a+) GC skews per gene cluster on the plus strand in *C.perfringens*. (a-) GC skews per gene cluster on the minus strand in *C.perfringens*. (b+) GC skews per gene clusters on the plus strand in *F.nucleatum*. (b-) GC skews per gene clusters on the minus strand in *F.nucleatum*. (c+) GC Skews per gene cluster on the plus strand in *E.coli*. 6.2(c-) GC Skews per gene cluster on the minus strand in *E.coli*.

The genome of *F.nucleatum* shows a little different appearance of GC skew per gene cluster. On the plus strand of *F.nucleatum* genome, the GC skews are evenly positive along the whole sequence. Considering about its irregular global GC skew profile, this may show that the replication direction may be only one. In the genome, no homologous gene with the DNA replication initiator, *dnaA*, is found (Kapatral et al. 2002).

In *E. coli* genome, there are two switching points in the global GC skew associated with the origin and terminus of replication but the extent of asymmetry (from -0.04 to 0.04 in the vertical axis) is much smaller than *C. perfringens* and *F. nucleatum*

(the extents from -0.30 to 0.30, and from -0.26 to 0.26, respectively) and the local GC skew is again observed. The asymmetries of global or local skews can be easily examined by setting arbitrary size of the window in the present software.

A local GC skew profile is obtained along the whole genome of *F.nucleatum* (Figure 6.2 (b+-)). In this case, GC skews for 99% of genes in transcribed DNA sequences are positive. This indicates that transcription directions are highly associated to produce bias of GC skews in the local level. Here, the local GC skew in sense or coding strand is referred to as transcription-coupled asymmetry of GC skew (taGC skew). This trend can be observed in *E. coli* (Figure 6.2 (c+-)) and *B. subtilis*.

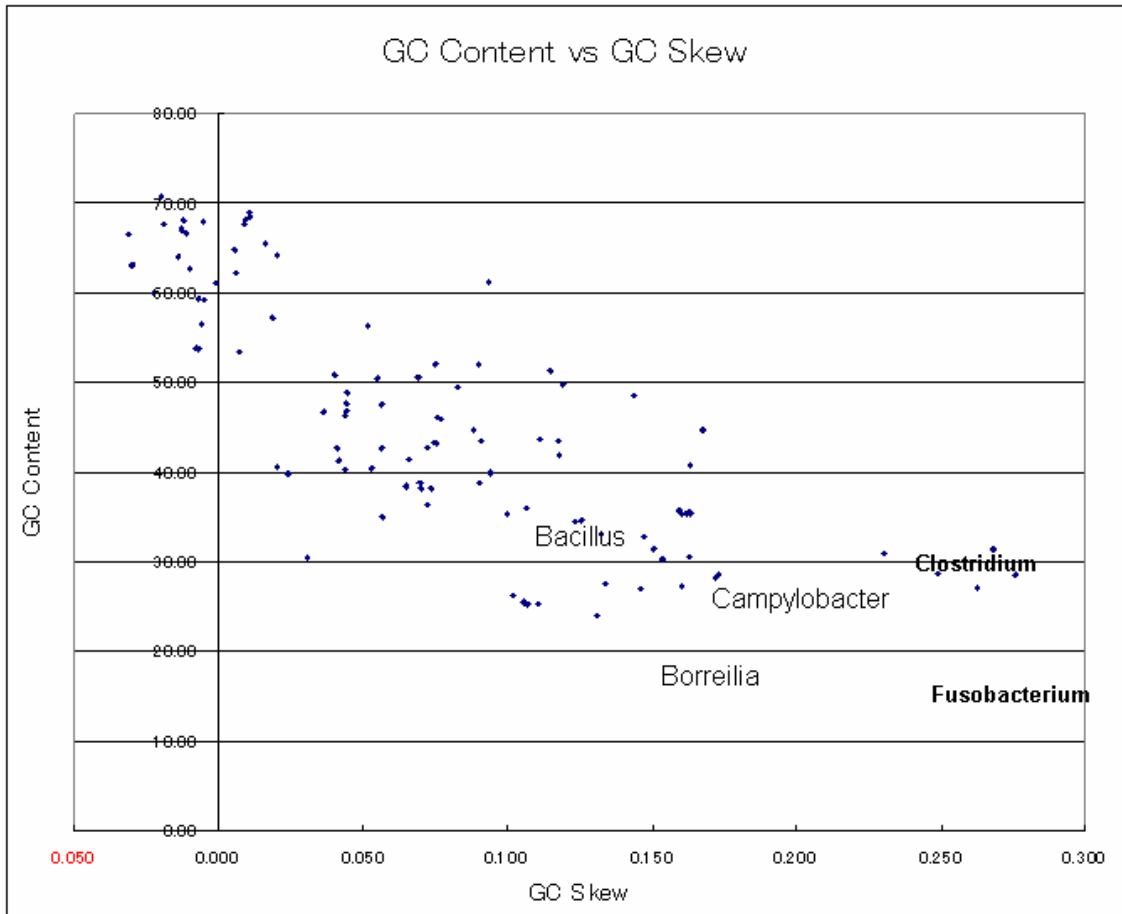


Figure 6.3 Negative correlation is observed between GC content and GC skew. *X* axis denotes GC skew value ($-1 < X < 1$). *Y* axis denoted GC content ($0 < Y < 1$)

positive GC skew, because of decrease of T and increase of C. Therefore, bias mutation effects derived by transcription-coupled asymmetry of base composition can be systematized by plot of genomes by averages of transcription-coupled asymmetries between AT and GT skews (**Figure 6.4**). Genomes for *Chlamydia sp.* *Synechocystis sp.* are obviously classified into the category affected by C to T deaminations. This type of the asymmetries is reported for several eukaryotic genomes (Touchon et al., 2004).

Relatively large number of genomes including *Fusobacterium*, *Clostridium*, *Camphylobacter*, *Bacillus*, and *Borrelia* are plotted into positive region for both transcription-coupled GC and AT skews. In this group, transcription-coupled GC skews are consistent with the fact that G is relatively abundant in comparison of C induced by the deamination process from C to T however reverse factor to this deamination process may reflect the transcription-coupled AT skews. Therefore there may be the other mutation mechanism coupled by transcription. IMC software makes it possible to systematize genomes by the properties based on base composition, and the local GC skew profile may be used as a prediction algorithm of unknown transcription unit.

To investigate the affection of natural selections, thirteen genomes are investigated for codon usages and amino acids usage analysis on every CDS on the genomes. Among them, 10 genomes are observed with the local GC skew, while the other 3 genomes are not observed with the local GC skew. What is clearly observed is that basic and hydrophobic amino acids composition changed when the GC content are increased by genome and third position of codons tend to be GC rich according to the increase of GC content. Especially, the arginine composition increases while the lysine composition decreases (**Figure 6.5**). According to the fact that arginine is coded by CGN and AGR and lysine is coded by AAR, where R is denoted by purine, it is natural that lysine is substituted by arginine. This fact supports that guanine composition increases than that of cytosine.

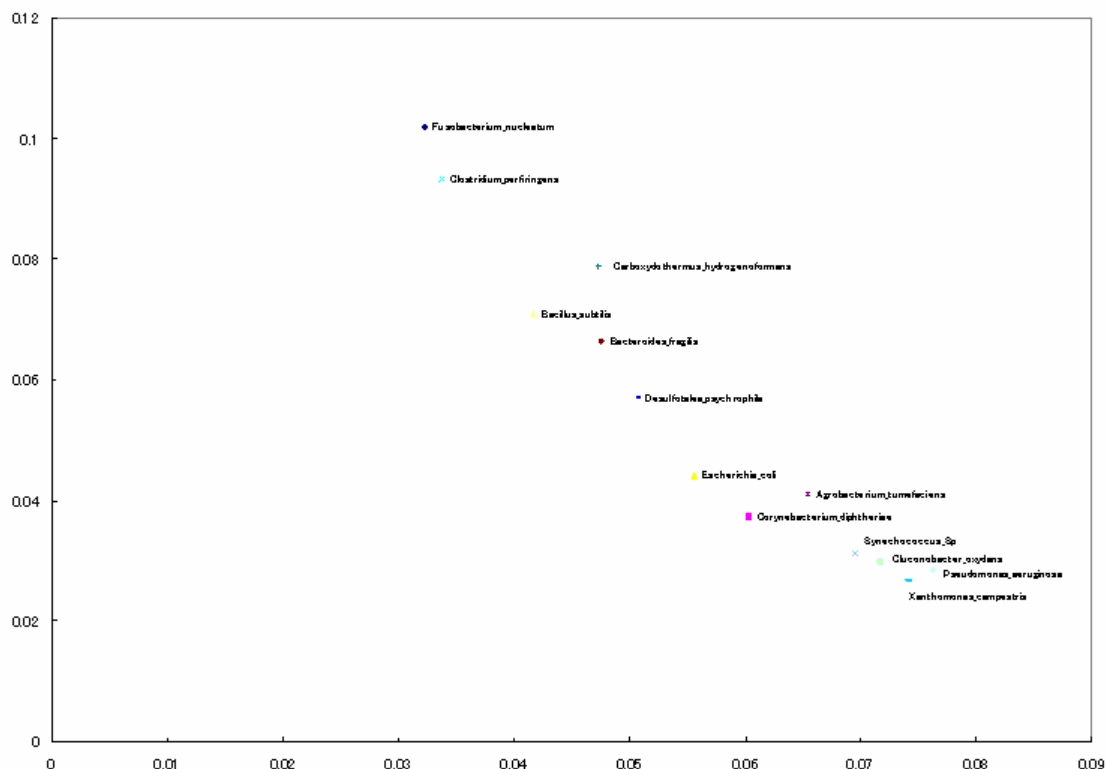


Figure 6.5 Amino acid usages of 13 different genomes. They are observed with the local GC skew and three other genomes which are not observed with the local GC skew. In this plot, composition of basic amino acids (Arg:Arginine, Lys:Lysine) are shown. On X axis denotes Lys composition in ratio. Species listed are *Methanocaldococcus jannaschii* DSM 2661, *Fusobacterium nucleatum nucleatum* ATCC 25586, *Clostridium tetani* E88, *Clostridium acetobutylicum* ATCC 824, *Bacillus anthracis* Sterne, *Lactobacillus acidophilus* NCFM, *Bacillus subtilis subtilis* 168, *Bacteroides fragilis* NCTC 9343, *Acinetobacter sp.* ADP 1, *Escherichia coli* K-12 MG1655, *Corynebacterium glutamicum* ATCC 13032, *Corynebacterium efficiens* YS-314 and *Pseudomonasaeruginosa* PA01, Y axis denotes Arg composition in ratio.

6.4 Discussion

The asymmetrical structure of the global GC skew is interpretable by the results of asymmetric structure in the DNA replication complex (Lobry, 1996). This type of mutation bias may be induced by transcription event, that is, RNA polymerase attached to transcribed strand induces the similar mutation bias in DNA replication complex. DNA polymerases and RNA polymerases may have an identical single ancestor, namely, they are regarded as paralogues against each other. Some infer leads to the conclusion that the cause of the local GC skew asymmetry is not the result of asymmetry of RNA polymerase, but rather it explain that they are merely the newcomers to the genome in comparing with other older habitants of the genome. Experimentally proved transcription initiation sites are reported in the genome of *B. subtilis*.

Asymmetry of the global GC skew is observed as sum of at least two contributions, DNA replication and RNA transcription. In the *E.coli* genome, CDS are evenly distributed either on the leading strand or lagging strand. This gene-organization affect to cancel replication-coupled asymmetry of GC skew because, if the direction of the replication fork is opposite to that of transcription in a CDS, the effects of transcription-coupled and replication-coupled asymmetries of GC skew can be canceled to each other. So the magnitude of the asymmetry of global GC skew is very small in comparison of species with genomes coupled in transcription and replication directions such as *B. subtilis*.

I found coincidence of the transcription direction and asymmetry of GC composition. Such asymmetry had been already known as a genome-scale phenomenon and related to the direction of chromosome replication, however, local and transcription level of this phenomenon is rarely reported (Francine and Ochman 1996). After checking all the available prokaryotic genomes, low GC-content genomes are likely to show this tendency. Especially remarkable are the genomes of some *Clostridii* and *Fusobacterium nucleatum*. Their AT compositional asymmetries are also correlated with those of GC.

The nucleotide compositional asymmetry found in microbial genomes is not clearly explained by any single hypothesis. Possible interpretation is that different causes, such as replication-coupled mutation, transcription-coupled mutation and repair,

biased distribution of tRNA and natural selections, more or less affected the asymmetry. Therefore, it is useful to describe the asymmetry by the following equation (eq. 6.2).

$$S_{\text{observed}} = f(M_{\text{Replication}}, M_{\text{Transcription}}, R_{\text{Transcription}}, D_{\text{tRNA}}, \text{NS}) \quad (6.2)$$

where S_{observed} is defined as the observed asymmetry; $M_{\text{Replication}}$ denotes asymmetry caused by the replication-coupled mutation; $M_{\text{Transcription}}$ denotes asymmetry caused by the transcription-coupled mutation, $R_{\text{Transcription}}$ denotes asymmetry caused by the Repair mechanism during transcription; D_{tRNA} denotes the asymmetry caused by the biased distribution of tRNA; and NS denotes the asymmetry caused by natural selections.

