

琉球大学学術リポジトリ

遺伝子塩基配列編集プログラム

メタデータ	言語: 出版者: 琉球大学理学部 公開日: 2008-10-31 キーワード (Ja): キーワード (En): 作成者: 澤岷, 敬一郎, 細谷, 将彦, Takushi, Keiichirou, Hosoya, Masahiko メールアドレス: 所属:
URL	http://hdl.handle.net/20.500.12000/7728

遺伝子塩基配列編集プログラム

澤舩 敬一郎*・細谷 将彦*

*琉球大学理学部物理学科

TAKUSHI, Keiichirou* & Masahiko HOSOYA*: An Editor to Adjust Gaps int DNA or RNA Sequences

Abstract

Three computer programs are presented to edit DNA or RNA sequences by adjusting gaps int them. All programs are written in Quick BASIC on Macintosh. The present editor can treat 50 species simultaneously each of which contains not exceeding 25000 DNA or RNA bases.

はじめに

現在、進化や遺伝を研究するためにはDNAやRNAの配列を直接比較する事が不可欠である。しかし、これらの配列は進化の過程で欠落や挿入、重複をおこしており、異なった種の間で真に対応する部分の配列を比較するためには、配列中に適当なギャップを設定し、その間隔を調整することが必要である。

村田氏は、生物種間のDNAデータを比較して一致の度合いを手動的に編集できるプログラム【DNAGE】[S.MURATA (HOSEI Univ. Department of Electrical Engineering) "DNAGE.BAS", 1987] を作成した。このプログラムは日本電気製MS-DOS上で動作するMS-DOS版N88BASICで記述されている。これを用いれば、最大10種の生物種間についてのDNAデータ(1種につき最大1万個)の比較・編集ができる。しかし、生物種の数・DNAデータ量は膨大であり、この【DNAGE】で扱えるデータ量ではあまり実用にならない。そこで我々は、【DNAGE】で扱える生物種・DNAデータの量を改良すると同時に、操作性の良いマッキントッシュに移植をした。また同時にユーティリティプログラム2本も開発した。使用した言語はマッキントッシュ上で動作するMAC版QuickBASICである。このプログラムを用いれば、最大50種の生物種間についてのDNAデータ(1種につき最大2万5千個)の比較・編集が可能である。つまり、【DNAGE】で扱えるデータの量が10万(10種×1万個)であったのに対して、このプログラムでは125万(50種×2万5千個)になった。また、ほとんどの操作がマウスで行えるようになり、操作性が向上した。

このように大量のデータが扱えるようになったのは、データファイルの仕様をシーケンシャルファイルからランダムファイルに変更した為である。そのために村田氏の【DNAGE】で扱う村田式ファイルからこのプログラムで扱う澤舩式ファイルへと変換するファイルコンバートプログラムも同時に開発した。また、このプログラムでは編集の過程をファイルに保存するようにしており、このファイルの内容を参照するための編集記録参照プログラムも同時に開発した。

本論文では、以上3つのプログラムを公表する。

受理 : 1993年8月15日

*Department of Physics, College of Science, University of the Ryukyus, Nishihara, Okinawa 903-01 Japan

プログラムの概要

2.1 DNA塩基配列編集プログラム

このプログラムは、DNA塩基配列データの編集をするためのエディタである。ただし、編集ができるDNA塩基配列データは次の条件を満たすものに限る。

1) 専用のファイルコンバータで作成された澤岬式ファイルであること、2) 種50、塩基データ長2万5千以内。

このプログラムは次の編集処理機能を持つ。

1) アンドゥ処理、2) カット処理、3) コピー処理、4) ペースト処理、5) 挿入処理、6) 編集処理記録

2.2 ファイルコンバータ

このプログラムは次の2種類のファイルコンバートを行うプログラムである。

①村田式ファイルから澤岬式ファイルへのコンバート、②澤岬式ファイルから村田式ファイルへのコンバート

なお、村田式ファイルとは、村田氏の制作した『DNAGE』で作成されたファイル、あるいは「4.1 ファイル仕様」の村田式ファイルの仕様と同等のファイルのことをさす。

また、澤岬式ファイルとは本論文で紹介する『DNA塩基配列プログラム』で使用するファイルであり、このプログラムで作成する。

2.3 編集記録参照プログラム

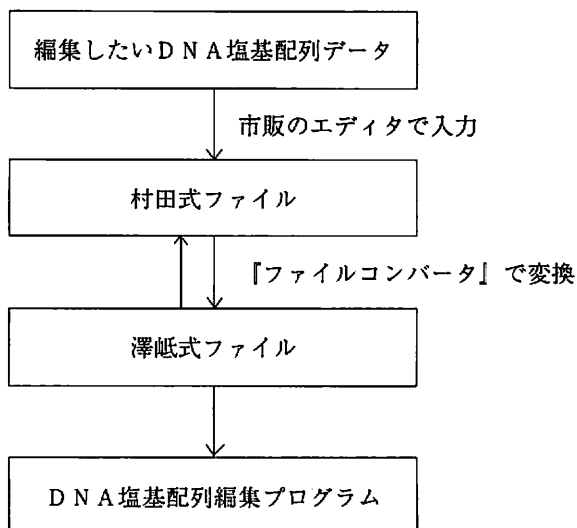
このプログラムは、DNA塩基配列編集プログラムで記録された編集処理を参照するユーティリティプログラムである。

編集方法

3.1 まえがき

本論文で紹介する3本のプログラムの使用方法を説明する前に、これらのプログラムの関連を述べる。まずこの3本のプログラムの中でメインとなるのは『DNA塩基配列編集プログラム』である。このプログラムを使ってDNA塩基配列を編集するのだが、その元になるデータの入力はこのプログラムでは処理できない。ひとまず市販のエディタで入力するしかなく、またファイルの書式は「5.1.1 村田式ファイル」で説明する村田式ファイルと同一のものでなくてはならない。さらにこの村田式ファイルを、もうひとつのプログラム『ファイルコンバータ』を使って澤岬式ファイルに変換する必要がある。この澤岬式ファイルを読み込んでDNA塩基配列の編集を行う。

上記の説明を図で表すと次のようになる。また、『DNA塩基配列編集プログラム』で編集したデータを市販のエディタで利用したい、あるいは村田氏の『DNAGE』で利用したい時には、逆に澤岬式ファイルを『ファイルコンバータ』で村田式ファイルに変換して利用することができる。



【DNA塩基配列編集プログラム】では編集の過程を記録する処理を行っている。そして、この編集記録はもう一つのプログラム【編集記録参照プログラム】で参照できる。

【DNA塩基配列編集プログラム】にその機能を加える事も可能だが、プログラムサイズが大きくなりすぎて編集可能なデータの量が小さくなる事をさけた。

また使用方法の説明を行う前に、マッキントッシュの基本操作を習得しておく必要があるが、その説明は本論文では行わない。マッキントッシュの各機種に付属の【Macintosh™ System software User's Guide】に詳しい説明があるので、それを参照されたい。

以下では、その基本操作を理解・習得しているものとみなして使用方法の説明を行う。

3.2 編集

ある塩基配列データを入手し、編集するまでの過程を以下に示す。

1) 塩基配列データの入手

まず、比較したい塩基配列 (DNA、RNA) データを文献、雑誌あるいは直接のデータから入手する。

2) 村田式ファイル作成

用意した塩基配列データを市販のエディタを用いて入力する。その入力は次に示すような村田式ファイルの形式で入力する。

種の数

フォーマットチェック文字 (===) 種名	} 1行に おさめる	} ×種の数
データ数,空白 (桁数任意),1 (チェック用)		
空白 (桁数任意),データ数 (チェック用)		
塩基配列データ (1行あたり59桁以内)		

例を以下に示す。

```

3
=== HUMAN M T
1630      1 1630
GCTAAACCTAGCCCCAAACCCACTCCACC*****TACTACCAGACACCTTAGCCAAAC
CATTTA**CCAAA**TAAA*GTATAGGCGATAGAAATT***GAAAC**CTGGCGCAAT
.....  途中省略
=== BOVINE
1630      1 1630
ACTAGACCTAGCC*AAAGATACCCCTCTCGACTAAACAACCAAGATAGAATAAAACAAAA
CATTTAATCCCAA*TTTAAA*GTATAGGAGATAGAAAT*AAAGTA*****CGGCGCTAT
.....  省略

```

以下の説明での便宜上、このファイル名を“sample.mrt”とする。

3) 澤舩式ファイルに変換

a) 準備

使用するマッキントッシュのハードディスクに【Microsoft™ Quick BASIC】がインストールされていることを確認する。

プログラムの入ったフロッピーディスクを用意する。このディスクはディスク名が“DNA Gap Editor”、そして中には次のファイルが入っているものとする。

```

DNA Gap Editor Ver 2.1    .... プログラムファイル
DGE-Converter Ver 2.1    .... プログラムファイル
Tracer of editing Ver 2.1 .... プログラムファイル

```

さらに、「2) 村田式ファイルの作成」で作成した村田式ファイル“sample.mrt”をいれておくフォルダを作成する。ここでは便宜上、そのフォルダ名を“sample”という名前にしておく。そうしてそのフォルダの中に“sample.mrt”をいれる。そうするとフロッピーディスクには次のファイルとフォルダが入っていることになる。

```

DNA Gap Editor Ver 2.1    .... プログラムファイル
DGE-Converter Ver 2.1    .... プログラムファイル
Tracer of editing Ver 2.1 .... プログラムファイル
sample                    .... データフォルダ
└── sample.mrt            .... 村田式ファイル