Probably, the major cause of the asymmetry is the transcription-coupled mutation on untranscribed strand, while other causes, such as replication-coupled mutation, biased distribution of tRNAs, or natural selection on codon usages, also contributes more or less to the asymmetry found in GC composition. However, gene density is not one of the causes but only a secondary appearance of the other causes. In some genomes, natural selections are much stronger than these causes of asymmetry, therefore in such genomes natural selections are regarded as the cause of asymmetry.

Until now, the GC skew of *E.coli* can not be explained by the density of genes on the leading strands which accounts for the presence of GC skew in the genomes of *Bacilli*. This finding accounts such presence of GC asymmetry in *E.coli* genome. The asymmetry of GC composition which is globally observed in complete genomic sequences, is the results of superposition between that from DNA replication and that from RNA transcription. The absolute value of fluctuations in GC asymmetry found in *Bacilli* is larger than that of *E.coli*. This shows that in *Bacilli* the contribution from transcription level asymmetry is added to that of replication level asymmetry, while in *E.coli* the contribution from transcription level asymmetry is subtracted from that of replication level.

Conclusions

Assembler and quality control in microbial genome sequencing project

Although it was used to be quoted that the time of genome sequencing had passed and now we are in the post genome era, genome sequencing projects are still necessary to provide basic resource to new areas of science and technology because every research would rely on the accumulated genome information more or less. In addition to that higher quality of such data is much required. However, such projects in coming age should not be conducted as those of the past. Labor-intensive and high-cost projects can not survive today, thus an integrated and compact sequencing project management system to be developed had been waited for.

A software system named MetaGenomeGAMBLER has been developed to meet such requirements of scientists. It provides semi-automated processing of whole genome or cDNA sequencing data. It runs on Windows PCs or Macintosh computers which are the most ordinary tools of the molecular biologists in the world. A researcher who would like to investigate a bacteria genome, now possesses in his hands almost all the tools required to accomplish his research targets.

The accomplishments of MetaGenomeGAMBLER(MGG) and *in silico* Assembler(ISA) are as follows. (1)Since the software tools handle almost every genomic and cDNA raw data, a user can utilize every available data around him. (2)Even a whole genome sequencing data is easily processed in a small PC on the desk. (3)With link to IMC software, most of annotation works are also performed on the same computer. (4)The software tools can also support meta genome projects or cDNA sequencing and clustering projects with clustering algorithms.

Sequence analysis in genome era

Together with the above-mentioned MGG/ISA, *in silico* MolecularCloning(IMC) is also providing all-in-one functionalities to perform almost everything about computer analysis of genes and genomes. With these integrated software tools, most of such analysis and data handling are coming within a reach of ordinary molecular biologists and its students.

IMC can handle from small DNA fragment of 13 nucleotides up to the whole chromosome of human chromosome no.1 whose length amounts to about 250,000,000 bases. Even such huge size data, IMC reads the 250 millions bases data in a few minutes and can view with fast scrolling and zooming on ordinary PC.

BLAST and ClustalW are excellent software tools which are widely used by molecular biologists. However, the user interfaces to these tools are not so easy to be operated by general users. In addition, the increasing data itself is making it difficult to locate. Without exact knowledge of the locations of such biological databases, BLAST and ClustalW could not provide adequate performance which they originally possess. IMC incorporates BLAST and ClustalW in it and with its data searching algorithms IMC also provides facilities to locate exact sites of useful databases.

Another contribution of IMC is that it has revitalized the sequence analysis algorithms of the past decades. These algorithms are still valuable.

***in silico* experiments**

Performing *in silico* cloning requires recording of the end shapes of digested products by restriction enzymes or amplified products by PCR. For this purpose, I introduce a new feature key ***endtype*** and its qualifier ***endtype***, and incorporate them into GenBank/EMBL database annotation convention. Some features on a DNA sequence might be truncated by PCR or digestion by restriction enzymes, therefore the annotations on the truncated features should also be modified. The ambivalent nature of DNA also requires occasional switching to the interested strand from one to another. In addition, I redefine information about the RE recognition sequences, end shapes after digestion and affection of methylation at the sites, to a new feature key and qualifiers. According to these definitions or data descriptions, I have developed a software for *in silico* experiments, and perform a few of typical molecular cloning experiments on computer, and verified that this approach would be effective as a recording tool of a series of experiments as a lab notebook, training tools for beginners to molecular biology, prior simulating tool for time or cost consuming experiments.

In molecular biology experiments, it is important how to describe the functionalities or activities of enzymes and how to use such description. According to the requirements of an *in silico* experiment, one data entry to one enzyme seems to be the best way. Descriptions on the enzymatic functional sites, are still poor in case of most of protein database entries. Therefore, in this stage, I assume that enzyme proteins act as only catalyst instead of multi-functional protein which has residue-specific activities around its amino acid sequence. Further research is necessary to take these activities into consideration for *in silico* experiments.

Tiling array

Exhaustive experiments such as those for DNA array, require further performance for its analysis software. A tiling array for a prokaryote genome has sub-millions order of probes on its array matrix. Until now, there have been few software tools available to handle such huge amount of data. A special edition of IMC is designed and is implemented for such tiling array data analysis. This software currently provides fast scrolling and zooming with accommodation of ten arrays at once. This limitation will be extended by facilitating simultaneous handling of up to 30 arrays in a near future.

Since the data structures of the array manufacturers are not so easy for scientists to understand, IMC solved this problem with a simple manner. Every related data are stored in a single file of GenBank/EMBL format and probe sequences are interpreted as features of the annotation on the genome sequence. In addition, expression profiling data of each array is also incorporated as qualifiers of probe features. By this method, a user should handle only one large-sized file which contains everything he needs to analyze. This software tool also provides traditional results of array analysis by interpreting tiling probe expression data to gene level summery data.

Transcription-coupled GC Skew is found in prokaryotic genomes

As for the purpose of proving usability of the above-mentioned software tools, I had been investigating available genomic and cDNA sequences one by one. When I was viewing the nucleotides composition profiles along one microbial genome, I found a strange coincidence of the transcription direction and asymmetry of GC composition. Such asymmetry had been already known as a genome-scale phenomenon and related to the direction of chromosome replication, however local and transcription level of this phenomenon is yet reported. After checking all the available prokaryotic genomes, low GC-content genomes are likely to show this tendency. Especially remarkable are the genomes of some *Clostridii* and *Fusobacterium nucleatum*. Their AT compositional asymmetries are also correlated with those of GC.

Until now, the GC skew of *E.coli* can not be explained by the density of genes on the leading strands which accounts for the presence of GC skew in the genomes of *Bacilli*. This finding accounts such presence of GC asymmetry in *E.coli* genome. The asymmetry of GC composition which is globally observed in complete genomic sequences, is the results of superposition between that from DNA replication and that from RNA transcription. The absolute value of fluctuations in GC asymmetry found in

Bacilli is larger than that of *E.coli*. This shows that in *Bacilli* the contribution from transcription level asymmetry is added to that of replication level asymmetry, while in *E.coli* the contribution from transcription level asymmetry is subtracted from that of replication level.

Since DNA polymerases might be originated from one common ancestral gene with RNA polymerases, both may cause the resulting asymmetry of GC composition in similar manner with different scale.

Acknowledgements

I wish to express my special thanks to **Professor Naotake Ogasawara** for his orientation in my challenge to the theme of the thesis, to **Professor Shigehiko Kanaya** for his appropriate guidance to my accomplishment, to **Associate Professor Ken Kurokawa** for his valuable suggestions and comments, to **Assistant Professor Md.Altaf-Ul-Amin** for his reading the draft of the thesis and to **Assistant Professor Taku Ohshima** and **Assistant Professor Shu Ishikawa** for their comments and suggestions in improvement of the software tools.

I also thanks to the following scientist for their valuable suggestions and comments for *in silico* MolecularCloning.

Dr. Masanari Kitagawa, Takara Bio
Associate professor Kaoru Nakasone, Kinki University
Dr. Hideto Takami, JAMSTEC
Dr. Yutaka Kawawabayashi, AIST
Associate professor Tohru Suzuki, Gifu University
Professor Masataka Tsuda, Tohoku University

And finally I would like to thanks to my colleagues in *in silico* biology, inc.

Kyotetsu Enai
Hitohiro Saitoh

References

- [1] Abril J.F., Guigo R. and Wiehe T., (2003), gff2aplot: Plotting sequence comparisons, *Bioinformatics*, **19**: 2477-2479.
- [2] Adams M.D. et al., The genome sequence of *Drosophila melanogaster*, *Science*, **287**: 2185-2195.
- [3] Altschul S.F., Gish W., Miller W., Myers E.W. and Lipman D.J., (1990), Basic local alignment search tool., *J. Mol. Biol.* **215**: 403-410.
- [4] Altschul S.F. et al., (1997), Gapped blast and psi-blast: a new generation of protein database search programs, *Nucl. Acids Res.*, **25**: 3389-3402.
- [5] Ansorge W. et al., (1987), Automated DNA sequencing: ultrasensitive detection of fluorescent bands during electrophoresis, *Nucl. Acids Res.*, **15**, 4593-4602.
- [6] Bairoch A., (1989), PROSITE: a dictionary of protein sites and patterns. (In) *EMBL Biocomputing Technical document 4*, EMBL, Heidelberg.
- [7] Ball C.A. et al., (2004), Submission of Microarray Data to Public Repositories, *PLoS Biol.*, **2**: 1276-1277.
- [8] Ball C.A. et al., (2005), The Stanford Microarray Database accommodates additional microarray platforms and data formats, *Nucl. Acids, Res.*, **33**:D580-D582.
- [9] Baran R.H., Ko H. and Jernigan R.W., (2003), Methods for Comparing Sources of Strand Compositional Asymmetry in Microbial Chromosomes, *DNA Res.*, **10**: 85-85.
- [10] Bhaskaran R. and Ponnuswamy P.K., (1988), *Int. J. Pept. Protein Res.*, **32**: 242-255.
- [11] Batzoglou S. et al., (2002), ARACHNE: a whole-genome shotgun assembler, *Genome Res.*, **12**: 177-189.
- [12] Beletskii A. and Bhagwat A.S., (1996), Transcription-induced mutations: increase in C to T mutations in the non-transcribed strand during transcription in *Escherichia coli*, *Proc. Natl. Acad. Sci. U.S.A.*, **93**: 13919-13924.
- [13] Beletskii A., Bhagwat A.S., (1998), Correlation between transcription and C- to T mutations in the non-transcribed DNA strand, *Biol. Chem.*, **379**: 549-551.
- [14] Benson D.A. et al., (2006), GenBank, *Nucl. Acids Res.*, **34**: D16-20.
- [15] Bernardi G., (1993), The vertebrate genome: Isochores and evolution, *Mol. Biol. Evol.*, **10**: 186-204.
- [16] Bernardi G, Olofsson B. and Filipski J., (1985), The Mosaic Genome of Warm-Blooded Vertebrates, *Science*, **228**: 953-958.
- [17] Bertone P. et al., (2004), Global Identification of Human Transcribed Sequences with Genome Tiling Arrays, *Science*, **306**: 2242-2246.

- [18] Blattner F.R., (1997), The Complete Genome Sequence of Escherichia coli K-12, *Science*, **277**: 1453-1462.
- [19] Bonfield J.K., Smith K.F. and Staden R., (1995), A new DNA sequence assembly program, *Nucl. Acids Res.* **23**: 4992-4999.
- [20] Borodovsky M. and McIninch J., (1993), GeneMark: parallel gene recognition for both DNA strands, *Comput. & Chem.*, **17**: 123-133.
- [21] Brazma A et al., (2001), Minimum information about a microarray experiment (MIAME)-toward standards for microarray data, *Nat. Genet.*, **29**: 365-371.
- [22] Brazma A. et al., (2003), ArrayExpress-a public repository for microarray gene expression data at the EBI, *Nucl. Acids Res.*, **31**: 68-71.
- [23] Brown N.P. et al., (1998), Frame: detection of genomic sequencing errors, *Bioinformatics*, **14**: 367-371.
- [24] Burge C. and Karlin S., (1997), Prediction of complete gene structures in human genomic DNA, *J.Mol.Biol.*, **268**: 78-94.
- [25] Burge C. and Karlin S., (1998), Finding the genes in genomic DNA, *Curr. Opin. Struct. Biol.*, **8**:346-354.
- [26] Butler B.A., (1998), Sequence analysis using GCG, *Methods Biochem. Anal.*, **14**: 452-457.
- [27] Campagna D. et al., (2005), RAP: a new computer program for *de novo* identification of repeated sequences in whole genomes, *Bioinformatics*, **21**: 582-588.
- [28] Chait B.T., (1996), Trawling for proteins in the post-genome era, *Nat. Biotechnol.*, **14**: 1579-1583.
- [29] Chee M. et al., (1996), Accessing genetic information with high-density DNA arrays, *Science*, **274**: 610-614.
- [30] Chen D.-t., Lin S.-h. and Soong S.-j, (2004), Gene selection for oligonucleotide array: an approach using PM probe level data, *Bioinformatics*, **20**: 854-862.
- [31] Chi B., deLeeuw R.J., Coe, B.P., MacAulay C. and Lam W.L., (2004), SeeGH – A software tool for visualization of whole genome array comparative genomic hybridization data, *BMC Bioinformatics*, **5**: 13
- [32] Chou P.Y. and Fasman G.D., (1978), Prediction of the secondary structure of proteins from their amino acid sequence. *Adv. Enzymol.* **47**: 45-148.
- [33] Churchill G.A., (2002), Fundamentals of experimental design for cDNA microarrays, *Nature Genet.*, **32**: 490-495.
- [34] Ciora T., Deneffe P. and Mayaux J.-F., 1991, Rapid one-step automated sequencing reaction for 16 DNA samples using Taq polymerase and fluorescent primers, *Nucl.*

Acids, Res., **19**, 188.