```

注意：“DNA Gap Editor Ver 2.1”で作成されるファイル数は最大で17個にもなる。

それで仮にミトコンドリアとヘモグロビンといったように、2種類の塩基配列データをいれたとすると、ファイル数が最大で34個になりオープニングファイルのオープンがスムーズに行えない。そこで、1種類の塩基配列データにつきデータフォルダを1個作成することを勧める。

b) 起動と終了

起動

①電源を入れ、システムが起動したら【Microsoft™ Quick BASIC】のあるフォルダを開いてこれをダブルクリックする。

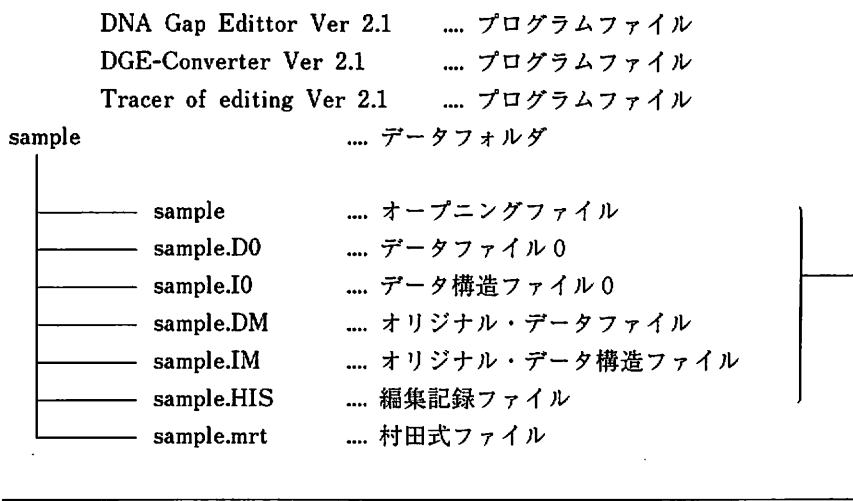
②【Microsoft™ Quick BASIC】が起動したら、プログラム・データの入ったフロッピーディスクをディスクドライブに挿入する。

プログラムの読み込み

- ③メニューバーの“File”のプルダウンメニューを開き“Open”を選択する。
- ④ファイルをオープンするドライブをフロッピーディスクにする。ディスク名は “DNA Gap Editor”である。表示されているファイル名の中から“DGE-Converter Ver 2.1”を選択する。
実行
- ⑤プログラムの読み込みが済んだら、メニューバーの“Run”のプルダウンメニューを開き “Run Program”を選択する。
プログラム終了
- ⑥メニューバーの“File”のプルダウンメニューを開き“Quit”を選択する。
【Microsoft™ QuickBASIC】の終了
- ⑦メニューバーの“File”のプルダウンメニューを開き“Quit”を選択する。

c) 変換

- ①メニューバーの“Convert”のプルダウンメニューを開き“Murata => Takushi”を選択する。
- ②ファイルをオープンするドライブをフロッピーディスクにする。ディスク名は “DNA Gap Editor”である。
“Select a data folder, then open a file with extension “mrt”. ”とメッセージが出るので、必要なデータフォルダ（例えば“sample”）を開き、その中から村田式ファイル（例えば “sample.mrt”）を開く。
- ③確認をする。ファイルの内容（生物種名とその塩基数）が表示されるので、ファイルの選択が正しければ“Yes”のボタンを、間違っていれば“No”のボタンをクリックする。“No”を選択した場合には、①から②までの処理を再度行う。
- ④変換が終了したら、“COMPLETE ! Click this window.”とメッセージが表示されるのでウインドウをクリックする。
- ⑤メニューバー“File”のプルダウンメニューを開き“Quit”を選択する。
- ⑥以上の操作でフロッピーディスクの内容は次のように変わる。



└── 新しく増えたファイル（澤岬式ファイル）

4) 「DNA塩基配列編集プログラム」で編集

すでに“DNA Gap Editor Ver 2.1”が起動しているものとする。起動および終了の方法については「3) 澤舩式ファイルに変換 b) 起動と終了」を参照せよ。

以下では一般的な各編集操作方法について述べる。よって説明の流れに沿って編集するのではなく、必要な編集操作の部分を読んで編集を行えばよい。

a) 塩基データファイルを読み込む

①メニューバー“File”のプルダウンメニューを開き“Open”を選択する。

②ファイルをオープンするドライブをフロッピーディスクにする。ディスク名は“DNA Gap Editor”である。

“Select a data folder, then open a file with no extension”とメッセージが出るので、必要なデータフォルダ（例えば“sample”）を開き、その中からオープニングファイル（例えば“sample”）を開く。

③ファイルのバージョンを選択する。（バージョン0が常に一番新しいファイルである。）通常は“0”のボタンをクリックする。

④確認をする。ファイルの内容（生物種名とその塩基数）が表示されるので、ファイルの選択が正しければ“Yes”のボタンを、間違っていれば“No”のボタンをクリックする。

“No”を選択した場合には、①から③までの処理を再度行う。

b) 操作する塩基データをクリップボードにコピーする。

今、編集画面には次の塩基データが表示されているとする。このデータの座標10から13までの“AGCC”をコピーするものとして以下の説明をする。

: 1

1 HUMA GCTAAACCT **A**GC**C**CCAAACCCACTCCACC*****TTACT

①メニューバー“Edit”のプルダウンメニューを開き“Copy”を選択する。

②“Click the left end”とメッセージが出るので、コピーしたい塩基データのうち左端の塩基データをクリックする。今の場合、座標10の“A”をクリックする。その文字が反転する。なお途中で処理を中断したい場合には、画面中にある“ESC”の文字をクリックすれば処理を中断して抜ける事ができる。

③“Click the right end”とメッセージが出るので、コピーしたい塩基データのうち右端の塩基データをクリックする。今の場合、座標13の“C”をクリックする。その文字が反転する。

④確認をする。指定した範囲が正しければ“Yes”のボタンを、間違っていれば“No”のボタンをクリックする。“No”を選択すると③の処理に戻る。また初めから範囲の指定を行いたい場合には、いったん“No”を選択した上で画面の“ESC”の文字をクリックしてもう一度①の処理から行う。

c) カット（指定した塩基データの削除、削除されたデータはクリップボードに格納される。）今、編集画面には次の塩基データが表示されているとする。このデータの座標10から13までの“AGCC”をカットするものとして以下の説明をする。

: 1

1 HUMA GCTAAACCT **A**GC**C**CCAAACCCACTCCACC*****TTACT

①メニューバー“Edit”のプルダウンメニューから“Cut”を選択する。

②これから先の操作手続きはコピーの操作と同様である。まずカットしたい塩基データの左端をクリックし、つぎに右端をクリックする。その後、指定した範囲の確認をとる。今の場合、まず座標10の“A”をクリックし、次に座標13の“C”をクリックする。そして確認では“Yes”のボタンをクリックする。

d) ペースト（クリップボードに格納されている塩基データを挿入する。）

今、編集画面には次の塩基データが表示されているとする。このデータの座標10の前にクリップボードの“AGCC”をペーストするものとして以下の説明をする。

: 1

1 HUMA GCTAAACCT **A** GCCCCAAACCCACTCCACC*****TTACT

①メニューバー“Edit”のプルダウンメニューから“Paste”を選択する。

②“Click insertion point”とメッセージが出るので、ペーストしたい位置をクリックする。今の場合、座標10の“A”をクリックする。その文字が反転する。

③確認をする。この位置で良ければ“Yes”のボタンを、間違っていれば“No”のボタンをクリックする。

e) 挿入（任意の塩基データを挿入する。）

今、編集画面には次の塩基データが表示されているとする。このデータの座標10の前に“GGCA”を挿入するものとして以下の説明をする。

: 1

1 HUMA GCTAAACCT **A** GCCCCAAACCCACTCCACC*****TTACT

①メニューバー“Edit”のプルダウンメニューから“Insert”を選択する。

②“Click insertion point”とメッセージが出るので、挿入したい位置をクリックする。今の場合、座標10の“A”をクリックする。その文字が反転する。

③確認をする。この位置で良ければ“Yes”のボタンを、間違っていれば“No”のボタンをクリックする。

④挿入するデータを作成するウィンドウがオープンする。

各ボタンの機能

T : 塩基データ“T”を入力する。

C : 塩基データ“C”を入力する。

A : 塩基データ“A”を入力する。

G : 塩基データ“G”を入力する。

* : 挿入データ“*”を入力する。

BS : 最後尾にある塩基データを1つ削除する。

CLEAR : 入力中の塩基データを全て削除する。

END : 塩基データの入力を終了し、そのデータの挿入処理を行う。

CANCEL : 挿入処理をとりやめる。

⑤ここで、“GGCA”を入力する。

“G”ボタンをクリックする。“G”ボタンをクリックする。“C”ボタンをクリックする。“A”ボタンをクリックする。“END”ボタンをクリックする。

⑥確認をする。この塩基データで良ければ“Yes”のボタンを、間違っていれば“No”のボタンを

クリックする。“No”をクリックした場合にはもう一度塩基データの入力をする。

⑦挿入の処理が行われる。

f) アンドゥ（処理を施したひとつ前の状態に戻す。）

①メニューバー“Edit”のプルダウンメニューから“Undo”を選択する。

②処理を施した前の状態に戻る。

g) 編集領域の変更

(1)水平方向の移動

例：塩基データの座標600を画面左端にもってくる。

①メニューバーの“Area”のプルダウンメニューを開き“Horizontally”を選択する。

②“Input the left end of an area ”とメッセージが出るので、“600”と入力する。

(2)縦方向の移動

これは11種以上の生物種の塩基配列編集をするときにのみ使用する。例えば、今30種類の生物種を扱っていて、画面最上位置にあるのが生物種1だとする。そうすると画面上には1から10までの生物種のデータが表示されていることになる。生物種16から25までを表示するように変更するには、次のように操作すればよい。

①メニューバーの“Area”のプルダウンメニューを開き“Uppermost position”を選択する。

②“Input the number of species for the uppermost position ”とメッセージが出るので、“16”と入力する。

h) 塩基データのスクロール

(1)右へのスクロール

・データ5個分の移動：画面上の“>”ボタンをクリックする。

・データ30個分の移動：画面上の“>>”ボタンをクリックする。

(2)左へのスクロール

・データ5個分の移動：画面上の“<”ボタンをクリックする。

・データ30個分の移動：画面上の“<<”ボタンをクリックする。

5) 澤岷式ファイルから村田式ファイルへの変換

澤岷式ファイルでの塩基配列データはランダムアクセスファイルであるので、テキストファイルとしてデータが必要な場合には村田式ファイルに変換して利用すれば良い。「4」【DNA塩基配列編集プログラム】で編集」で編集した澤岷式ファイルを村田式ファイル（例えば“test.mrt”）へ変換する手順を以下に示す。

a) 変換

①メニューバーの“Convert”のプルダウンメニューを開き“Takushi => Murata”を選択する。

②ファイルをオープンするドライブをフロッピーディスクにする。ディスク名は“DNA Gap Editor”である。

“Select a data folder, then open a file with no extension.”とメッセージが出るので、必要なデータフォルダ（例えば“sample”）を開き、その中からオープニングファイル（例えば“sample”）を開く。

③ファイルのバージョンを選択する。（バージョン0が常に一番新しいファイルである。）通常は“0”のボタンをクリックする。

④確認をする。ファイルの内容（生物種名とその塩基数）が表示されるので、ファイルの選択が正しければ“Yes”のボタンを、間違っていれば“No”のボタンをクリックする。“No”を選

択した場合には、①から③までの処理を再度行う。

- ③ここで“name of destined file ”とメッセージが出るので、ファイル名（例えば“test.mrt”）を入力する。
- ④変換が終了したら、“COMPLETE ! Click this window.” とメッセージが表示されるのでウインドウをクリックする。
- ⑤メニューバーの“File”のプルダウンメニューを開き“Quit”を選択する。

3.3 編集記録参照

「3.2 編集」での1)～4)までの処理を行ったものとして、以下にその編集記録の参照方法を示す。

- ①「3.2 編集」で使用したフロッピーディスクをディスクドライブに挿入する。
- ②【編集記録参照プログラム】（“Tracer of editing Ver 2.1”）を起動する。（起動と終了の方法は「3.2・3）澤舩式ファイルに変換 b）起動と終了」を参照せよ。）
- ③メニューバー“File”のプルダウンメニューを開き“Open”を選択する。
- ④ファイルをオープンするドライブをフロッピーディスクにする。ディスク名は“DNA Gap Editor”である。
 “Select a data folder, then open a file with extension” “HIS”.とメッセージが出るので、必要なデータフォルダ（例えば“sample”）を開き、その中から編集記録ファイル（例えば“sample.HIS”）を開く。
- ⑤参照ウインドウがオープンする。
- ⑥編集記録の参照は次のボタンをクリックして行う。ただし、次の操作は11個以上の編集記録がなければ必要ない。

UP ... 編集記録を上方向に5つスクロールする。
 DOWN ... 編集記録を下方向に5つスクロールする。
 TOP ... 編集記録1番を画面最上位にもってくる。
 BOTTOM. .. 編集記録の最後を画面最下位にもってくる。

主要なアルゴリズム

4.1 ファイル仕様

ここで扱う塩基データファイルは大きく分けて次の2通りある。

- ①村田式ファイル：村田氏が作成したプログラム、【DNAGE】（PC-9801 N88-BASIC）で作成されたファイルである。このファイルは1種類のシーケンシャルファイルからなり、一般のエディタでも読み込み・編集が可能である。
- ②澤舩式ファイル：今回作成したプログラム【DNA Gap Editor Ver 2.1】（マッキントッシュ Quick BASIC）で扱うファイルである。このファイルは次の4種類からなる。またデータファイルの場合、一般のエディタでの読み込みは可能だが、ランダムファイルであるために編集には適さない。①オープニングファイル、②データ構造ファイル、③データファイル、④編集記録ファイル

1) 村田式ファイル

- ①ファイルの種類

ひとつのファイル名につきファイルはひとつ（単独）

【例】ファイル名 mit の場合 → mit.mrt

②データ形式

種の数

フォーマットチェック文字 (===) 種名 データ数,空白 (桁数任意),1 (チェック用), 空白 (桁数任意),データ数 (チェック用) 塩基配列データ (1行あたり59桁以内)) 1行に おさめる	} ×種の数
--	---------------	--------

【例】

2

=== HUMAN MT

1630 1 1630

GCTAAACCTAGCCCCAAACCCACTCCACC*****TTACTACCAGACACCTTAGCCAAAC
 CATTTA**CCAAA**TAAA*GTATAGGCGATAGAAATT***GAAAC**CTGGCGCAAT

…… 途中省略

=== BOVINE

1630 1 1630

ACTAGACCTAGCC*AAAGATACCCTCTCGACTAAACAACCAAGATAGAATAAAACAAAA
 CATTTAATCCCAA*TTTAAA*GTATAGGAGATAGAAAT*AAGTA*****CGGCGCTAT

…… 省略

2) 澤舩式ファイル

①ファイルの種類

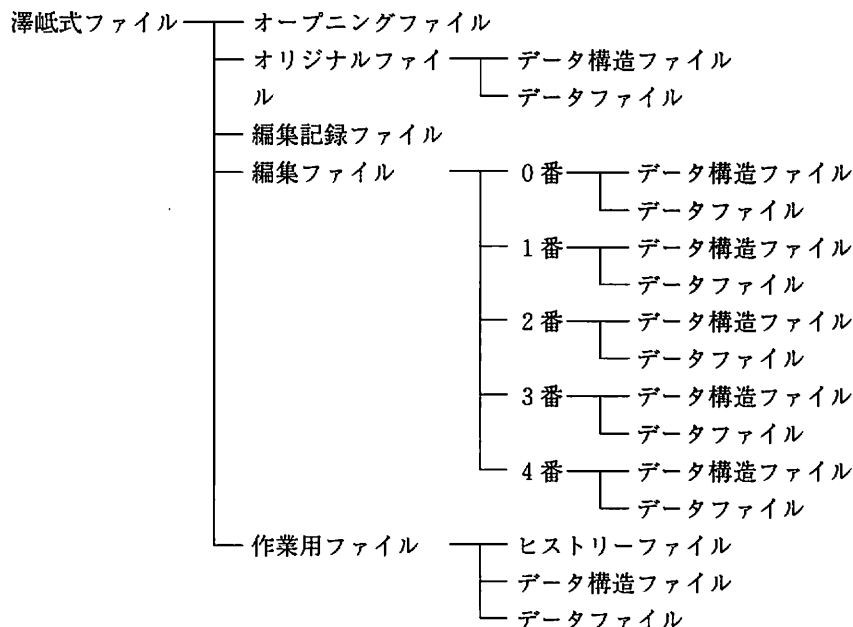
ひとつのファイル名につきファイル数は最大17個、ただし作業ファイルはプログラム実行時に作成され、終了時に削除される。

属性による分類

分 類	拡 張 子	最大ファイル数
オープニングファイル	なし	1
オリジナルファイル	DM,IM	2
編集記録ファイル	HIS	1
編集用ファイル	D*,I* (* = 0,1,2,3,4,5)	10
作業用ファイル	DT,IT,HISTORY TEMP	3

データ構造による分類

分類	ファイル名 (例: mit)
オープニングファイル	mit
データ構造ファイル	mit.I* (* = 0,1,2,3,4,M,T)
データファイル	mit.D* (* = 0,1,2,3,4,M,T)
編集記録ファイル	mit.HIS



<オープニングファイル>

DNA塩基配列編集プログラムやファイルコンバートプログラムで、澤岷式ファイルでファイルをオープンするときにはこのファイルを選択する。

わざわざオープニングファイルを用意したのには2つの理由がある。

ひとつには、澤岷式ファイルはファイル数が多すぎてファイルを選ぶのが面倒だからであるからであり、すぐに選びやすいファイル名をもつオープニングファイルを用意することで解決しようとした。

具体的に説明すると、村田式ファイルが1つで済むのに対し、澤岷式ファイルは最大で17のファイルが使われる。

例えば、ミトコンドリアのDNA塩基配列を扱った村田式ファイル mit.nuc がある。このファイルをDNA塩基配列編集プログラムで使用するために、澤岷式ファイルにコンバートすると、作成されるファイルは5つになる。

mit.DM mit.IM mit.HIS mit.D0 mit.I0 mit

また、編集が進むにつれて編集バージョンが0~4と増えて17個にもなる。この17個の中から拡張子を見分け、目的とするバージョンのファイルを選ぶのはかなり面倒である。しかし、オープニングファイルのように拡張子のないファイルを見つけだすのは、比較的容易である。

もうひとつの理由は、ファイル名を作成するのが容易になるからである。マッキントッシュ

版 Quick BASIC ではファイル名は次のような構造をもつ。

ファイル名 = フォルダ名 + ファイル名

フォルダ名とは MS-DOS でいうところのディレクトリ名 (パス名) に相当する。また、DNA 塩基配列編集プログラムでは MS-DOS の概念を持ち込みファイル名に拡張子を含んでいる。