- [35] Cochrane G. et al., (2006), EMBL Nucleotide Sequence Database: development in 2005, *Nucl. Acids Res.*, **34**:D10-D15.
- [36] Dayhoff M.O., Schwartz R.M. and Orcutt B.C., (1978), (in) *Atlas of Protein Sequence and Structure*, Vol.5, Suppl. 3.
- [37] Dear S. and Staden R. (1992), A standard file format for data from DNA sequencing instruments, *DNA Sequence*, **3**: 107-110.
- [38] Delcher A.L., Douglas H., Kasif S., White O. and Salzberg S., (1999), Improved microbial gene identification with GLIMMER, *Nucl. Acids Res.*, **27**: 4636-4641.
- [39] Delcher A.L. et al., (1999), Alignment of whole genomes, *Nucl. Acids Res.*, **27**: 2369-2376.
- [40] DeRisi J.L. et al., (1996), Use of a cDNA microarray to analyze gene expression patterns in human cancer, *Nat. Genet.*, **14**: 457-460.
- [41] Devereux J., Haeberli P. and Smithies O., (1984), A comprehensive set of sequence analysis programs for the VAX, *Nucl. Acids Res.*, **12**: 397-395.
- [42] Dolz R., (1994), GCG: assembly of sequences into new sequence constructs, *Methods Mol. Biol.*, **24**: 57-63.
- [43] Dolz R., (1994), GCG: displaying restriction sites and possible translations in a DNA sequence, *Methods Mol. Biol.*, **24**: 47-55.
- [44] Dolz R., (1994), GCG: drawing linear restriction maps, *Methods Mol. Biol.*, **24**: 25-33.
- [45] Dolz R., (1994), GCG: preparing sequence data for publication, *Methods Mol. Biol.*, **24**: 167-171.
- [46] Dolz R., (1994), GCG: analysis of protein sequences, *Methods Mol. Biol.*, **24**: 143-157.
- [47] Dong X., Stothard P., Forsythe I.J. and Wishart D.S., (2004), PlasMapper: a web server for drawing and auto-annotating plasmid maps, *Nucl. Acids Res.*, **32**: W660-W664.
- [48] Duret L. and Abdeddaim S., (2000), Multiple Alignments for structural, functional, or Phylogenetic analysis, (in) *Bioinformatics, Sequence, structure and databanks*, ed. Higgins D. and Taylor W., p74, Oxford University Press.
- [49] Eisenberg D., Schwarz E., Komaromy M. and Wall R., (1984), Analysis of membrane and surface protein sequences with the hydrophobic moment plot., *J. Mol. Biol.* **179**: 125-142.
- [50] Eisenberg D., Weiss R.M. and Terwilliger T.C., (1984), The hydrophobic moment detects periodicity in protein hydrophobicity, *Proc. Natl. Acad. Sci. U.S.A.* **81**: 140-144.

- [51] Ewing B et al., (1998A), Base-calling of automated sequencer traces using phred. I. Accuracy assessment. *Genome Res.*, **8**: 175-185.
- [52] Ewing B. and Green P., (1998B), Base-calling of automated sequencer traces using phred. II. Error probabilities. *Genome Res.*, **8**: 186-194.
- [53] Felsenstein J., (1981), Evolutionary trees from DNA sequences: a maximum likelihood approach, *J. Mol. Evol.*, **17**: 368-376.
- [54] Felsenstein J., (1988), Phylogenies from molecular sequences: inference and reliability, *Annu. Rev. Genet.*, **22**: 521-565.
- [55] Felsenstein J., (1997), An alternating least squares approach to phylogenies from pairwise distances, *Syst. Biol.*, **46**: 101-111.
- [56] Feng D.F. and Doolittle R.F., (1990), Progressive alignment and phylogenetic tree construction of protein sequences, *Methods Enzymol.*, **183**: 375-387.
- [57] Fickett J.W., (1982), Recognition of Protein Coding Regions in DNA Sequences, *Nucl. Acids Res.*, **10**: 5303-18.
- [58] Fleischmann R.D. et al., (2005), Whole-genome random sequencing and assembly of *Haemophilus influenzae* Rd., *Science*, **269**: 496-512.
- [59] Filipinski J., (1987), Correlation between molecular clock ticking, codon usage, fidelity of DNA repair, chromosome banding and chromatin compactness in germline cells, *FEBS Lett.*, **217**: 184-187.
- [60] Fodor S.P.A. et al., (1993), Multiplexed biochemical assays with biological chips, *Nature*, **364**, 555-556.
- [61] Francino M.P., Chao L., Riley M.A. and Ochman H., (1996), Asymmetries Generated by Transcription-Coupled Repair in Enterobacterial Genes, *Science*, **272**: 107-109.
- [62] Francino M.P. and Ochman H., (2001), Deamination as the basis of strand-asymmetric evolution in transcribed *Escherichia coli*, *Mol. Biol. Evol.*, **18**: 1147-1150.
- [63] Frank A.C., Lobry J.R., (1999), Asymmetric substitution patterns: a review of possible underlying mutational or selective mechanisms, *Gene*, **238**: 65-77.
- [64] Fraser C.M. et al., (1995), The minimal gene complement of *Mycoplasma genitalium*, *Science*, **270**: 397-403.
- [65] Garnier J., Osguthorpe D.J. and Robson B., (1978), Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. *J. Mol. Biol.*, **120**: 97-120.
- [66] Garnier J. and Robson B., (1989), The GOR method for predicting secondary structure in proteins. (In) *Prediction of protein structure and the principles of protein conformation*, Fasman G.D., Ed., 417-465, Plenum Press, New York.

- [67] The Gene Ontology Consortium, (2000), Gene Ontology: tool for the unification of biology, *Nat. Genet.*, **25**: 25-29.
- [68] Gibbs A.J. and McIntyre G.A., (1970), The diagram, a method for comparing sequences, *Eur. J. Biochem.*, **16**: 1-11.
- [69] Gibson R. and Smith D.R., (2003), Genome visualization made fast and simple, *Bioinformatics*, **19**: 1449-1450.
- [70] Gleeson T.J. and Staden R., (1991), An X windows and UNIX implementation for sequence analysis package, *Comput. Applic. Biosci.*, **7**: 398.
- [71] Green P., (1997), Against a Whole-Genome Shotgun, *Genome Res.*, **7**: 418-421.
- [72] Green P. and Brent E., (1999A), PHRED Document Version: 0.990722.
- [73] Green P., (1999B), DOCUMENTATION FOR PHRAP AND CROSS_MATCH (VERSION 0.990319).
- [74] Gribskov M. and Devereux J., (1990), (*in*) *Sequence Analysis Primer*, W.H. Freeman and Company.
- [75] Grigoriev A., (1998), Analyzing genomes with cumulative skew diagrams, *Nucl. Acids, Res.*, **26**:2286-2290.
- [76] Grigoriev A., (1999), Strand-specific compositional asymmetries in double-stranded DNA viruses, *Virus Res.*, **60**:1-19.
- [77] Henikoff S. and Henikoff J.G., (1991), Automatic assembly of protein blocks for database searching., *Nucl. Acids Res.* **19**: 6565-6572.
- [78] Henikoff S. and Henikoff J.G., (1992), Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. U.S.A.*, **89**: 10915-10919.
- [79] Higgins D.G. and Sharp P.M., (1988), CLUSTAL: a package for performing multiple sequence alignment on a microcomputer, *Gene*, **73**: 237-244.
- [80] Holton T.A. and Graham M.W., (1991), A simple and efficient method for direct cloning of PCR products using ddT-tailed vectors, *Nucl. Acids Res.*, **19**: 1156.
- [81] Hopp T.P. and Woods K.R., (1981), Prediction of Protein Antigenic Determinants from Amino Acid Sequences, *Proc. Natl. Acad. Sci. U.S.A.*, **78**: 3824-3828.
- [82] Hu Z. and Willsky G.R., (2006), Utilization of two sample t-test statistics from redundant probe sets to evaluate different probe set algorithms in GeneChip studies, *BMC Bioinformatics*, **7**: 12.
- [83] Huang Y. and Zhang L., (2004), Rapid and sensitive dot-matrix methods for genome analysis, *Bioinformatics*, **20**: 460-466.
- [84] Iazzetti G., Santini G., Rau M., Bucci E. and Calogero A., (1998), VIRTLAB: a virtual molecular biology laboratory, *Bioinformatics*, **14**: 815-816.

- [85] Ikemura T. and Wada K., (1991), Evident diversity of codon usage patterns of human genes with respect to chromosome banding patterns and chromosome numbers; relation between nucleotide sequence data and cytogenetic data, *Nucl. Acids Res.*, **19**: 4333-4339.
- [86] Iris F.J.M., (1994), Optimized Methods for Large-scale Shotgun DNA Sequencing in Alu-rich Genomic Regions, (in) *Automatic DNA Sequencing*, Academic Press: 199-209.
- [87] Ishkanian A. et al., (2004), A tiling resolution DNA microarray with complete coverage of the human genome, *Nature Genet.*, **36**: 299-303.
- [88] Iyer V.R. et al., (2001), Genomic binding sites of the yeast cell-cycle transcription factors SBF and MBF., *Nature*, **409**: 533-538.
- [89] Janin J., (1979), Surface and inside volumes in globular proteins, *Nature*, **277**: 491-492.
- [90] Ji H. and Wong W.H., (2005), TileMap: create chromosome map of tiling array hybridizations, *Bioinformatics*, **21**: 3629-3636.
- [91] Junier T. and Pagni M., (2000), Dotlet: diagonal plots in a Web browser, *Bioinformatics*, **16**: 178-179.
- [92] Jurka J., (1994), Approaches to Identification and Analysis of Interspersed Repetitive DNA Sequences, (in) *Automatic DNA Sequencing*, Academic Press: 294-298.
- [93] Kampa D. et al., (2004), Novel RNAs Identified From an In-Depth Analysis of the Transcriptome of Human Chromosomes 21 and 22, *Genome Res.*, **14**: 331-342.
- [94] Kapatral V. et al., (2002), Genome Sequence and Analysis of the Oral Bacterium *Fusobacterium nucleatum* Strain ATCC 25586, *J. Bacteriol.*, **184**: 2004-2018.
- [95] Kapranov P., Cawley S.E., Drenkow J., Bekiranov S., Strausberg R.L., Fodor S.P.A. and Gingeras T.R., (2002), Large-Scale Transcriptional Activity in Chromosomes 21 and 22, *Science*, **296**: 916-919.
- [96] Karlin S., Burge C. and Campbell A.M., (1992), Statistical analyses of counts and distributions of restriction sites in DNA sequences. *Nucl. Acids Res.*, **20**: 1363-1370.
- [97] Karlin S. Campbell A.M. and Mrazek J., (1998), Comparative DNA analysis across diverse genomes, *Annu. Rev. Genet.*, **32**: 185-225.
- [98] Karplus P.A. and Schulz G.E., (1989), Substrate binding and catalysis by glutathione reductase as derived from refined enzyme: substrate crystal structures at 2 Å resolution, *J. Mol. Biol.*, **5**: 163-180.
- [99] Kerkhoven R., van Enckevort F.H.J., Boekhorst J., Molenaar D. and Siezen R., (2004), Visualization for genomics: the Microbial Genome Viewer, *Bioinformatics*, **20**: 1812-1814.

- [100] Kohonen T., (1984), Self-Organization and Associative Memory, *Springer Series in Information Sciences*, 8.
- [101] Kunst F., Ogasawara N. et al., The complete genome sequence of the gram-positive bacterium *Bacillus subtilis*, *Nature*, **390**: 249-256.
- [102] Kyte J. and Doolittle R.F., (1982), A simple method for displaying the hydropathic character of a protein., *J. Mol. Biol.*, **157**: 105-132.
- [103] Lee L.G. et al., (1991), DNA sequencing with dye-labeled terminators and T7 DNA polymerase: effect of dyes and dNTPs on incorporation of dye-terminators and probability analysis of termination fragments., *Nucl. Acids Res.*, **20**: 2471-2483.
- [104] Lee Y. et al., (2002), Cross-Reference Eukaryotic Genomes: TIGR Orthologous Gene Alignments (TOGA), *Genome Res.*, **12**: 493-502.
- [105] Lefebvre A., Lecroq T., Dauchel H. and Alexandre J., (2003), FORRepeats: detects repeats on entire chromosomes and between genomes, *Bioinformatics*, **19**: 319-326.
- [106] Lexa M., Horak J. and Brzobohaty B., (2001), Virtual PCR, *Bioinformatics*, **17**: 192-193.
- [107] Li C. and Wong W.H., (2000), Model-based analysis of oligonucleotide arrays: Expression index computation and outlier detection, *Proc. Natl. Acad. Sci. U.S.A.*, **98**: 31-36.
- [108] Li J., Spletter L. and Johnson J.A., (2005), Dissecting tBHQ induced ARE-driven gene expression through long and short oligonucleotide arrays, *Physiol. Genomics*, **21**: 43-58.
- [109] Lipshutz R.J. et al., (1999), High density synthetic oligonucleotides arrays, *Nature Genet.*, **21**: 20-24.
- [110] Liu W.-m. et al., (2002), Analysis of high density expression microarrays with signed-rank call algorithms, *Bioinformatics*, **18**: 1593-1599.
- [111] Lobry J.R., (1996), Asymmetric Substitution Patterns in the Two DNA Strands of Bacteria, *Mol. Biol. Evol.*, **13**: 660-665.
- [112] Lobry J.R., (1996), Origin of Replication of *Mycoplasma genitalium*, *Science*, **272**: 745-746.
- [113] Lobry J.R. and Sueoka N., (2002), Asymmetric directional mutation pressures in bacteria, *Genome Biol.*, **3** (10).
- [114] Lobry J.R. and Louarn J.-M., (2003), Polarisation of prokaryotic chromosomes, *Curr. Opinion*, **6**: 101-108.
- [115] Lukashin A. and Borodovsky M., GeneMark.hmm: new solutions for gene finding, (1998), *Nucl. Acids Res.*, **26**: 1107-1115.
- [116] Lupas A.N., van Dyke M. and Stock J., (1991), Predicting coiled coils from

- protein sequences. *Science*, **252**: 1162-1164.
- [117] Luckey J.A. et al., (1990), High speed DNA sequencing by capillary electrophoresis, *Nucl. Acids Res.*, **18**: 4417-4421.
- [118] Lowe T.M. and Eddy S.R., (1997), tRNAscan-SE a Program for improved detection of transfer RNA genes in genomic sequence, *Nucl. Acids Res.*, **25**:955-964.
- [119] Mackiewicz P., Gierlik A., Kowalczyk M., Dudek M.R. and Cebrat S., (1999), How does replication-associated mutational pressure influence amino acid composition of protein, *Genome Res.*, **9**: 409-411.
- [120] Marchuk D., Drumm M., Saulino A., and Collins F., (1991), Construction of T-vectors, a rapid and general system for direct cloning of unmodified PCR products, *Nucl. Acids Res.*, **11**: 1154.
- [121] Mead D.A., Pey N.K., Herrstadt C., Marcil R.A. and Smith L.M., A universal method for the direct cloning of PCR amplified nucleic acid, (1991), *Bio Technol.*, **9**: 657-663.
- [122] Menten B, et al., (2005), arrayCGHbase: an analysis platform for comparative genomic hybridization microarrays, *BMC Bioinformatics*, **6**: 124.
- [123] Miller W., (2001), Comparison of genomic DNA sequences: solved and unsolved problems, *Bioinformatics*, **17**: 391-397.
- [124] Mockler T.C. and Ecker J.R., (2005), Applications of DNA tiling arrays for whole-genome analysis, *Genomics*, **85**: 1-15
- [125] Mrazek J. and Karlin S., (1998), Strand compositional asymmetry in bacterial and large viral genomes, *Proc. Natl. Acad. Sci. U.S.A.*, **95**: 3720-3725.
- [126] Myers E.W. et al., (2000), A Whole-Genome Assembly of Drosophila, *Science*, **287**: 2196-2204.
- [127] Nakamura Y et al., (2000), Codon usage tabulated from international DNA sequence databases: status for the year 2000, *Nucl. Acids Res.*, **28**: 292.
- [128] Nishio Y. et al., (2003), Comparative Complete Genome Sequence Analysis of the Amino Acid Replacements Responsible for the Thermostability of *Corynebacterium efficiens*, *Genome Res.*, **13**: 1572-1579.
- [129] Nishizawa M. and Nishizawa K., (1998), Biased Usages of Arginines and Lysines in Proteins Are Correlated with Local-Scale Fluctuations of the G + C Content of DNA Sequences, *J. Mol. Evol.*, **47**: 385-393.
- [130] Ouzounis C et al., (1996), Computational comparisons of model genomes, *Trends Biotechnol.*, **14**: 280-285.
- [131] Overbeek R. et al., (2000), WIT: integrated system for high-throughput genome sequence analysis and metabolic reconstruction, *Nucl. Acids Res.*, **28**: 123-125.
- [132] Parkinson H., et al., (2005), ArrayExpress-a public repository for microarray gene

- expression data at EBI, *Nucl. Acids Res.*, **33**: D553-D555.
- [133] Peterson M.G., (1988), DNA sequencing using Taq polymerase, *Nucl. Acids Res.*, **16**: 10915.
- [134] Pearson W.R. and Lipman D.J., (1988), Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci. U.S.A.*, **85**: 2444-2448.
- [135] Pop M. et al., (2002), Genome Sequence Assembly: Algorithms and Issues, *IEEE Comput.*, :47-54.
- [136] Quackenbush J., (2002), Microarray data normalization and transformation, *Nature Genet.*, **32**: 496-501.
- [137] Quandt K., Frech K., Karas H., Wingender E. and Werner T., (1995), MatInd and MatInspector: new fast and versatile tools for detection of consensus matches in nucleotide sequence data. *Nucl. Acids Res.* **23**: 4878-4884.
- [138] Raskin. D.M., Seshadri R., Pukatzki S.U., and Mekalanos J.J., (2006), Bacterial genomics and pathogen evolution, *Cell*, **124**: 703-714.
- [139] Rayner T.F., et al., (2006), A simple spreadsheet- based, MIAME-supportive format for microarray data, *submitted to BMC Bioinformatics*
- [140] de Reynies A et al., (2006), Comparison of the latest commercial short and long oligonucleotide microarray technologies, *BMC Genomics*, **7**: 51.
- [141] Ren B., et al., (2000), Genome-Wide Location and Function of DNA Binding Proteins, *Science*, **290**: 2306-2309.
- [142] Riley M., (1993), Functions of the gene products of *Escherichia coli*, *Microbiol. Rev.*, **57**: 862-952.
- [143] Roberts R.J., Vincze T., Posfai J. and Macelis D., (2005), REBASE-restriction enzymes and DNA methyltransferase, *Nucl. Acids Res.*, **33**: D230-D232.
- [144] Rutherford K. et al., (2000), Artemis: sequence visualization and annotation, *Bioinformatics*, **16**: 944-945.
- [145] Saccone S. et al., (1993), Correlations Between Isochores and Chromosomal Bands in the Human Genome, *Proc. Natl. Acad. Sci. U.S.A.*, **90**: 11929-11933.
- [146] Saitou N. and Nei M., (1987), Neighbor-joining Method: A New Method for Reconstructing Phylogenetic Trees, *Mol. Biol. Evol.*, **4**: 406.
- [147] Sakiyama T. et al., (2000), An Automated System for Genome Analysis to Support Microbial Whole-genome Shotgun Sequencing, *Biosci, Biotechnol. Biochem.*, **64**: 670-673
- [148] Salzberg S.L., Delcher A.L., Kasif S. and White O., (1998), Microbial gene identification using interpolated Markov models, *Nucl. Acids Res.*, **26**: 544-548.
- [149] Sato N. and Ehira S., (2003), GenoMap, a circular genome data viewer,

- Bioinformatics*, **19**: 1583-1584.
- [150] Sayle R.A. and Milner-White E.J.,(1995), RasMol: Biomolecular graphics for all, *Trends Biochem. Sci.*, **20**: 374-376.
- [151] Shavilik, J.W., (1994), Finding Frameshift Errors in Anonymous DNA, in *Automatic DNA Sequencing*, Academic Press: 280-288.
- [152] Selinger D.W., (2000), RNA expression analysis using a 30 base pair resolution Escherichia coli genome array, *Nat. Biotechnol.*, **18**: 1262-1268.
- [153] Shimizu T., et al., (2002) Complete genome sequence of *Clostridium perfringens*, an anaerobic flesh-eater, *Proc. Natl. Acad. Sci. U.S.A.*, **99**: 996-1001.
- [154] Shoemaker D.D. et al., (2001), Experimental annotation of the human genome using microarray technology, *Nature*, **409**: 922-927.
- [155] Sneath H.A. and Sokal R.R., (1973), (*in*) *Numerical taxonomy*, W.H.Freeman, San Francisco.
- [156] Sokal R.R. and Michener C.D., (1958), A statistical method for evaluating systematic relationships, *University of Kansas Scientific Bulletin*, **28**: 1409-1438.
- [157] Sonnhammer E.L.L. and Durbin R., (1995), A dot matrix program with dynamic threshold control suited for genomic DNA and Protein sequence analysis, *Gene*, **167**: GC1-10.
- [158] Spellman P.T. et al., (2002), Design and implementation of microarray gene expression markup language (MAGE-ML), *Genome Biol.*,**3**: 0046.1-9.
- [159] Staden R., (1996), The Staden Sequence Analysis Package, *Mol. Biotechnol.*,**5**: 233-241.
- [160] Staden R. and McLachlan A.D., (1982), Codon preference and its use in identifying protein coding regions in long DNA sequences. *Nucl. Acids Res.* **10**: 141-156.
- [161] Stoeckert C.J. et al., (2002), Microarray databases: standards and ontologies, *Nature Genet.*, **32**: 469-473.
- [162] Stothard P. and Wishart D.S., (2005), Circular genome visualization and exploration using CGView, *Bioinformatics*, **21**: 537-539.
- [163] Sueoka N., (1988), Directional Mutation Pressure and Neutral Molecular Evolution, *Proc. Nat. Acad. Sci. U.S.A.*, **85**: 2653-2657.
- [164] Sutton G.G. et al., (1995), TIGR Assembler: A New Tool for Assembling Large Shotgun Sequencing Projects, *Genome Sci. Technol.*, **1**: 9-19.
- [165] Swofford D.L. and Olsen G.J., (1990), (*In*) *Molecular systematics* (eds. Hillis D.M. and Moritz C.), Sunderland, Massachusetts, pp.411-501.
- [166] Takami H. et al., (2000), Complete genome sequence of the alkaliphilic *Bacillus*

- halodurans* and genome sequence comparison with *Bacillus subtilis*, *Nucl. Acids Res.*, **21**: 4317-4331.
- [167] Tatusov, R.L. et al., (2001), The COG database: new developments in phylogenetic classification of proteins from complete genomes, *Nucl. Acids Res.*, **29**: 22-28.
- [168] Thompson J.D., Higgins D.G. and Gibson T.J., (1994), CLUSTALW: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties and weight matrix choice, *Nucl. Acids Res.*, **22**: 4673-4680.
- [169] Touchon M., Arneodo A., d'Aubenton-Carafa Y. and Thermes C., (2004), Transcription-coupled and splicing-coupled strand asymmetries in eukaryotic genomes, *Nucl. Acids Res.*, **23**: 4969-4978.
- [170] Touchon M. et al., (2005), Replication-associated strand asymmetries in mammalian genomes: Toward detection of replication origins, *Proc. Natl. Acad. Sci. U.S.A.*, **102**: 9836-9841.
- [171] Trovato R. et al., (1999), A Lysine-to-Arginine Change Found in Natural Alleles of the Human T-Cell Lymphotropic/Leukemia Virus Type 1 p12 Protein Greatly Influences Its Stability, *J. Virol.*, **73**: 6460-6467.
- [172] Uchiyama I., (2003), MBGD: microbial genome database for comparative analysis, *Nucl. Acids Res.*, **31**: 58-62.
- [173] Wang Y., Geer L.Y., Chappey C., Kans J.A. and Bryant S.H., (2000), Cn3D: sequence and structure views for Entrez, *Trends Biochem. Sci.*, **25**:3 00-302.
- [174] Wang Y., et al., (2006), Large scale real-time PCR validation on gene expression measurements from two commercial long-oligonucleotide microarrays, *BMC Genomics*, **7**: 59.
- [175] Waterman, M.S., (1995), Sequence Assembly, (*in*) *Introduction to Computational Biology*, Chapman & Hall: 135-160.
- [176] Weber J.L. and Myers E.W., (1997), Human Whole-Genome Shotgun Sequencing, *Genome Res.*, **7**: 401-409.
- [177] Womble D.D., (2000), GCG: The Wisconsin Package of sequence analysis programs, *Methods Mol Biol.*, **132**: 3-22.
- [178] Wootton J.C. and Federhen S., (1993), Statistics of local complexity in amino acid sequences and sequence databases. *Comput. Chem.* **17**: 149-163.
- [179] Wu C. I. and Maeda N., (1987), Inequality in mutation rates of the two strands of DNA, *Nature*, **327**: 169-170.
- [180] Wu M. et al., (2004), Phylogenomics of the reproductive parasite *Wolbachia pipientis* wMel: a streamlined genome overrun by mobile genetic elements, *PLoS Biol.*, **2**: 327-341.