ファイル名 = フォルダ名 + ファイル名 + 拡張子

例 → 6.07:DNAGE:mit:mit.I0

フォルダ名はコロン (:) で区切られていて上記の例では、“6.07”、“DNAGE”、“mit” がフォルダ名になる。拡張子はピリオド (.) で区切られていて上記の例では “I0” が拡張子になる。

ところで、上記の例で DNA 塩基配列編集プログラムでバージョン 0 番のファイルを編集したいときに扱うファイルは次の 7 つになる。

6.07:DNAGE:mit:mit	[オープニングファイル]
6.07:DNAGE:mit:mit.D0	[編集用データファイル 0 番]
6.07:DNAGE:mit:mit.I0	[編集用データ構造ファイル 0 番]
6.07:DNAGE:mit:mit.HIS	[編集記録ファイル]
6.07:DNAGE:mit:mit.IT	[作業用データファイル]
6.07:DNAGE:mit:mit.DT	[作業用データ構造ファイル]
6.07:DNAGE:mit:HISTORY TEMP	[作業用編集記録ファイル]

このとき “HISTORY TEMP” 以外のファイルのファイル名は、“mit” という部分が共通になっている。逆に言えば、“mit” の部分を知っていると拡張子の部分を変えるだけでファイル名が設定できることになる。

つまり、オープニングファイルでファイルをオープンすると、他のファイルはオープニングファイルのファイル名に拡張子の部分を加えるだけで設定ができる。

もしすべてのファイル名が拡張子を含むなら、一度はファイル名の部分と拡張子の部分に分ける処理をし、そのファイル名の部分に他のファイルの拡張子を加えることになり、手間がいくつかふえることになる。

<オリジナルファイル>

ファイルコンバータプログラムで村田式ファイルから澤舩式ファイルにコンバートされたときに作成されるデータ構造ファイルとデータファイル。

このファイルは編集によって変更されることがない。従って、はじめから編集をやり直したときのためのファイルとして利用する。

<編集記録ファイル>

編集処理を記録している。このファイルは編集記録参照プログラムで内容を見ることが出来る。

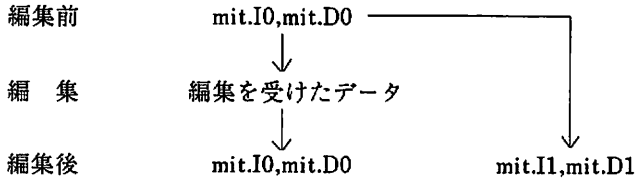
<編集用ファイル>

データ構造ファイルとデータファイルのうち、オリジナルファイルを除いたもの。拡張子の D や I の後に続く数字は編集バージョンである。その番号の値は 0 ~ 4 の範囲で、小さい番号ほど新しく編集されたファイルである。

例えば、村田式ファイル mit.nuc をファイルコンバートプログラムで澤舩式ファイルにコンバートした場合に、編集用ファイルは mit.I0 と mit.D0 のバージョン 0 番となる。

そのバージョン 0 番の編集用ファイルに編集を施しセーブすると、元の編集用ファイルバージョ

ン0番はバージョン1番になり、編集を施された新しいデータはバージョン0番としてセーブされる。



<作業用ファイル>

編集の時に一時的に作成されるファイル。

<データ構造ファイル> (拡張子の先頭がI)

データファイルに関する構造を記録。記録される構造は、種の数、種名、データ数、セクタ番号・セル先頭データ番号テーブルなど、データファイル（ランダムファイルアクセス）に必要なデータである。

<データファイル> (拡張子の先頭がD)

塩基配列データを記録。ランダムファイルになっていて、データ構造ファイルを利用してアクセスする。

②データ形式

(1) オープニングファイル

澤舐式ファイルチェック文字列	[例]	DNA GAP EDITOR
ファイル名	[例]	6.07:DNAGE:mit:mit
フォルダ名	[例]	6.07:DNAGE:mit
編集用ファイルのファイル数	[例]	2

(2) ヒストリーファイル

編集日付	} ×編集回数
編集回数	
編集モード	
種 (番号)	
塩基データ番号 (始点)	
編集した塩基データ (文字列)	

[例]

```

9 2 : 1 0 : 1 1
2
1
2
3 4
T C A
1
2
1 4
C C
  
```

9 2 : 1 0 : 1 2

1

2

3

1 4

T C

(3) データ構造ファイル

種の数

種名とデータ数 (× 種の数)

テーブル数

セル先頭データ番号テーブル、セクタ番号テーブル (×テーブル数) ×種の数

[例]

3

=== HUMAN M T 1630

=== BOVINE 1630

=== M OUSE 1630

9

1 1

201 2

401 3

601 4

801 5

1001 6

1201 7

1401 8

1601 9

9

1 10

201 11

401 12

601 13

801 14

1001 15

1201 16

1401 17

1601 18

9

1 19

201 20

401 21

601 22

801	23
1001	24
1201	25
1401	26
1601	27
0	0

(4) データファイル

ランダムファイルなのでレコードのフィールド構造を示す。

全フィールド長	256	バイト
主データ領域	200	バイト DNA塩基配列データ
予備データ領域	50	バイト
データ数	6	バイト

※挿入処理などで塩基データが主データ領域をこえた場合に、予備データ領域が使われる。それ以外の場合には文字'r'が書き込まれている。データ数は、そのレコード内の塩基データ数を表す。

4.2 澤岬式ファイルのデータアクセス

澤岬式ファイルのうち、塩基配列データが記録されているのはデータファイルで、そのファイルはランダムアクセス型ファイルになっている。そのデータファイルから任意のレコードを取り出すには、データ構造ファイルに記録してるデータ（テーブル）を使う。

データアクセスの説明のために、以下にランダムファイルの構造とプログラム上で用いるデータの呼び名の定義を示す。

[定義]

- ・データファイル（ランダムファイル）から一回に読み込む塩基データの単位をレコードと呼ぶ。
- ・レコードの物理的な番号をセクタと呼ぶ。
- ・各種のレコードの理論的な番号をセルと呼ぶ。

[構造]

次の設定で塩基データファイルの構造を図3-2.1に示す。

【設定】

- ・種は2種類
- ・第1種目の塩基データのデータ数が840で4個のレコードをもつ、セル数は4
- ・第2種目の塩基データのデータ数が880で4個のレコードをもつ、セル数は4

◎図3-2.1の読み取り方

【例】

- セクタ2 → 種1の2番目のレコード → 種1のセル2
- セクタ7 → 種2の4番目のレコード → 種2のセル4
- 種1のセル3を読み込む → セクタ3のレコードを読み込む
- 種2のセル2を読み込む → セクタ6のレコードを読み込む

セクタ	塩基配列データ (種、セル)	データ数	先頭
1	(1, 1)	2 2 0	1
2	(1, 2)	2 2 0	2 2 1
3	(1, 3)	2 2 0	4 4 1
4	(2, 1)	2 2 0	1
5	(2, 2)	2 2 0	2 2 1
6	(2, 3)	2 2 0	4 4 1
7	(2, 4)	2 2 0	6 6 1
8	(1, 4)	1 8 0	6 6 1

図3-2.1

1) レコード管理変数

ランダムアクセス型ファイルである塩基データファイルは、次のレコード管理変数を利用して読み書きする。(図3-2.1を参照)

①セル先頭データ番号テーブル

NumTBL(種、セル) → 任意の種・セルの先頭データ番号を表す。

[例]

種1・セル1の先頭データ番号 : NumTBL(1,1) → 1

種2・セル3の先頭データ番号 : NumTBL(2,3) → 441

②セクタ番号テーブル

SecTBL(種、セル) → 任意の種・セルのセクタを表す。

[例]

種1・セル2のセクタ : SecTBL(1,2) → 2

種2・セル3のセクタ : SecTBL(2,3) → 6

2) アクセス方法

レコード管理変数を用いたデータのアクセス方法を例を用いて説明する。

[例] 種2の442番目の塩基データを取り出したい場合

① NumTBL(n,m)を利用し、442番目のデータがどのセルにあるかを求める。

NumTBL(2,m)にはセルmの先頭データの番号が格納されているから、NumTBL(2,m)の最初(m=1)から順に値をみていく。もし442より大きい値のNumTBL(2,m)があれば、そのひとつ前のNumTBL(2,m-1)が442番目の塩基データが記録されているセルである。

実際に図3-2.1に沿って行くと、

m=1 : NumTBL(2,1) → 1 < 442

m=2 : NumTBL(2,2) → 221 < 442

m=3 : NumTBL(2,3) → 441 < 442

m=4 : NumTBL(2,4) → 661 > 442

となり、求めるセルは m-1 = 3 となる。

② SecTBL(n,m)を利用し、セル3のセクタを求める。

- ①で求めたセル 3 と種 2 を SecTBL(n,m) にあてはめ、442 番目のデータのあるセル 3 のセクタを求める。

SecTBL(2,3) → 6

これで、セクタ 6 に種 2 の 442 番目のデータが含まれていることがわかったのでそのレコードを GET コマンドで読み込むことにする。

GET #1,6

- ③ 読み込んだレコードから塩基データの部分を取り出す。

GET 命令で読み込んだレコードは fDNA\$ に格納される。

ファイル仕様でも説明したように、フィールド長 250 バイトの fDNA\$ のうち、塩基データはデータ数 fLength\$ 分である。残りの部分は "r" で埋められている。

そこで、次のようにして塩基データの部分だけを取り出す。

- (1) fLength\$ を数値に戻す。

Length = CVI(fLength\$)

- (2) 塩基データの長さの分だけ取り出す。

eDNA\$ = LEFT\$(fDNA\$,Length)