- [181] Yamada et al., (2003), Empirical Analysis of Transcriptional Activity in the Arabidopsis Genome, *Science*, **302**: 842-846.
- [182] Yang Y.H. et al., (2002), Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation, *Nucl. Acids Res.*, **30**: e15.
- [183] Yuan P.Y., (1998), Towards detection of orthologues in sequence databases, *Bioinformatics*, **14**: 285-289.
- [184] Zhang Z., Schwartz S., Wagner L. and Miller W., (2000), A greedy algorithm for aligning DNA sequences, *J. Comput. Biol.*, **7**: 203-214.
- [185] Zimmerman J.M., Eliezer N. and Simha R., (1968), The characterization of amino acid sequences in proteins by statistical methods, *J. Theor. Biol.*, **21**: 170-201.
- [186] -, (2005), Affymetrix GeneChip Operating Software With Autoloader, Version 1.4,


















Appendix

A. List of functions of *in silico* MolecularCloning

















B. Molecular Cloning protocols *in silico*

C. List of microbial genomes with the local GC skew

Appendix A

	Menus	Operation	Main Function	Sub Operation	Sub Functions	Standard Edition	Genomics Edition	Array Edition		
File Handling	Read Sequence(s)		Read Genbank, EMBL, FastA, Plain Text Format files and Show Features onto Feature Map	Select File(s) from File Chooser	Read Genbank Format Sequence File(s)	+	+	+		
					Read EMBL Format Sequence File(s)	+	+	+		
					Read FastA Format Sequence File(s)	+	+	+		
					Read Simple Text Format Sequence File(s)	+	+	+		
	Read from		Read DNA sequence with copy and paste operation		Direct Key In	Input Nucleotides from Keyboard	+	+	+	
					Push ctrl-v Key	Paste from the Clipboard	+	+	+	
	Save Sequence		Save DNA sequence with Genbank, EMBL and FastA format				+	+	+	
	Save Sequence as		Save as DNA sequence with Genbank, EMBL and FastA format file	Specify a File Name in the File Chooser	Set Qualifiers to be Excluded from Save File and Specify a File Name to be written in the File Chooser	Save as Genbank Format File	+	+	+	
						Save as EMBL Format File	+	+	+	
						Save as Genbank/EMBL File without Specified Qualifiers	+	+	+	
	Print & Print Settings		Print Feature Map and Set Parameters for Printing		Click "Page Setup"	Set Parameters for Printing	+	+	+	
					Click "Print"	Print the Feature Map	+	+	+	
					Click "PDF"	Write PDF of the Feature Map	+	+	+	
	Read Reference(s)		Read Reference DNA Sequence(s) and Show Features onto Reference Map		Select from the File Chooser		+	+		
					Click Icon in the Reference Tube	Change the visible DNA sequence	+			
	Import Feature		Import feature files with CSV format		Select "CSV"	In CSV format	+	+	+	
					Select "Glimmer2"	Features created by Glimmer2	+	+	+	
					Select "Xana"	Features created by tRNAScanSE	+	+	+	
					Select "tRNAScanSE"	Features created by XanaGenome				
	Export Feature		Export features and sequences in CSV format		Click "CSV"	Export Features in CSV Format	+	+	+	
					Check "With Sequence and Click "CSV"	Export Features with Sequence in CSV Format	+	+	+	
	Write FastA Sequence		Write FastA format files from Current Sequence		Click "Set"	Write FastA Format Files with Forward Strand Sequence	+	+	+	
					Check "Complement" and Click "Set"	Write FastA Format Files with Reverse Strand Sequence	+	+	+	
	Create Database		Create databases for homology search			Create Amino Acid Database	+	+	+	
						Create Nucleotide Database	+	+	+	
	Expand Multi Format File		Expand multifasta format file into taxonomy tree structure		Select from Options	Expand by Definition Name	+	+	+	
					Select from Options	Expand by Locus Name	+	+	+	
	Access Internet		Access registrated data sites directly				+	+	+	
Download Commands		Download commands from internet sites		Select "BLAST"	Download and Install BLAST Program	+	+	+		
				Select "ClustalW"	Download and Install ClustalW Program	+	+	+		
				Select "NetBLAST"	Download and Install	+	+	+		
Update IMC		Download and install the latest version of IMC				+	+	+		
Delete Marked Features Completely		Completely Delete all the Features with Deleted Status				+	+	+		
				Right Button on Feature Map	Delete Completely all the Features with Deleted Status in the Shaded Region	+	+	+		
				Right Button on Feature	Delete Completely this Feature	+	+	+		
Restore from deleted		Restore deleted features				+	+	+		
				Right Button on Feature Map	Restore from Deleted Status all the Features in the Shaded	+	+	+		
				Right Button on Feature		+	+	+		
Delete Features		Delete features of specified Feature Keys or for specified region			Delete Features of Specified Keys	+	+	+		
					Delete Features in Specified Regions	+	+	+		

Appendix A

	Menus	Operation	Main Function	Sub Operation	Sub Functions	Standard Edition	Genomics Edition	Array Edition
Feature Operation on the Feature Map and the Current Reaction Tube	Delete Features		Delete Feature(s)	Right Button on Feature Map	Delete all the Features in the Shaded Region	+	+	+
				Right Button on Feature	Delete This Feature	+	+	+
	Duplicate Feature		Duplicate THIS Feature	Right Button on Feature		+	+	+
	Change Feature(s)		Change Feature Key of the Feature(s)	Right Button on the Feature Map	Change Feature Key of the Features in the Shaded Region	+	+	+
				Right Button on the Feature		+	+	+
	Browse Genbank/EMBL		Browse current Genbank/EMBL files contents			+	+	+
	Browse Sequence		Browse current DNA sequence			+	+	+
	Change Strand		Change this Feature on to the Complementary Strand			+	+	+
	Set Qualifier to Feature		Set Qualifier to this Feature	Right Button on the Feature and Select "Color"	Set Color to this Feature	+	+	+
				Right Button on the Feature and Select "Journal"	Set Link to Journal File to this Feature	+	+	+
				Right Button on the Feature and Select "Label"	Change Labelling Status to this Feature	+	+	+
				Right Button on the Feature and Select "Feature Map"	Change Visibility Status to this Feature	+	+	+
				Right Button on the Feature and Select "3D Structure"	Set Link to 3D-Structure File to this Feature	+	+	+
	Set Qualiifer to Region		Set Qualifier to the Features in the Selected(Shaded) Region	Right Button on the Feature Map and Select "Color"	Set Color to this Feature	+	+	+
				Right Button on the Feature Map and Select "Label"	Change Labelling Status to this Feature	+	+	+
				Right Button on the Feature Map and Select "Feature Map"	Change Visibility Status to this Feature	+	+	+
	Set Annotation Grade to the Feature		Set Annotation grade to the Feature(s)	Select "Fixed"	Set "Fixed" Annotation to this Feature	+	+	+
				Select "Adopted"	Set "Adopted" Annotation to this Feature	+	+	+
				Select "Annotated"	Set "Annotated" Annotation to this Feature	+	+	+
					Set Any Customized Annotation Category to this Feature	+	+	+
	Remove DNA Sample		Remove the Selected DNA Sample(s) from the Feature Map	Click Icons in the Reaction Tube	Remove the Selected DNA Sample from the Feature Map	+	+	+
				Click in the Reaction Tube	Remove the Selected DNA Samples from the Feature Map	+	+	+
	Scroll Map		Scroll Reference Map to the Left			+	+	
			Scroll Reference Map to the Right			+	+	
Homology Search			Homology Search from a Feature on the Referece Map against Features on the Feature Map and All the Features of Different Sequences on the Reference Map	Click on the Features and Select "Show Homology N.A"	Homology Search by Nucleotides	+	+	
				Click on the Features and Select "Show Homology A.A"	Homology Search by Amino Acids	+	+	
Remove DNA Sample			Remove the Selected DNA Sample(s) from the Reference Map	Select One Icon and Click "Remove"	Remove One Sequence from Reference Map	+	+	
				Select Icons and Click "Remove"	Remove Sequences from Reference Map	+	+	
Change Current DNA			Change Currenty Visible DNA Sequence			+		
Move DNA Sample		Move DNA Sample to New Position	Drag Icon to New Position and Drop there			+		
Select Sequence		Select Nucleotides or Amino Acids		Select Nucleotides from the Upper Sequence Area	+	+	+	
				Select Nucleotides from the Description Window	+	+	+	
				Select Amino Acids from the Description Window	+	+	+	
				Copy Selected Sequence on the Upper Sequence Area to	+	+	+	
Copy Sequence		Copy Sequence to Clipboard		Copy Selected Sequence on the Future Map to Clipboard	+	+	+	
					+	+	+	









Appendix A

	Menus	Operation	Main Function	Sub Operation	Sub Functions	Standard Edition	Genomics Edition	Array Edition	
Sequence Operation	Paste Sequence		Paste Sequence from the Clipboard			+	+	+	
				cntl-v	Paste the Sequence in the Clipboard to this Area	+	+	+	
	Save Sequence as a Primer		Save the Selected Nucleotides on the Upper Sequence Area as a New Primer Sequence			+	+	+	
	Save Sequence as a Feature		Save the Selected Nucleotides on the Feature Map as a New Primer Sequence			+	+	+	
	Delete Nucleotides		Delete the Selected Nucleotides			+	+	+	
	Insert Nucleotides		Insert New Nucleotides at either side of the Selected Nucleotide	Right Button on Upper Sequence Area		+	+	+	
	Replace Nucleotides		Replace the selected nucleotides with new ones			+	+	+	
Map Region Operation	Clear Selection		Clear the Nucleotides from "Selected" Status	Click "Clear Selection" on the Upper Sequence Area		+	+	+	
	Select Region		Select a Region by Dragging Mouse on the Feature Map			+	+	+	
Labelling and Coloring	Change Coloring		Show feature with default color for feature key or individually specified color			+	+	+	
	Switch for function classification code coloring		Show feature with color by classification code			+	+	+	
	Switch for EC Number coloring		Show feature with color by EC number			+	+	+	
	Switch for Feature category coloring		Show feature with coloring by category			+	+	+	
	Switch for Homology score coloring		Show feature with coloring by homology score			+	+	+	
	Feature Key		Create and register new feature keys			+	+	+	
Searching	Search Feature Key		Search specified feature key from current features			+	+	+	
	Search by Keywords		Search qualifier value from current features			+	+	+	
	Register Pattern		Registration of Sequence patterns			+	+	+	
	Search by Patterns		Search pattern from current sequence			+	+	+	
	Search by Classification Code		Search by classification code			+	+	+	
Jumping	Jump		Show specified region or position of feature map			+	+	+	
Option Settings	Set Feature Attributes & Set Options		Set Feature Attributes or Set Various Parameters for Operation of IMC	Click "Feature Map" Pane	Set Parameter for Feature Map	+	+	+	
				Click "Plasmid Map" Pane	Set Parameter For Plasmid Map	+	+	+	
				Click "Content Map" Pane	Set Parameter For Content Profile	+	+	+	
				Click "Reference Map" Pane	Set Parameters for Reference Map	+	+		
				Click "Genome Map" Pane	Set Parameter For Genome Map		+		
				Click "Directories" Pane	Set Directories	+	+	+	
				Click "Commands" Pane	Set Commands Names and their Locations	+	+	+	
				Click "Classifications" Pane	Set Colors for Classification Categories	+	+	+	
				Click "EC Number" Pane	Set Colors for EC Numbers	+	+	+	
				Click "Category" Pane	Set Colors for Annotation Grades	+	+	+	
				Click "Blast Color" Pane	Set Colors for Scores and Overlap Lengths of Homology Search Results	+	+	+	
				Click "Screening" Pane	Set Qualifiers to be Excluded when Writing as Genbank/EMBL	+	+	+	
				Click "CSV" Pane	Set Minimum Bases When Exporting CSV with Sequences	+	+	+	
				Click "Expand" Pane	Set Rule for Naming Nodes	+	+	+	
Click "Cloning" Pane	Set Dam/Dcm or End Check or Adding of Adenine	+	+	+					


















Appendix A

	Menus	Operation	Main Function	Sub Operation	Sub Functions	Standard Edition	Genomics Edition	Array Edition	
				Click "ORF" Pane	Set choices of Start Codons and Minimum Lengths of ORFs	+	+	+	
				Click "BLAST" Pane	Set Minimum Length for BLAST Search	+	+	+	
				Click "MegaBLAST" Pane	Set Parameters for MegaBLAST Search	+	+	+	
				Click "AutoCopy" Pane	Select Qualifiers to be Copied to Annotated Feature(s)	+	+	+	
				Click "AminoAcid Profile" Pane	Set Pamameters for Amino Acid Profile	+	+		
				Click "PCR Primer Design" Pane	Set Parameter for PCR Design	+	+	+	
				Click "Proxy Server" Pane	Set Port for Proxy Server	+	+	+	
				Click "Lab Note" Pane	Set Host Name and Port Number	+	+	+	
				Click "Set Up" Pane	Set Basic Options for IMC Operation	+	+	+	
Zooming and Scrolling	Zoom In		Zoom in the current feature map			+	+	+	
	Zoom out		Zoom out the current feature map			+	+	+	
	Zoom in for reference feature map		Zoom in the referenced feature map			+	+		
	Zoom out for reference map		Zoom out the referenced feature map			+	+		
	Switch button for Feature map and Reference map co-movement		Toggle switch for co-movement of feature map and referenced map			+	+		
	Scroll Feature Map			Scroll Feature Map		Right Scroll the Feature Map	+	+	+
						Left Scroll the Feature Map	+	+	+
	Scroll Sequence			Scroll Upper Sequence		Right Scroll the Sequence	+	+	+
						Left Scroll the Sequence	+	+	+
Scroll Reference Map			Scroll Each Map Separately		Right Scroll one of DNA on the Reference Map	+	+		
					Left Scroll of DNA on the Reference Map	+	+		
Operation History	View Update History		Show history of editing operation on features		Select Sort Keys and Click "Sort"	+	+	+	
					Check Entries and Click "CSV"	+	+	+	
					Check Entries and Click "Delete"	+	+	+	
	Write Lab Notes		Show and Print Lab Notebook			+	+	+	
Help	View Documents		Show user's manual, tutorial, reference manual			+	+	+	
	Show Version		Show IMC software version			+	+	+	
	Show Java Info		Show Java Information			+	+	+	
Digest by RE			Find RE recognition sites on genomes and digest DNA sequences		Click "Show Recognition Sites"	Show RE Recognition Sites	+	+	+
					Click "Map"	Show RE Recognition Sites on Map	+	+	+
					Click "Add", "Delete", "Edit"	Edit RE Information	+	+	+
					Click "Save"	Save RE Library	+	+	+
					Click "Restore"	Restore RE Library	+	+	+
					Click "More Search"	Digest by Second RE	+	+	+
					Click "Digest"	Digest the Current Sample into Fragments	+	+	+
					Click "Insert As New Feature"	Insert the Recognition Site as New Features	+	+	+
					Click "Gel Electrophoresis"	Perform 1D Gel Eletrophoresis for the Digested Fragments	+	+	+
					Click "CSV"	Save the Fragments as CSV file	+	+	+
					Click "FastA"	Save the Fragments as FastA File	+	+	+
						Click "Import"	Import Primer(s) to the Primer Library	+	+
	Click "Read"	Read a Primer Lbirary	+	+	+				


















Appendix A

	Menus	Operation	Main Function	Sub Operation	Sub Functions	Standard Edition	Genomics Edition	Array Edition	
Molecular Cloning	Register Primer Sets		Input new PCR primer set and register them to the Primer Library	Click "Save as"	Save as a New Primer Library	+	+	+	
				Click "Add"	Add New Primer to Primer Library	+	+	+	
				Click "Delete"	Delete Primer(s) from Primer Library	+	+	+	
				Click "Edit"	Edit a Primer	+	+	+	
				Key in Capital Letters	Add Nucleotide Recognized When Priming	+	+	+	
				Key in Small Letters	Add Nucleotide Un-Recognized When Priming	+	+	+	
				Click "Add RE Sites to Primer"	Attach the Recognition Sites to the Primer Sequence	+	+	+	
	Design Primer		Design Optimal Primer Sets to Amplify the Selected Region of the Feature Map	Click "Design PCR Primer"	Design Optimal Primer Sets	+	+	+	
				Click "Save As Primer"	Save as New Primer Sets	+	+	+	
				Click "Insert As New Feature"	Save as New Features	+	+	+	
				Click "CSV"	Save the Selected Primer Sets as CSV file	+	+	+	
				Click "Amplify"	Start PCR using the Checked Primer Set	+	+	+	
				Click "Save"	Save PCR Product	+	+	+	
	PCR(Amplify DNA)		Find priming sites on DNA sequence and amplify	Click "Find Priming Sites"	Find Priming Site(s)	+	+	+	
				Click "Insert As New Feature"	Save as New Features				
				Click "Amplify"	Amplify the Region(s)	+	+	+	
	Ligation		Ligate two DNA sequences or one fragment	Check "End Check" and Click "Ligation"	Ligate two DNA Fragment with End Type Check	+	+	+	
				Check off "End Check" and Click "Ligation"	Ligate two DNA Fragment without End Type Check	+	+	+	
				Check "End Check" and "Self Ligation" and Click "Ligation"	Self Ligate this DNA Fragment with End Check	+	+	+	
				Check off "End Check" and Check on "Self Ligation" and Click "Ligation"	Self Ligate this DNA Fragment without End Check	+	+	+	
			Find Optimal Restriction Enzymes for Insert Check	Click "Search Enzymes"	Find Optimal Restriction Enzymes for Insert Check of the Ligation Products	+	+	+	
				Click "Gel Electrophoresis"	Perform 1D Gel Electrophoresis for the Digested Fragments of Ligation Products	+	+	+	
			Save Ligation Products	Click "Load"		+	+	+	
	Draw Plasmid Map		Show and Print Circular Plasmid Map	Select Insert Region	Select Insert Sequence to Plasmid	+	+	+	
				Check "Plasmid + Insert" and Click "Show"	Draw Plasmid Map with the Insert Ballooned Out	+	+	+	
				Check "Without Insert" and Click "Show"	Draw Plasmid Map with the Insert invisible	+	+	+	
				Check "All" and Click "Show"	Draw Plasmid Map with Insert in a circle	+	+	+	
				Key In Size of Diameter and Click "Show"	Change the Size of Diameter of Plasmid Map	+	+	+	
				Check Off "Proportional" and Key In Sizes of Insert and Original Plasmid and Click "Show"	Draw Plasmid Map with Insert without Keep Proportion between both lengths	+	+	+	
				Click "Print"	Print Plasmid Map	+	+	+	
				Click "PDF"	Write Plasmid Map as a PDF file	+	+	+	
				+	+	+			
Cut Sequence		Cut off a Region of DNA Sequence							
Attach RE Sites		Attach RE recognition sites on both ends of DNA sequence							
Add T-base		Add one T-base at the ends of DNA fragment							

Appendix A

	Menus	Operation	Main Function	Sub Operation	Sub Functions	Standard Edition	Genomics Edition	Array Edition	
Sequence and Genome Analysis	Blunting		Make blunt ends		Blunting by T4 DNA Polymerase	+	+	+	
					Blunting by Mung Bean Nuclease	+	+	+	
	Dephosphorylation		Apply Phosphatase			+	+	+	
					Apply Kanase	+	+	+	
	Show Codon Usage		Show codon usage table for regions or genes	Right Button on Feature Map		+	+	+	
				Right Button on CDS		+	+	+	
	Show and Change Start Codon Candidates		Show Start Codon Candidates on the ORF and Change Start Codon	Right Button on CDS		+	+	+	
				Right Button on Start Codon	Change Start Cdon from Current to Another	+	+	+	
	Show A.A. Profiles		Show Amino Acid Sequence Profile(s)	Right Button on CDS		Alpha Helix by Chou-Fasman (1978)	+	+	+
						Beta Sheet by Chou-Fasman (1978)	+	+	+
						Beta Turn by Chou-Fasman (1978)	+	+	+
						Relative Mutability by Dayhoff (1978) Ala=100	+	+	+
						Hydrophobicity by Hopp-Woods (1981) For Detection of Antegenic Regions	+	+	+
						Accessive Residues by Janin (1979)	+	+	+
						Buried Residues by Janin (1979)	+	+	+
						Chain Flexibility by Karplus-Schulz (1985)	+	+	+
						Hydrophobicity by Kyte-Doolittle (1982) For the Prediction of Membrane Regions	+	+	+
						Polarity by Zimmerman (1968)	+	+	+
						Bulkiness by Zimmerman-Eliezer-Simha (1968)	+	+	+
	Show Genome Statistics		Show statistics of features and genomic sequence	Check Entries	Show the Detailed Information on each Feature	+	+	+	
				Click a Line	Move the Feature Map to Corresponding Postion	+	+	+	
	Draw Genome Map		Draw and Print Genome Map of the Current Sequence				+		
	Reverse Complement		Generate the reverse complement sequence with complete features on				+	+	+
	Extract ORFs		Extract ORF candicates from unknown genomic sequence		Extract ORFs in Specified Regions	+	+	+	
					Extract ORFs in Intregenic Regions	+	+	+	
	Translate		Change ORF fearute to CDS and translate into amino acid sequeunce				+	+	+
	Import cDNAs		Import cDNA sequences and paste them on to edited sequence				+	+	+
	Import cDNAs into Files		Import cDNA sequences and paste them on to genomics sequences				+	+	+
	Find Repeats		Find Repetitive Sequences in a Genome Sequence					+	
	Compare Genomes		Dot Plot between two Genomes					+	
	Find Non Homologous Regions		Find Non Homologous Regions Between Two Genomes					+	
	Simple Homology Search		Simple homogy search by copy and paste				+	+	+
Homology Search in Batch		Automatic homology search and copy annotation to features				+	+	+	
			Click "Show Homology A.A.	Search Amino Acid Homology against NETBLAST Database	+	+	+		
			Click "Show Homology N.A.	Search Nucleotide Homology against NETBLAST Database	+	+	+		
			Click "Show Homology A.A (Amino	Search Amino Acid Homology against the Selected Database(s)	+	+	+		

Appendix A

	Menus	Operation	Main Function	Sub Operation	Sub Functions	Standard Edition	Genomics Edition	Array Edition
	Homology Search		Search Homology of this Feature	Click "Show Homology A.A."	Search Amino Acid Homology against CDS and RNA Features on the Reference Map	+	+	+
				Click "Show Homology N.A."	Search Nucleotide Homology against All the Features on the Reference Map	+	+	+
				Click "Show Homology A.A."	Search Amino Acid Homology against all the CDS and RNA Features on the Feature Map the Reference Map	+	+	+
				Click "Show Homology N.A."	Search Nucleotide Homology against All the Features on the Feature Map and Reference Map	+	+	+
				Click "Homology Search by Selected CDS and RNA"		+	+	+
Profiling	Show GC Content		Show GC content profile along DNA sequence			+	+	(+)
	Show AT Content		Show AT content profile along DNA sequence			+	+	(+)
	Show A Content		Show A content profile along DNA sequence			+	+	(+)
	Show C Content		Show C content profile along DNA sequence			+	+	(+)
	Show G Content		Show G content profile along DNA sequence			+	+	(+)
	Show T Content		Show T content profile along DNA sequence			+	+	(+)
	Show GC Skew		Show GC skew profile along DNA sequence			+	+	(+)
	Show AT Skew		Show AT skew profile along DNA sequence			+	+	(+)
	Show cumulative GC Skew		Show cumulative GC skew profile along DNA sequence			+	+	(+)
	Show cumulative AT Skew		Show cumulative AT skew profile along DNA sequence			+	+	(+)
Tiling Array Analysis	Show EX Profile(s)		Show Expression Profiles on Feature Map					+
	Set Parameters and Execute Processes		Set Parameters for Array Analysis	Click "Map" Pane	Set Expression Profile Map Parameters			+
				Click "Color" Pane	Set Coloring on Expression Profile Map			+
				Click "Outlier" Pane	Set Parameter for Outliers Cut Off			+
				Click "Reduced" Pane	Set Parameter for Reduced Profile Map			+
				Click "Probe" Pane	Import Probes			+
					Add Probes			+
				Click "Array" Pane	Import Arrays			+
		Add Arrays			+			
	Set Automatic Data Importing				Set Parameters for Automatic Data Importing(in Batch)			+
				Click "SET"	Run Automatic Data Importing(in Batch)			+
	Show Expression per Gene		Show Expression Lists per Gene and per Intergenic Region	Click "SORT"	Sort the List by Intensity and Arrays			+
				Click "SAVE"	Save the Results as CSV file			+
Show Error List		Show Probe List without Expression Value					+	
Plot Scattered Graph		Plot Scattered Graph between any two Arrays					+	

B. Molecular Cloning protocols *in silico* (Part)

Protocol in silico No.1

1. samples
 - Template DNA sequence file B.subtilis
 - Primer set sequence file
2. devices
 - IMC software
3. Methods
 - I. Import a template DNA sequence file into IMC and display the desired region of the sequence.
 - II. Drag on the DNA sequence on forward strand and
 - III. Insert tagged sequence with small characters at the 5' upstream site of the primer sequence.
 - IV. Insert extra sequence with small characters at intermediate site among the primer sequence.
 - V. Perform PCR by PCR button
 - VI. Obtain the amplified product as a new files and view it on the feature map.

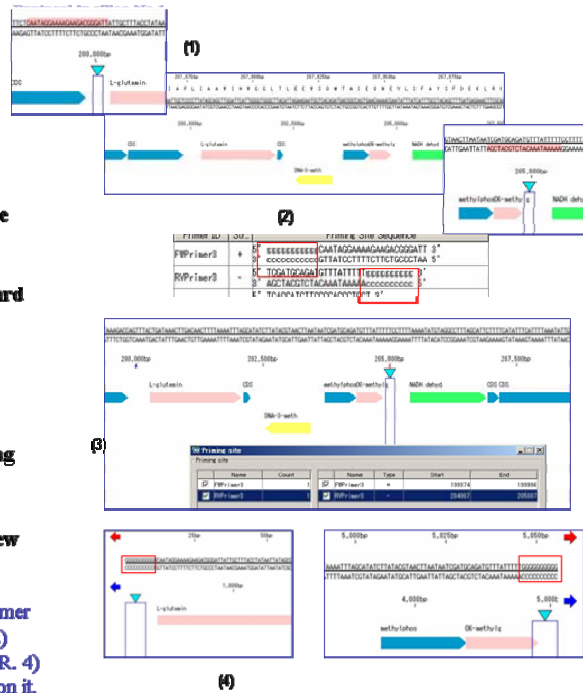


Figure 3.1 Introduction of tags or mutations to primer sequence. 1) Registration of the forward primer. 2) Registration of the reverse primer. 3) Perform PCR. 4) Sequence of the products with insert or mutation on it.

Protocol in silico No.2

1. Sample
 - Plasmid vector sequence pColdIV
 - Template DNA sequence B.subtilis
 - List of Restriction Enzyme Recognition Sites
 - A set of primer sequences
2. Devices
 - IMC Software
3. Methods
 - I. Read a plasmid vector sequence into IMC and select an enzyme which cut once the vector and digest the vector.
 - II. Read the template DNA sequence into IMC and apply a set of primers and perform PCR.
 - III. Perform ligation between the linear vector and the PCR product.
 - IV. View the ligation products and draw the plasmid maps of them.