- ④ DNA 塩基配列編集プログラムでの記述

- ①と②を DNA 塩基配列編集プログラムで記述するとつぎのようになる。

```
SeaSpe=2 : sNum=442 : GOSUB SearchCell
gRec=SecTBL(SeaSpe,fCell) : GOSUB GetRecord
```

SearchCell:

Cell=1

WHILE NumTBL(SeaSpe,Cell) < sNum+1 : Cell = Cell + 1 : WEND

fCell=Cell - 1

RETURN

GetRecord:

GET #1,gRec

Length = LEN(eDNA\$)

eDNA\$ = LEFT\$(fDNA\$,Length)

RETURN

3) データの取り込み

DNA 塩基配列編集プログラムでは、1 回の読み込みで、ひとつの種あたりレコード 2 個分の塩基データを取り込む。

その塩基データは変数 dtDNA%(n,m) に取り込まれる。この変数のことを塩基データブロックと呼ぶ。また、塩基データブロックを読み込む時には、文字型の塩基データを整数値に変換して取り込む。文字型の変数では大きな配列がとれないからである。

整数値への変換は、次の通りである。

T → 1

C → 2

A → 3
G → 4
* → 5

変換の方法については「4.5 ユーザー定義関数の(1) FNvData(Cchr\$,P)」を参照。

[例] 次の種1、セル1・2のレコードを塩基データブロックに読み込む

セル1	CATTTA**CCCAAA**TAAA*GTATAGGCGATAGAAATT***GAAAC**CTGGCGCAAT
セル2	AGATATAGTACCGCAAGGGAAAGATGAAA**AATTATAACCAAGCATAATATAGCAAGGA

↓

塩基データ ブロック	231113552223335513335413134424313433311555543332552144242331 343131341322423344433343143335533113133223342313313134233443
---------------	--

4.3 編集処理

※説明の後ろにあるコロン(:)+番号はプログラムリストの行番号をさす。

[例] ~である。:234-233

通常QuickBASICには行番号はないが、プログラムリストを追い易くするために便宜上行番号を添えた。

1) 塩基データのサンプル (編集処理アルゴリズム説明用)

各編集処理のアルゴリズムの説明では以下に示す塩基データをサンプルとして使用する。(編集を施す前の状態を次のサンプルのように設定する)

ことわり:

このサンプルデータは、各編集処理の説明を簡単にするために1レコードあたりの塩基データ数を60とした。実際には、1レコードの塩基データ数は最大で250、最小で101になるような処理がなされるので60になることはない。

サンプルデータ構造

種名=HUMAN : 種の数=1 : 種1のデータ数=360 : レコード数=6
最終セクタ=6

セクタ	塩基データ	データ数
1	GCTAAACCTAGCCCCAAACCCACTCCACC*****TTACTACCAGACAACCTTAGCCAAAC	60
2	CATTTA**CCCAAA**TAAA*GTATAGGCGATAGAAATT***GAAAC**CTGGCGCAAT	60
3	AGATATAGTACCGCAAGGGAAAGATGAAA**AATTATAACCAAGCATAATATAGCAAGGA	60
4	CTAACCCCTATACCTTCTGCATAATGAATTAAGTAGAAA*TAACCTTTCGAAGGAGAGCCA	60
5	AAGCTAAGACCCCCGAAACCAGACGAGCTACCTAAGAACAGC**TAAAAGAGCACACCCG	60
6	TCTATGTAGCAAAATAGTGGGAAGATTTATAGGTAGAGGCGACAAACCTACCGAGCCTGG	60

テーブル

```

NumTBL(1,1) = 1      :      SecTBL(1,1) = 1
NumTBL(1,2) = 61    :      SecTBL(1,2) = 2
NumTBL(1,3) = 121   :      SecTBL(1,3) = 3
NumTBL(1,4) = 181   :      SecTBL(1,4) = 4
NumTBL(1,5) = 241   :      SecTBL(1,5) = 5
NumTBL(1,6) = 301   :      SecTBL(1,6) = 6
    
```

塩基データブロック (dtDNA%)

塩基データブロック (変数dtDNA%) にはセル 2・3 の 2 レコードが読み込まれているものとする。よってdtDNA%(1,1)は 2、TopNumDB(1)は 61 である。

※塩基データブロック (dtDNA%) は塩基が数値として格納されている。

(T->1,C->2,A->3,G->4,*->5)

dtDAN%とセル 2・3

```

CATTTA**CCCAAA**TAAA*GTATAGGCGATAGAAATT***GAAAC**CTGGCGCAAT < セル 2
231113552223335513335413134424313433311555543332552144242331 < dtDNA%(1,
1~60)
AGATATAGTACCGCAAGGGAAAGATGAAA**AATTATAACCAAGCATAATATAGCAAGGA < セル 3
343131341322423344433343143335533113133223342313313134233443 < dtDNA%(1,
61~120)
    
```

2) 編集処理全体における説明

①編集範囲について

一回の編集処理で編集を施すことができるのは、編集ウィンドウに表示されている60個の塩基データの範囲内である。ただし、挿入処理では1度に20個以上の塩基データを扱えない。

また2種以上にまたがる編集処理はできない。よって編集範囲の指定は同一種上の塩基データで行うのに限られる。

[例]

○可能な範囲指定

```

                                左端                                右端
                                v                                v
SYU1  CATTTA**CCC AAA**TAAA*GTATAGG CGATAGAAATT***
SYU2  AGATATAGTAC CGCAAGGGAAAGATGAA A**AATTATAACCA
    
```

×不可能な指定範囲

```

                                左端
                                v
SYU1  CATTTA**CCC AAA**TAAA*GTATAGG CGATAGAAATT***
SYU2  AGATATAGTAC CGCAAGGGAAAGATGAA A**AATTATAACCA
                                右端
    
```

②セーブフラグについて

編集を施した後に、ファイルセーブせずDNA塩基配列編集プログラムの終了やファイルクローズを行おうとするユーザーに、セーブの確認を行うためのフラグである。つまり、編集を行ったのにセーブを忘れるようなことをさせない処理をするためのフラグである。

このフラグは編集処理をするとオンなる。そしてフラグがオンの状態（編集を施した）でプログラムを終了したり、ファイルクローズを行うとセーブの確認をとる。また、いったんセーブを行えばオフになる。

③編集フラグについて

編集画面処理をするルーチン用のフラグである。編集ウィンドウがオープンの状態では、マウスカーソルの位置・ボタン入力待ちなどの処理をする。これらの処理はこのフラグがオンのときに行われる。

このフラグは、ファイルオープンして塩基データを読み込んだときにオンになり、ファイルクローズやプログラムの終了のときにオフになる。

3) アンドゥ処理

【機能】 編集処理を行った前の状態に戻す。

【アルゴリズム説明】

ひとつ前の編集処理に応じて次の処理を行う（各処理ルーチンを呼び出す）。

ひとつ前の処理	アンドゥで行う処理
カット処理	ペースト処理。カットした塩基データを元の位置に挿入する
コピー処理	なにもしない
ペースト処理	カット処理。ペーストした塩基データをカットする
挿入処理	カット処理。挿入した塩基データをカットする

※アンドゥフラグについて

アンドゥ処理の場合でも、通常の編集処理（プルダウンメニューで呼び出す場合）の場合でも呼び出すルーチンは同じである。通常、各編集処理のルーチンを呼び出すと、処理を施す範囲・位置を指定する処理を行う。しかしアンドゥ処理では処理を施す位置は決まっている（ひとつ前に処理を施した位置や範囲）ので、その処理は必要ない。そこでこのフラグを使って、通常の処理とアンドゥ処理の区別する。

つまりこのフラグがオフの場合には、処理を施す範囲・位置を指定する処理を行い、オンの場合にはその処理を行わずに、ひとつ前の処理で施した範囲・位置に編集処理を行う。

4) コピー処理

【機能】 マウスクリックで指定した範囲（左端と右端の間）の塩基データを編集クリップボードにコピーする

【アルゴリズム説明】

【設定】

反転表示した塩基データの部分をクリップボードにコピー処理する。

:61

HUMAN CA TTTA**CC CAAA**TAAA*GTATAGGCGATAGAAATT****GAAAC**CTGGCGCAAT

種=1 : 左端=63 : 右端=70 : コピーする塩基データ数=7

【流れ】

指定された塩基データの部分を塩基データブロックから取り出し、クリップボードにコピーする。

(次の①～②の処理を左端から右端まで行っている。) :424-427

①塩基データブロック (dtDNA%) 内での位置を計算する。:425

計算式=塩基のデータ番号 - 塩基データブロックの先頭データ番号 + 1

[例:左端の場合]

$$63 - 61 + 1 = 3 \rightarrow WK2$$

②もとの位置のデータ (数値) を文字型に変換してクリップボード (buf\$) に加える。:426

変換式 MID\$ ("TCAG*",dtDNA%(1,WK2),1)

上の変換式は BASIC の関数を利用したもので、dtDNA%(1,WK2)が1なら“T”に、2なら“C”になる。

上の①②の処理を表であらわすと次のようになる。

塩基データブロック (dtDNA%(1,1) ~ dtDNA%(1,60))

23 11135522 23335513335413134424313433311555543332552144242331

	WK2	dtDNA%(1,WK2)	buf\$
63	3	1	T
64	4	1	TT
65	5	1	TTT
66	6	3	TTTA
67	7	5	TTTA*
68	8	5	TTTA**
69	9	2	TTTA**C
70	10	2	TTTA**CC