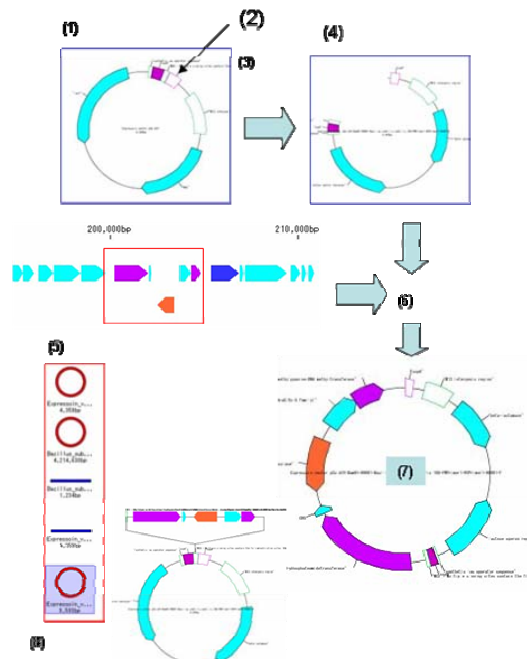


Figure 3.2 Insert of the fragment originated by genomic sequence into plasmid a vector. 1) Plasmid Vector. 2) MCS, 3) Digestion by a restriction enzyme. 4) Linear vector. 5) Amplified region of the template DNA by PCR. 6) Ligation. 7) Plasmid vector with insert. 8) Reaction tube.

Protocols in silico No.3

1. Sample

**Template DNA Sequence of B.subtilis
Sequence of the Restriction Enzyme I-Ceu I**

2. Devices

IMC Software

3. Methods

- I. Read the template DNA sequence into IMC, and show the feature map.
- II. Digest the template DNA sequence by the Restriction Enzyme I-Ceu I.
- III. Obtain the digestion products and check these by 1D gel electrophoresis.

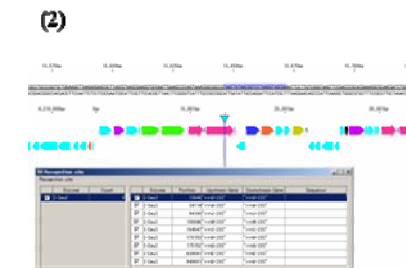


Figure B.3 Digested fragmentation of a genomics Sequence with rare cutter I-Ceu I. 1) Recognition sequence of Restriction Enzyme I-Ceu I. 2) I-Ceu I sites on the genome of B.subtilis. 3) Ladder of digested products by I-Ceu I. 4) Reaction tube of IMC

Protocol in silico No.4

1. Sample

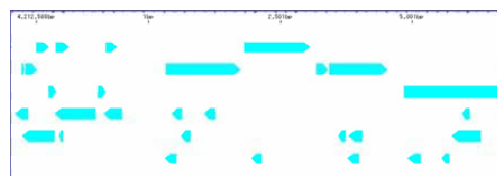
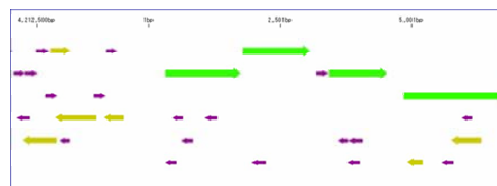
Template DNA sequence derived from prokaryote genome

2. Device

IMC software

3. Methods

- I. Read a template DNA sequence into IMC and show it on the feature map.
- II. Set minimum length for the ORF size and set codons used for the start codon.
- III. Click "Extract ORF" button and extract ORFs from the genomic sequence and insert them on to the feature map.
- IV. Show the candidates of the startcodon and check it.



```

M Y S A I C S F F Y F Y N R E R
TAKTGTGTACGCAATGCTAAAGCCGATTTGGCTTTTFTTFTTGTATACGCAAGAA
ATTACAGATGCTTACGATTCGGGTAAAGGAGGAGGAGGAGGAGATATGCTTCTTCT
R H F L R E G G T C R K M E N I L D L Y
GGCCATTTTCTAAGAAAGCGAGGAGCTGGCGAGCATGAAATATATTAGAGCTGTGG
GGGTAAAGGATTTTTCGGCTGGAGGGGCTTCTAGCTTTTATATAATGTGGACAGC
A Q I E K K L S K P S F E T R W
TGTGTAATGAAAGGATGAGCAAGGCTTTTGAGACTTGCATG
AGGAGTTTAGCTTTTTCAGCTGTTTGGCTCAAACTCGAAGCTAC
A H S L O G D T L T I A P N E
AGGCCACTGACTCAAGGGATACATTAACAATCAGCCCTCCAAAGAA
TCCGGTGAAGTGGTTCGGTATGTAATTTAGTGGGAGGTTACTT
W L E S R Y L H L I A D T I Y E
ACTGGCTGGAGTCCAGATAGTTGCACTGATGGAGACTATATATGAA
AAAGGCTTCTGAGCCAGCTCAGGCTCTATGAGGTAGACTAAGCTGATATATATCTT
    
```

Figure 3.3 Extraction of the longest ORFs from prokaryotic genomic sequence

	A	F	G	H	I	J	K	N	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AV	BP	BX
	Domain	Linear/Circular	Label	genus	species	sub. sp.			GCS Rep	GCS Trans	Average	Whole	Operons Assigned CDS Omitted or Modified	CDSs per Operon	GCS Period	ATS Rep	Average	Whole	ATS Trans	ATS Period	GC content	bases	CDS	rRNA	tRNA	
1									+	++	0.206	0.276	618		4.3		0.085	0.108	+			28.57	3,031,430	2,660	29	96
2	B	C	C.per	Clostridium	perfringens		13																			
3	A	C	M.jan	Methanocaldococcus	jannaschii		DSM 2661		m	++	0.271	0.268	510		3.4	m	0.105	0.108	++			31.43	1,664,970	1,715	6	37
4	B	C	F.nuc	Fusobacterium	nucleatum	nucleatum	ATCC25586			++	0.228	0.262	386		5.4		0.089	0.118				27.15	2,174,500	2,068	15	47
5	B	C	C.tet	Clostridium	tetani		E88		+	++	0.177	0.249	536		4.4		0.090	0.135	+			28.75	2,799,251	2,373	18	54
6	B	C	C.ace	Clostridium	acetobutylicum		ATCC 824		+	++	0.165	0.230	836		4.4		0.086	0.105	+			30.93	3,940,880	3,672	33	73
7	B	L	B.bur	Borrelia	burgdorferi		B31			+	0.117	0.173	227		3.7		0.076	0.052	+			28.59	910,724	850		
8	B	L	B.gar	Borrelia	garinii		PBi			+	0.119	0.172	232		3.6		0.074	0.051	+			28.30	904,246	832	5	31
9	A	C	P.ab	Pyrococcus	abyssi		GE5		m	++	0.174	0.168	564		3.2	m	0.109	0.102	+			44.71	1,765,118	1,784	5	46
10	B	C	B.ant	Bacillus	anthracis		Sterne			+	0.118	0.163	1502		3.5		0.064	0.073	-			35.38	5,228,663	5,287	33	95
11	A	C	P.fur	Pyrococcus	furius		DSM 3638		m	++	0.174	0.163	630		3.3	m	0.103	0.105	++			40.77	1,908,256	2,065	4	46
12	B	C	C.jej	Campylobacter	jejuni	jejuni	NCTC11168			+	0.144	0.163	356		4.6		0.047	0.045	+			30.55	1,641,481	1,654	9	43
13	B	C	B.cer	Bacillus	cereus		ATCC 10987			+	0.109	0.163	1710		3.3		0.063	0.073	-			35.58	5,224,283	5,603	36	97
14	B	C	B.thu	Bacillus	thuringiensis		97-27			+	0.115	0.162	1524		3.4		0.062	0.072	-	+		35.41	5,237,682	5,117	39	105
15	B	C	B.ant	Bacillus	anthracis		Ames		++	+	0.114	0.160	1560		3.4	+	0.062	0.071	+			35.38	5,222,293	5,311	33	95
16	B	C	C.flo	Candidatus Blochmannia	floridanus				-		0.079	0.160	158		3.7		0.027	0.123	+			27.38	705,557	589	3	37
17	A	C	S.sof	Sulfolobus	sofataricus		p2		m	++	0.160	0.159	1076		2.8	m	0.070	0.066	+			35.79	2,992,245	2,995	4	46
18	B	L	C.jej	Campylobacter	jejuni		RM1221			+	0.134	0.154	398		4.6		0.046	0.049	+			30.31	1,777,831	1,838	6	44
19	A	C	N.equ	Nanoarchaeum	equitans		kir4-M		m	++	0.174	0.150	222		2.4	m	0.150	0.139	++			31.56	490,885	536	17	
20	A	C	S.tok	Sulfolobus	tokodaii		7		m	++	0.147	0.147	1016		2.8	m	0.059	0.056	+			32.79	2,694,756	2,826	3	46
21	B	C	M.flo	Mesoplasma	florum		L1		++	++	0.152	0.146	102		6.7				+			27.02	793,224	683		29

	A	F	G	H	I	J	K	N	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AV	BP	BX
	Domain		Linear/Circular	Label	genus	species	sub. sp.		GCS Rep	GCS Trans	Average	Whole	Operons Assigned	CDS Omitted or Modified	CDSs per Operon	GCS Period	ATS Rep	Average	Whole	ATS Trans	ATS Period	GC content	bases	CDS	rRNA	tRNA
1																										
22	A	C	A.ful	Archaeoglobus	fulgidus			DSM 4304	-	+	0.151	0.144	692		3.5	-		0.076	0.070	-		48.58	2,178,400	2,407	3	46
23	B	C	E.rum	Ehrlichia	ruminantium			Gardel	+	+	0.105	0.134	338		2.8			0.026	0.023	+		27.51	1,499,920	950	3	36
24	A	C	M.mar	Methanococcus	maripaludis			S2	m	++	0.139	0.132	510		3.4	m		0.111	0.115	+		33.10	1,661,137	1,722	10	38
25	B	C	M.myc	Mycoplasma	mycoides	mycoides		SC PG1	+	+	0.116	0.131	212		4.8			0.084	0.091	+		23.97	1,211,703	1,016	6	30
26	B	C	L.aci	Lactobacillus	acidophilus			NCFM	+	+	0.090	0.126	450		4.1			0.039	0.040	-		34.71	1,993,564	1,864	13	61
27	B	C	L.joh	Lactobacillus	johnsonii			NCC533	+	+	0.086	0.123	410		4.4			0.037	0.046	+		34.61	1,992,676	1,821	18	79
28	B	C	A.mar	Anaplasma	marginale			St.Maries	+-	+	0.109	0.119	328	5	2.9	+		0.009	0.005	-		49.76	1,197,687	949		37
29	A	C	P.hor	Pyrococcus	horikoshii			OT3	+ / 2	++	0.100	0.118	600		2.9	m		0.057	0.058	+		41.88	1,738,505	1,731	3	40
30	B	C	A.aeo	Aquifex	aeolicus			VF5	-	+	0.102	0.118	450	5	3.4	-		0.138	0.143	+		43.48	1,551,335	1,522	6	44
31	A	C	P.aer	Pyrobaculum	aerophilum			IM2	m	+	0.120	0.115	978		2.7	m		0.051	0.050	-		51.36	2,222,430	2,605	3	36
32	B	C	B.hal	Bacillus	halodurans			C-125	+	+	0.084	0.111	996		4.1			0.046	0.049	-	+	43.69	4,202,352	4,066	25	78
33	B	C	B.aph	Buchnera	aphidicola			Sg-Schizaphis graminum	+	+	0.102	0.111	166		3.3			0.070	0.068	+		25.33	641,454	545	3	32
34	B	C	B.aph	Buchnera	aphidicola			Bp=Baizongia pistaciae	+	+	0.081	0.107	156		3.2			0.064	0.055	+		25.34	615,980	504	3	32
35	A	C	P.tor	Picrophilus	torridus			DSM 9790	m	++	0.112	0.106	562		2.7	m		0.101	0.100	+		35.97	1,545,895	1,535	3	44
36	B	C	U.par	Ureaplasma	parvum			ATCC70970	+	+	0.101	0.106	116		5.3			0.069	0.073	+		25.50	751,719	611	6	30
37	B	C	B.aph	Buchnera	aphidicola			APS=Acyrtosiphon pisum	+	+	0.087	0.102	160		3.5			0.071	0.066	+		26.31	640,681	564	3	32
38	B	C	L.lac	Lactococcus	lactis	lactis		IL1403	+	+	0.062	0.100	436		5.2			0.030	0.041	+		35.33	2,365,589	2,266		
39	A	C	T.vol	Thermoplasma	volcanium			GSS1	m	+	0.098	0.094	536		2.8	m		0.085	0.079	+		39.92	1,584,804	1,526	3	46
40	A	C	M.kan	Methanopyrus	kandleri			AV19	m	+	0.104	0.093	542		3.1	m		0.044	0.049	+		61.16	1,694,969	1,691	3	35
41	B	C	B.sub	Bacillus	subtilis	subtilis		168	++	+	0.076	0.091	1108		3.7	+		0.164	0.073	+		43.52	4,214,630	4,106	30	86
42	B	C	H.pyl	Helicobacter	pylori			26695	+	+	0.074	0.090	362		4.3			0.046	0.056	+		38.87	1,667,867	1,553	7	36
43	B	C	G.kau	Geobacillus	kaustophilus			HTA426	+	+	0.058	0.090	842		4.2			0.047	0.041			52.09	3,544,776	3,498	27	87
44	B	C	B.cla	Bacillus	clausii			KSM-K16	+	+	0.064	0.088	1030		3.9			0.050	0.049	-	+	44.75	4,303,871	4,066	25	78