<= 左端

<= 右端

5) カット処理

【機能】 マウスクリックで指定した範囲 (左端と右端の間) の塩基データをカットする。

【アルゴリズム説明】

【設定】

反転表示した塩基データの部分をカット処理する。

:61

HUMAN CA TTTA**CC CAAA**TAAA*GTATAGGCGATAGAAATT****GAAAC**CTGGCGCAAT

種 = 1 : 左端 = 63 : 右端 = 70 : カットする塩基データ数 = 7

【流れ】

- ①左端の塩基データがどのセルにあるかを検索する [SerachCell]。(セル検索については「4.2 澤岷式ファイルのデータアクセス 2) アクセス方法」を参照):364
サンプルデータからセル 2 があることがわかる。
- ②クリップボードに、カットする塩基データをコピーする。(このアルゴリズムは、「4) コピー編集」と同じである。):366-369
- ③カットする塩基データが2つのセルにまたがっているかどうかを判断する。:370
- ④-a : ひとつのセル内におさまっている場合 :371-374
- (1) セル 2 から 1 レコードの塩基データを読み込む。 → eDNA \$ → wDNA \$:371
- (2) カットする塩基データ (反転表示部分) から左の部分のデータ数を計算する :372
(4.5 ユーザー定義関数を参照)
計算式 1 = 左端 - セル 2 の先頭データ番号
= 63 - 61 = 2 → WK
- (3) カットする塩基データ (反転表示部分) から右の部分のデータ数を計算する。:372
計算式 2 = セル 2 のデータ数 - 右端のひとつ前のデータの位置 - 1
= 60 - 9 - 1 = 50 → WK2
- (4) 次のように、カットしない部分を取り出してつなぎ合わせれば、結果としてカットされたことになる。:373

CA **TTTA*CC** CAAA**TAAA*GTATAGGCGATAGAAATT***GAAAC**CTGGCGCAAT (wDNA \$)

↓ ↓

CA + CAAA**TAAA*GTATAGGCGATAGAAATT***GAAAC**CTGGCGCAAT
= CATTTA**CCCAAAA**TAAA*GTATAGGCGATAGAAATT***GAAAC**CTGGCGCAAT

反転部分から左の部分は LEFT \$ (wDNA \$, 2) : wDNA \$ の左から 2

反転部分から右の部分は RIGHT \$ (wDNA \$, 50) : wDNA \$ の右から 50

カットした塩基データは LEFT \$ (wDNA \$, 2) + RIGHT \$ (wDNA \$, 50)

- (5) この塩基データをレコードとしてセル 2 に書き込めばファイルも更新されたことになる。:374

- ④-b : 2つのセルにまたがっている場合 :376-382

・今回の設定ではひとつのセルにおさまってしまっているため、説明のために、仮に次の位置にセル 2 と 3 の境界があるとする。

↓

CATTTA**CCCAAAA**TAAA*GTATAGGCGATAGAAATT***GAAAC**CTGGCGCAAT

するとセル 2・3 は次のようになる

セル	塩基データ	範囲
2	CA TTTA*	61 - 67
3	*CC CAAA**TAAA*GTATAGGCGATAGAAATT***GAAAC**CTGGCGCAAT	68 - 120

・上記の条件で①～③の処理が行われたとしてアルゴリズムを考えていく。するとカットする塩基データ“TTTA* *CC”のうち、“TTTA*”はセル2から、“*CC”はセル3からカットするとよいことがわかる。

(1) セル2から1レコード塩基データを読み込む。 → eDNA\$ → wDNA\$:376

(2) カットする塩基データ（反転表示部分）から左の部分のデータ数を計算する :377

$$\begin{aligned} \text{計算式 1} &= \text{左端} - \text{セル2の先頭データ番号} \\ &= 63 - 61 = 2 \end{aligned}$$

(3) カットするのは反転表示部分であるから、読み込んだ塩基データ（wDNA\$）の反転表示部分から左の部分だけを残せばよい。:377

反転表示部分から左の部分は LEFT\$(wDNA\$,2)

この塩基データ（wDNA\$）をレコードとしてセル2に書き込めばファイルが更新されたことになる。:378

(4) セル3から1レコード塩基データを読み込む。 → eDNA\$ → wDNA\$:379

(5) カットする塩基データ（反転表示部分）から右の部分のデータ数を計算する。:380

$$\begin{aligned} \text{計算式 2} &= \text{セル3のデータ数} - \text{右端のひとつ前} - 1 \\ &= 53 - 2 - 1 = 50 \end{aligned}$$

(6) カットするのは反転表示部分であるから、読み込んだ塩基データ（wDNA\$）の反転表示部分から右の部分だけを残せばよい。:381

反転から右の部分は RIGHT\$(wDNA\$,50)

この塩基データ（wDNA\$）をレコードとしてセル3に書き込めばファイルが更新されたことになる。382

⑤カット処理によって減った塩基データ数の分、セル先頭データ番号テーブルを更新する。:384

・カット処理で減った塩基データ数は7になる。そしてカット処理でセルの先頭データ番号が変わったのはセル3からセル6（最後のセル）までである。よって更新するセル先頭データ番号テーブルは3から6である。

$$\text{更新式: セル先頭データ番号} = \text{セル先頭データ番号} - 7$$

セル先頭データ番号更新(④-aの場合)

セル	1	2	3	4	5	6
更新前のセル先頭データ番号	1	61	121	181	241	301
更新後のセル先頭データ番号	1	61	114	174	234	294

6) ペースト処理

【機能】 マウスクリックで指定した位置に、クリップボードに格納されている塩基データを挿入する

【アルゴリズム説明】

【設定】

クリップボードに塩基データ“TCCGAT”が格納されているとき、次の位置にペースト処理する。


```

クリップボード > "TCCGAT"
                :61      v
HUMAN  CATTTA**C CCAAAA**TAAA*GTATAGGCGATAGAAATT****GAAAC**CTGGCGCAAT
種= 1   :   挿入点= 6 8   :   ペーストする塩基データ数= 6

```

【流れ】

- ①クリップボードからペーストする塩基データを取り込む。 buf \$ → ins \$:436
- ②ペーストする位置にある塩基データがどのセルにあるかを検索する [SerachCell]。(セル検索については「4.2 澤舩式ファイルのデータアクセス 2) アクセス方法」を参照) サンプルデータからセル2にあることがわかる。:603
- ③セル2から1レコードの塩基データを読み込む。 → eDNA \$:604
- ④ペーストする位置(反転表示)から左の部分の長さを求める。:605
 計算式 = ペーストする位置 - セル2の先頭データ番号
 = 6 8 - 6 1 = 7
 ・今計算したのは次の下線部分の塩基データになる。

```

CATTTA**C CCAAAA**TAAA*GTATAGGCGATAGAAATT****GAAAC**CTGGCGCAAT (eDNA $)

```

- ⑤上の塩基データ(eDNA \$)を下線の部分とそうでない部分に分け、その間にペーストする塩基データをはさみこむとでペースト処理をする。:606
 下線の部分は LEFT \$ (eDNA \$,7)
 下線より右の部分は RIGHT \$ (eDNA \$,60-7)
 ペーストする塩基データは ins \$
 ペーストした塩基データは LEFT \$ (eDNA \$,7) + ins \$ + RIGHT \$ (eDNA \$,60-7)
 CATTTA** + TCCGAT + CCAAAA**TAAA*GTATAGGCGATAGAAATT****GAAAC**CTGGCGCAAT
- ⑥ペーストした後の塩基データのデータ数が255以上(セルのサイズをこえる)であるかを判断する。:607
- ⑦-a : ペーストした塩基データのデータ数が255以上の場合(セルのサイズをこえる)。
 :608-615
 ・①~⑥までの処理の結果、セル2のデータ数は66であり、255をこえない。そこで説明の為に、セルのサイズを60と設定する。
 そうすると、ペーストした塩基データ(eDNA \$)は、次のようにセルのサイズを6だけ越えたことになる。

```

                                     セルの境界
                                     ↓
CATTTA**TCCGATCCCAAAA**TAAA*GTATAGGCGATAGAAATT****GAAAC**CTGG CGCAAT
(eDNA $)

```

- (1) ペーストした塩基データを2つに分ける。:608-609
 ※実際の処理では、255の約半分の125の長さ分を、ペーストした塩基データの右側から取り出してセル2に書き込み、残った部分は、新しくセルを作成し、そのセルに書き込む。

・いまはセルのサイズを60としているので、60の約半分の30の長さ分と残った36の部分にわける。

```
CATTTA**TCCGATCCCAAA**TAAA*GTATAGCGCATAGAAATT***GAAAC**CTGGCGCAAT
      ↓                               ↓
      30                               36
```

右の30は LEFT \$(eDNA \$,30)

左の36は RIGHT \$(eDNA \$,66-30)

(2) 右の30をセル2に、左の36を新しいセルに書き込む。新しいセルは今までの最終セクタ6の続きのセクタ7になる。そして最終セクタを7とする。:610-612

・ところで、この左の36はセル2の続きの部分であるから、新しいセルはセル2の次のセルつまりセル3になるべきである。そこで、今までのセル3はセル4とし、セル4はセル5とする更新(シフト)が必要になってくる。

(3) 元のセル3からセル6までの各テーブルを更新(シフト)する。

```
セル6の先頭データ番号 (3 0 1) → セル7の先頭データ番号
セル5の先頭データ番号 (2 4 1) → セル6の先頭データ番号
セル4の先頭データ番号 (1 8 1) → セル5の先頭データ番号
セル3の先頭データ番号 (1 2 1) → セル4の先頭データ番号
セル6のセクタ番号 (6) → セル7のセクタ番号
セル5のセクタ番号 (5) → セル6のセクタ番号
セル4のセクタ番号 (4) → セル5のセクタ番号
セル3のセクタ番号 (3) → セル4のセクタ番号
```

(4) 新しいセル3のテーブルを設定する :613

```
セル3の先頭データ番号 ← 9 1
セル3のセクタ番号 ← 7
```

(5) ベースト処理で増えた塩基データ数の分、セル先頭データ番号テーブルを更新する。

```
セル7の先頭データ番号 ← 3 0 1 + 6 = 3 0 7
セル6の先頭データ番号 ← 2 4 1 + 6 = 2 4 7
セル5の先頭データ番号 ← 1 8 1 + 6 = 1 8 7
セル4の先頭データ番号 ← 1 2 1 + 6 = 1 2 7
```

(6) (1)~(5)までの結果

セクタ	セル	先頭	塩基データ
1	1	1	GCTAAACCTACCCCAAACCCACTCCACC*****TTACTACCAGACAACCTTAGCCAAAC
2	2	61	CATTTA**TCCGATCCCAAA**TAAA*GTA
3	4	127	AGATATAGTACCGCAAGGGAAGATGAAA**AATTATAACCAAGCATAATATAGCAAGGA
4	5	187	CTAACCCCTATACCTTCTGCATAATGAATTAAGTAACTAGAAA*TAACCTTTCGAAGGAGAGCCA

5	6	247	AAGCTAAGACCCCGAAACCAGACGAGCTACCTAAGAACAGC**TAAAAGAGCACACCCG
6	7	307	TCTATGTAGCAAATAGTGGGAAGATTTATAGGTAGAGGCGACAAACCTACCGAGCCTGG
7	3	91	TAGGCGATAGAAATT***GAAAC**CTGGCGCAAT

⑦-b: ペーストした塩基データのデータ数が255未満の場合(セルのサイズをこえない)。
:617-618

- (1) セル2にペーストした塩基データ(eDNA \$)を書き込む。
- (2) ペースト処理で増えた塩基データ数の分、セル先頭データ番号テーブルを更新する。
セル3の先頭データ番号 ← 121 + 6 = 127
セル4の先頭データ番号 ← 181 + 6 = 187
セル5の先頭データ番号 ← 241 + 6 = 247
セル6の先頭データ番号 ← 301 + 6 = 307

7) 挿入処理

[機能] 塩基配列データを作成し、その塩基データをマウスクリックで指定した位置に挿入する。

[アルゴリズム説明]

挿入処理のアルゴリズムはペースト処理とほぼ同様である。ペースト処理の場合には、挿入する塩基データをクリップボードから取り込む。それに対し挿入処理は、任意に作成した塩基データを挿入する。

なお、任意に塩基データを作成する処理については、「5.1 DNA塩基配列編集プログラムのSPD」から[GetInsData]を参照。またその塩基データを挿入する処理については、「4.3 編集処理 6) ペースト処理」の【流れ】②~⑦を参照せよ。

4.4 ヒストリー(編集記録)

DNA塩基配列編集プログラムでは編集の過程を編集記録ファイルに記録している。その編集記録は、次のような過程で記録される。

- ①ファイルオープンの時、新規にテンポラリファイルを開く。
- ②各編集処理後、次の形式でテンポラリファイルに記録し、その編集回数はカウントしておく。
編集モード,種(番号),塩基データ番号(始点),編集した塩基データ(文字列)
- ③ファイルセーブの時、追加モードで編集記録ファイルを開く。もしセーブせず終了やクローズを強行実行した場合には、その時点でテンポラリファイルを削除する。
- ④次の項目をヒストリーファイルに書き込む
編集日付,編集回数
- ⑤次のデータを、テンポラリファイルから編集記録ファイルにコピーする。
編集モード,種(番号),塩基データ番号(始点),編集した塩基データ(文字列)
- ⑥テンポラリファイルを削除する。

4.5 ユーザー定義関数

DNA塩基配列編集プログラムでは次の3つのユーザー定義関数を定義している。

- (1) 塩基配列の任意の位置にある文字型の塩基データを整数値に変換する

[定義式] DEF FNvData(Cchr\$,P) = INSTR ("TCAG*",MID\$(Cchr\$,P,1))

[引き数] Cchr\$: 文字型の塩基データ

P : 文字型の塩基データの位置 (この文字型データを整数値に変換する)

[処理] T → 1

C → 2

A → 3

G → 4

* → 5

[例] 塩基データ“TTCAGA”があるとき、3番目のデータ“C”を整数値2に変換する。

Cchr\$ = “TTCAGA” : P = 3

FNvData(Cchr\$,3) → 2

- (2) 任意の塩基データがセルのどの位置にあるかを計算する。(計算結果に1を加えた値がその位置になる)

[定義式] DEF FNpData(KoSpe,NoCell,dataPoint) = dataPoint - NumTBL(KoCell, NoCell)

[引き数] KoSpe : 種 (番号)

NoCell : セル

dataPoint : 塩基データ番号

[例] 種2のセル3には、201番目から400番目の塩基データがあるとして、325番目の塩基データのセル内での位置を求める。

KoSpe=2 : NoCell=3 : dataPoint=325

Point = FNpData(KoSpe,NoCell,dataPoint) → 124

DPinCell = Point + 1

計算 : 325 - 201 = 124 → セル内での位置 = 124 + 1 = 125

- (3) 任意のセルのデータ長を計算する。

[定義式] DEF FNnData(KoSpe,NoCell) = NumTBL(KoSpe,NoCell) - NumTBL(KoSpe,NoCell)

[引き数] KoSpe : 種 (番号)