	A	F	G	H	I	J	K	N	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AV	BP	BX
	Domain	Linear/Circular	Label	genus	species	sub. sp.		strain etc.	GCS Rep	GCS Trans	Average	Whole	Operons Assigned	CDS Omitted or Modified	CDSs per Operon	GCS Period	ATS Rep	Average	Whole	ATS Trans	ATS Period	GC content	bases	CDS	rRNA	tRNA
1									m	+	0.089	0.083	482		3.9	m	m	0.083	0.101	+		49.54	1,751,377	1,869	6	39
45	A	C	M.the	Methanothermobact	thermoautotrophicus			Delta H	m	+	0.089	0.083	482		3.9	m	m	0.083	0.101	+		49.54	1,751,377	1,869	6	39
46	A	C	T.aci	Thermoplasma	acidophilum			DSM 1728	m	+	0.079	0.077	570		2.6	m	m	0.084	0.079	++		45.99	1,564,906	1,478	3	45
47	B	C	B.lic	Bacillus	licheniformis			ATCC14580(G)		+	0.066	0.076	1068		3.9			0.076	0.083	-	+	46.19	4,222,645	4,196	21	72
48	B	C	B.fra	Bacteroides	fragilis			NCTC9343		-	0.065	0.075	1034		4.1			0.051	0.046	-	+	43.19	5,205,140	4,260	19	73
49	A	C	T.kod	Thermococcus	kodakaraensis				m	-	0.083	0.075	722		3.2	m	m	0.101	0.100	+		52.00	2,088,737	2,306	6	46
50	B	C	B.fra	Bacteroides	fragilis			YCH46		-	0.062	0.074	1116		4.1			0.049	0.046	-	+	43.27	5,277,274	4,578	18	74
51	B	C	H.duc	Haemophilus	ducreyi			35000HP		+	0.057	0.074	464		3.7			0.026	0.029	+		38.22	1,898,955	1,717	19	45
52	B	C	B.the	Bacteroides	thetaiotaomicron			VPI-5482		+	0.062	0.072	1176		4.1			0.052	0.055	-	+	42.84	6,260,361	4,778	15	71
53	B	C	P.mar	Prochlorococcus	marinus	marinus		CGMP1375		++	0.074	0.072	726		2.6					+		36.44	1,751,080	1,882	3	40
54	B	C	B.hen	Bartonella	henselae			Houston-1		+	0.039	0.070	468		3.4			0.002	0.008	-		38.23	1,931,047	1,612	7	44
55	B	C	B.qui	Bartonella	quintana			Toulouse		-	0.044	0.070	396		3.3			0.007	0.019	-		38.80	1,581,384	1,308	6	42
56	B	C	B.bac	Bdellovibrio	bacteriovorus			HD100		-	0.028	0.069	1182		3.0			0.026	0.020	-		50.65	3,782,950	3,583		36
57	A	C	M.maz	Methanosarcina	mazei			Go1	m	+	0.068	0.066	1086		3.1	+/-3		0.068	0.071	+		41.48	4,096,345	3,371	10	57
58	B	L	L.pne	Legionella	pneumophila			Lens		+	0.051	0.065	966		3.1			0.026	0.027	-		38.41	3,345,687	2,947	9	43
59	B	c1	C L.int	Leptospira	interrogans			Fiocruz L1-130		+	0.045	0.057	1044		3.3			0.026	0.027	+		35.05	4,277,185	3,394	5	37
60	A	C	M.ace	Methanosarcina	acetivorans			C2A	m	+	0.058	0.056	1516		3.0	m	m	0.063	0.069	+		42.68	5,751,492	4,540	10	
61	B	C	Y.pse	Yersinia	pseudotuberculosis			IP32953		+	0.052	0.056	1092		3.6			0.005	0.007	-		47.61	4,744,671	3,974	21	85
62	B	L	E.col	Escherichia	coli			O157-H7		+	0.054	0.055	1441		3.7			0.008	0.010	-		50.54	5,498,450	5,361	22	105
63	B	C	A.sp.	Acinetobacter	sp.			ADP1	++	+	0.066	0.053	1004	0	3.3	-		0.001	0.000	-		40.43	3,598,621	3,325	21	76
64	A	C	A.per	Aeropyrum	pernix			K1	-	+	0.032	0.052	958		2.8	-		0.012	0.023	-	+	56.31	1,669,695	2,694	5	47

	A	F	G	H	I	J	K	N	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AV	BP	BX
	Domain		Linear/Circular	Label	genus	species	sub. sp.		GCS Rep	GCS Trans	Average	Whole	Operons Assigned	CDS Omitted or Modified	CDSs per Operon	GCS Period	ATS Rep	Average	Whole	ATS Trans	ATS Period	GC content	bases	CDS	rRNA	tRNA
1																										
65	B		C	D.eth	Dehalococcoides	ethenogenes		195	+	0.044	0.045	432	3.7	0.033	0.037	1,469,720	48.85						1,580	3	46	
66	B	c1	C	V.cho	Vibrio	cholerae	O1	N16961	+	0.037	0.044	848	3.2	0.008	0.007	2,961,149	47.69						2,736	24	94	
67	B	c2	C	V.cho	Vibrio	cholerae	O1	N16961	+	0.036	0.044	392	2.8	0.003	0.005	1,072,315	46.91						1,092	4	4	
68	B		C	Z.mob	Zymomonas	mobilis		ZM4	+	0.033	0.044	668	3.0	0.022	0.031	2,056,416	46.33						1,998	9	51	
69	B		C	C.mur	Chlamydia	muridarum			+	0.035	0.044	270	3.3	0.035	0.027	1,072,950	40.34						904	6	37	
70	B		C	C.tra	Chlamydia	trachomatis		D/UW-3/CX	-	0.029	0.041	264	3.4	0.031	0.026	1,042,519	41.31						895	6	37	
71	B		C	C.bur	Coxiella	burnetii		RSA493	-	0.034	0.041	630	3.2	0.016	0.013	1,995,275	42.66						2,010	3	42	
72	B		C	E.col	Escherichia	coli		K-12 MG1655	+	0.045	0.040	1310	3.2			4,639,675	50.79						4,254	22	86	
73	B		C	D.psy	Desulfotalea	psychrophila		LSv54	-	0.035	0.036	764	4.1	0.004	0.004	3,523,383	46.81						3,118	22	65	
74	B		C	C.cav	Chlamydophila	caviae		GPIC	+	0.032	0.031	290	3.4	0.016	0.008	1,173,390	30.45						998	3	3	
75	B		C	C.abo	Chlamydophila	abortus		S26-3	-	0.025	0.024	286	3.4	0.017	0.008	1,144,377	39.87						961	3	38	
76	B		C	S.pom	Silicibacter	pomeroyi		DSS-3	+	0.020	0.020	1214	3.1			4,109,442	64.22						3,810	9	51	
77	B		C	C.pne	Chlamydophila	pneumoniae		AR39	+	0.017	0.020	310	3.6	0.016	0.009	1,229,853	40.57						1,110	3	38	
78	A	c2	C	H.mar	Haloarcula	marismortui		ATCC 43049	-	0.014	0.019	105	2.7	0.048	0.060	288,050	57.23						281	3	1	
79	B		C	M.tub	Mycobacterium	tuberculosis		CDC1551	-	0.010	0.016	1390	3.0			4,403,837	65.61						4,189			
80	B		C	B.mal	Burkholderia	mallei		ATCC 23344 chr.2	-	0.010	0.011	542	3.3	0.008	0.004	2,325,379	68.99						1,768	6	3	
81	B		C	B.pse	Burkholderia	pseudomallei		K96243 chr.2	+	0.008	0.011	688	3.5	0.013	0.005	3,173,005	68.49						2,395	3	8	
82	B		C	B.mal	Burkholderia	mallei		ATCC 23344 chr.1	-	0.009	0.009	870	3.4	0.009	0.010	3,510,148	68.15						2,996	4	47	
83	B		C	B.pse	Burkholderia	pseudomallei		K96243 chr.1	-	0.007	0.009	962	3.6	0.004	0.007	4,074,542	67.72						3,460	9	53	
84	B		C	C.dip	Corynebacterium	diphtheriae		NCTC 13129	-	0.015	0.007	704	3.3	0.001	0.010	2,488,635	53.48						2,320	15	54	
85	A	c1	C	H.mar	Haloarcula	marismortui		ATCC 43049	-	0.007	0.006	1198	2.6	0.060	0.067	3,131,724	62.36						3,131	6	48	

	A	F	G	H	I	J	K	N	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AV	BP	BX
	Domain	Linear/Circular	Label	genus	species	sub. sp.		strain etc.	GCS Rep	GCS Trans	Average	Whole	Operons Assigned	CDS Omitted or Modified	CDSs per Operon	GCS Period	ATS Rep	Average	Whole	ATS Trans	ATS Period	GC content	bases	CDS	rRNA	tRNA
1																										
86	B	C	C.vio	Chromobacter	violaceum			ATCC12472	-	-	0.005	0.006	1296		3.4			0.031	0.040	-		64.83	4,751,080	4,407	24	98
87	B	C	G.oxy	Gluconobacter	oxydans			621H	-	-	0.006	0.001	766		3.2			0.024	0.026	-		61.07	2,702,173	2,432	12	50
88	B	L	A.tum	Agrobacterium	tumefaciens			C58 Linear	+	-	0.001	0.005	577		3.3	+		0.013	0.019	-		59.28	2,075,560	1,876	6	13
89	A	C	H.sp.	Halobacterium	sp.			NRC-1	+/-	-	0.010	0.006	854		2.4	-		0.061	0.066	+		67.91	2,014,239	2,058	4	47
90	B	C	C.tep	Chlorobium	tepidum			TLS	-	-	0.005	0.006	722		3.1			0.041	0.035	-		56.53	2,154,946	2,252	6	50
91	B	C	C.glu	Corynebacterium	glutamicum			ATCC 13032	-	-	0.013	0.007	1026		3.0					-		53.81	3,309,401	3,099	19	60
92	B	C	A.tum	Agrobacterium	tumefaciens			C58 Circular	+	-	0.003	0.007	934		3.0	+		0.011	0.014	-		59.38	2,841,490	2,785	6	40
93	B	C	C.glu	Corynebacterium	glutamicum			ATCC 13032	-	-	0.014	0.008	1002		3.1			0.003	0.007	-		53.84	3,282,708	3,058	18	60
94	B	C	M.lot	Mesohrizobium	loti				-	-	0.007	0.010	2268		3.0			0.005	0.005	-		62.75	7,036,071	6,752	6	50
95	B	c2	C.D.rad	Deinococcus	radiodurans			R1 chr.2	-	-	0.008	0.011	370		1.0			0.025	0.033	-		66.69	412,348	357		1
96	B	C	B.bro	Bordetella	bronchiseptica			RB50	-	-	0.016	0.012	1382		3.6			0.006	0.012	+		68.08	5,339,179	5,006	9	55
97	B	C	B.par	Bordetella	parapertussis			12822	-	-	0.016	0.012	1292		3.4			0.004	0.007	+		68.10	4,773,551	4,402	9	54
98	B	c1	C.D.rad	Deinococcus	radiodurans			R1 chr.1	-	-	0.013	0.013	904		2.9			0.028	0.032	+		67.01	2,648,638	2,579	9	48
99	B	C	C.cre	Caulobacter	crenscetus			CB15	-	-	0.010	0.013	1260		3.0			0.003	0.002	-		67.21	4,016,947	3,737	6	51
100	B	C	B.jap	Bradyrhizobium	japonicum			USDA110	-	-	0.010	0.014	2714		3.1			0.009	0.007	+		64.06	9,105,828	8,317	3	50
101	B	C	B.per	Bordetella	pertussis			Tahama I	-	-	0.039	0.019	1130		3.4			0.011	0.011	+		67.72	4,086,189	3,806	9	51
102	B	L	S.ave	Streptomyces	avermitilis			MA-4680	-	-	0.015	0.020	2696		2.8			0.011	0.011	+		70.72	9,025,608	7,575	18	68
103	B	C	B.lon	Bifidobacterium	longum			NCC2705	-	-	0.017	0.022	546		3.2			0.022	0.034	+		60.12	2,256,646	1,727	4	57
104	B	C	D.vul	Desulfovibrio	vulgaris			Hildenborough	-	-	0.014	0.030	978		3.5			0.001	0.016	+		63.14	3,570,858	3,379	15	68
105	B	C	C.eff	Corynebacterium	efficiens			YS-314	-	-	0.029	0.030	984		3.0			0.001	0.001	-		63.14	3,147,090	2,942	15	56
106	B	C	P.aer	Pseudomonas	aeruginosa			PA01	RV		0.032	0.032	1644		3.4			0.019	0.025	-		66.56	6,264,403	5,566	12	63