NoCell : セル

[例] 種2のセル2の先頭データが201番目、セル3の先頭データが421番目として、セル2のデータ長を計算する。

KoSpe=2 : NoCell=2

CellLength = FNnData(KoSpe,NoCell) → 220

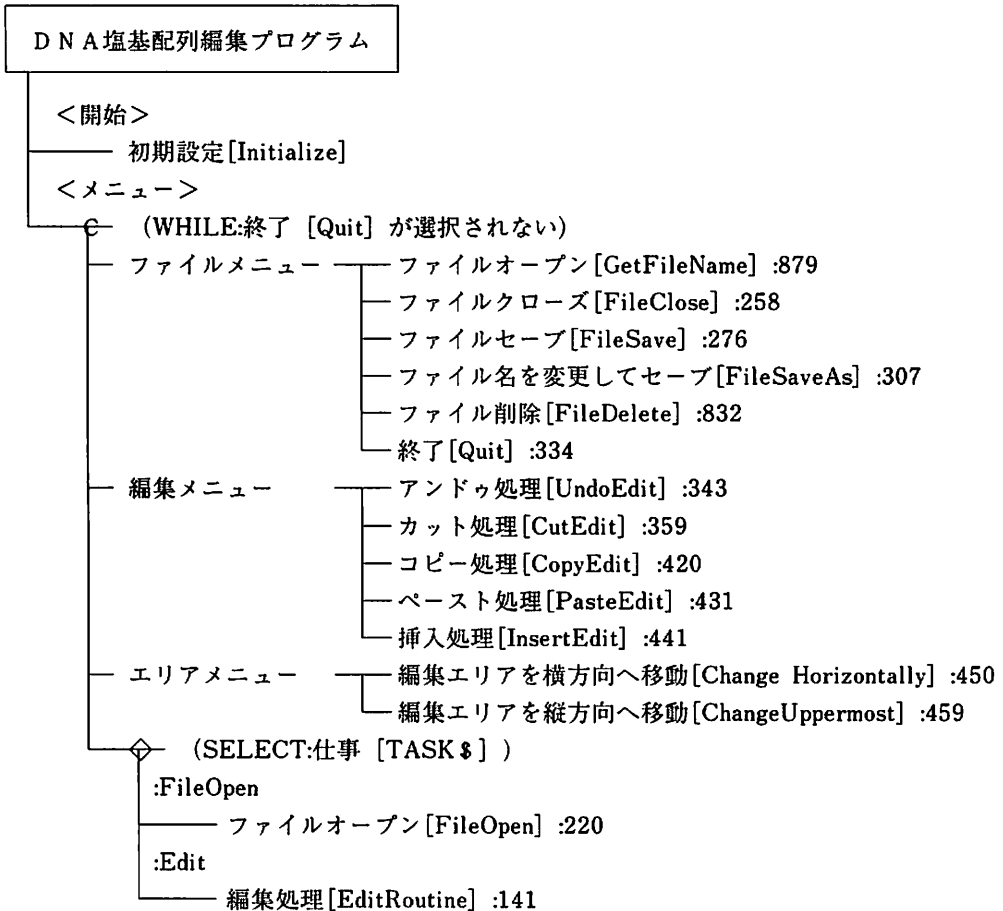
計算 : 421 - 201 = 220 → セル2のデータ長

プログラムのSPD

5.1 DNA塩基配列編集プログラム

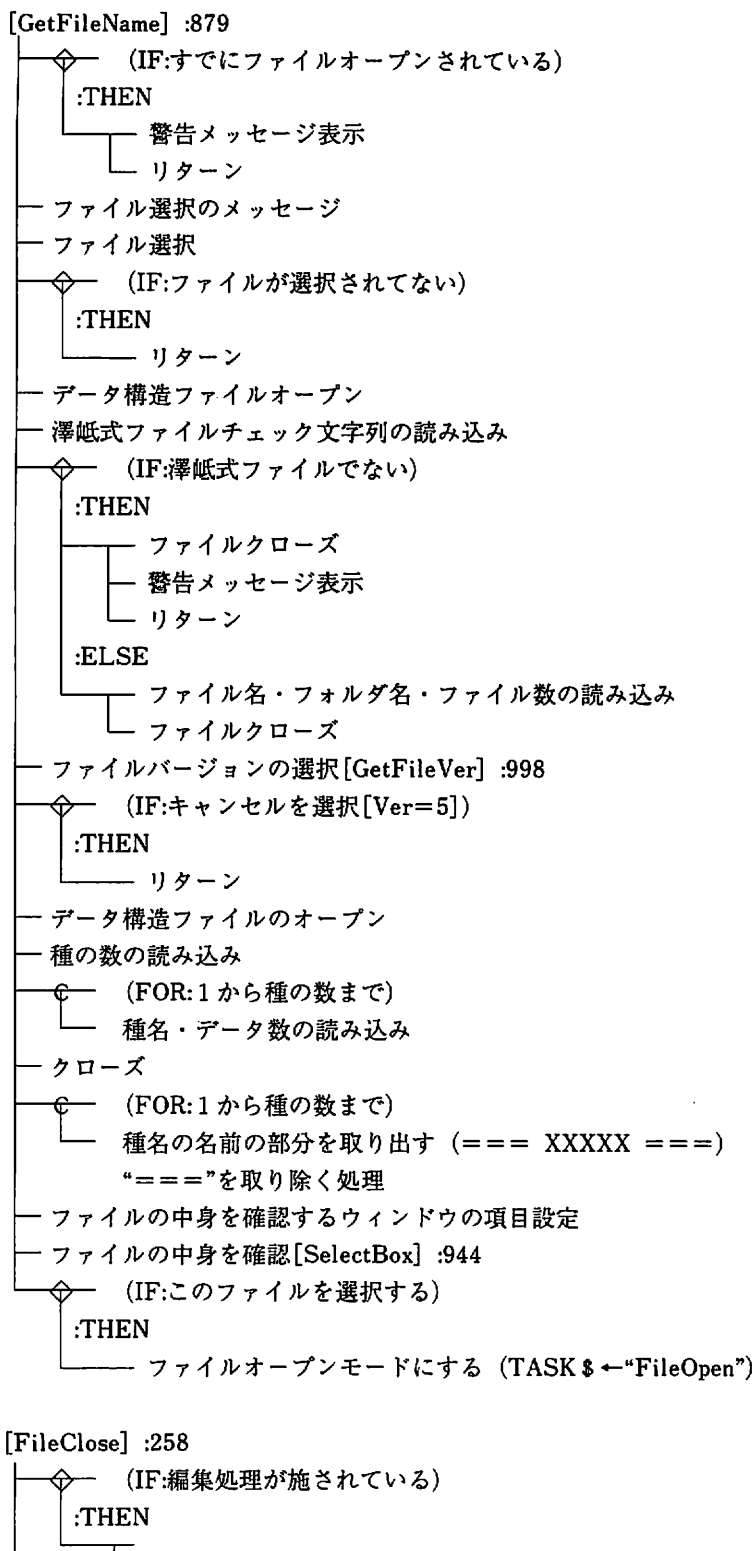
※ルーチン名の後ろの番号はプログラムリストの行番号をさす。

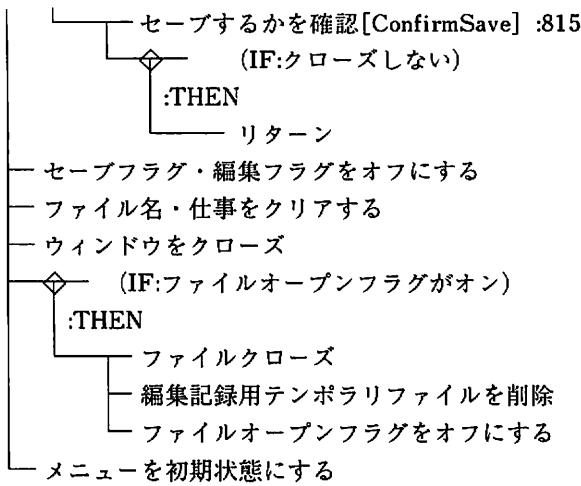
通常QuickBASICには行番号はないが、プログラムリストを追い易くするために便宜上行番号を添えた。



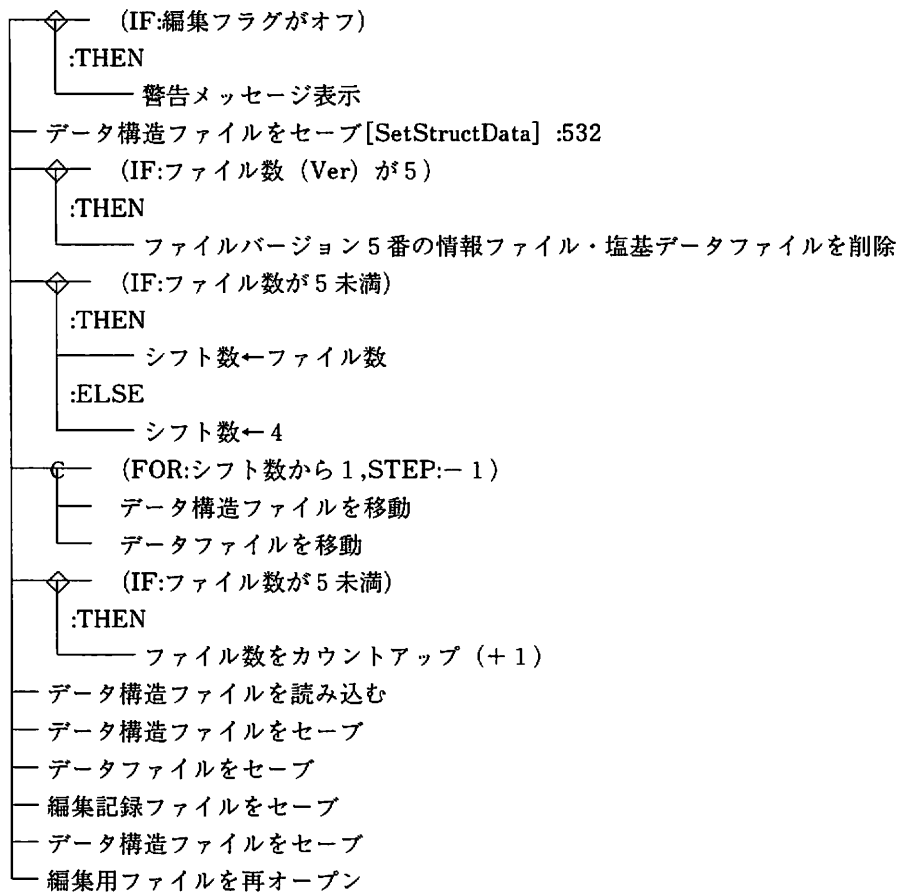
[Initialize]

- 編集可能な種数の上限を設定 :15
- 配列の設定 :17-20
- 定数の設定 :22-28
- ファイル拡張子の設定 :34-38
- メニューの設定 :40-60

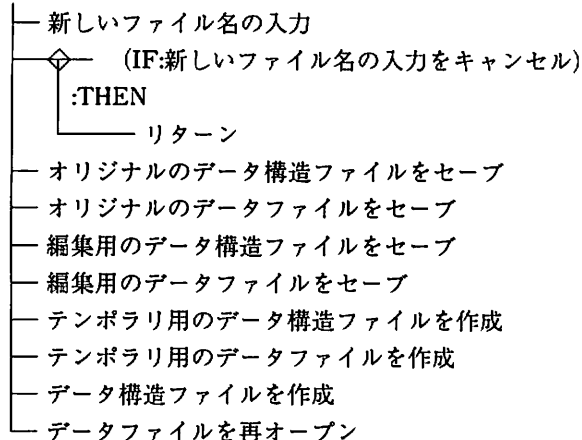




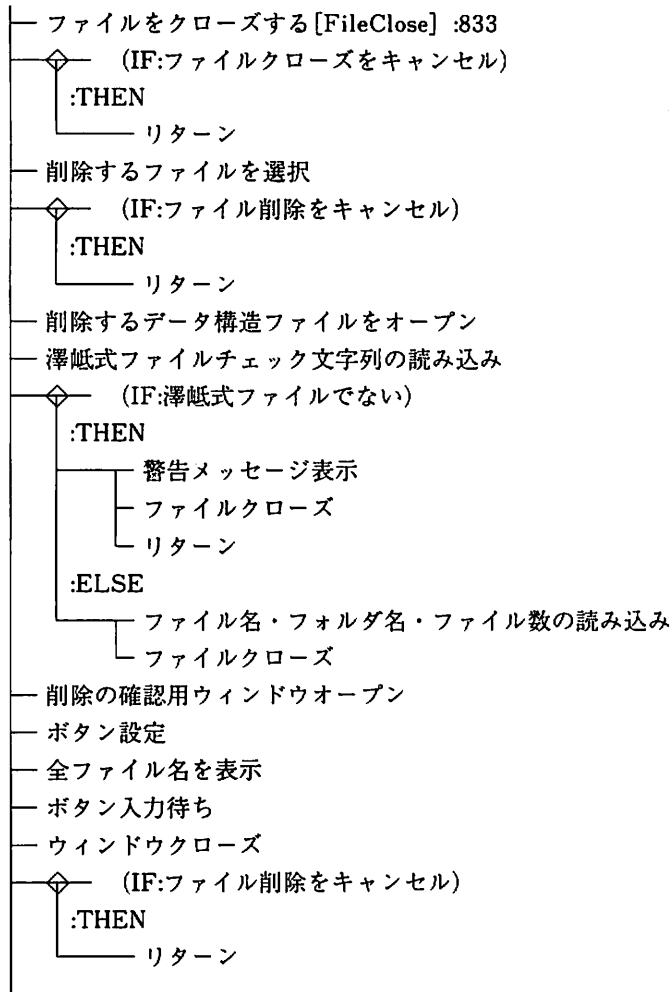
[FileSave] :276



[FileSaveAs] :307

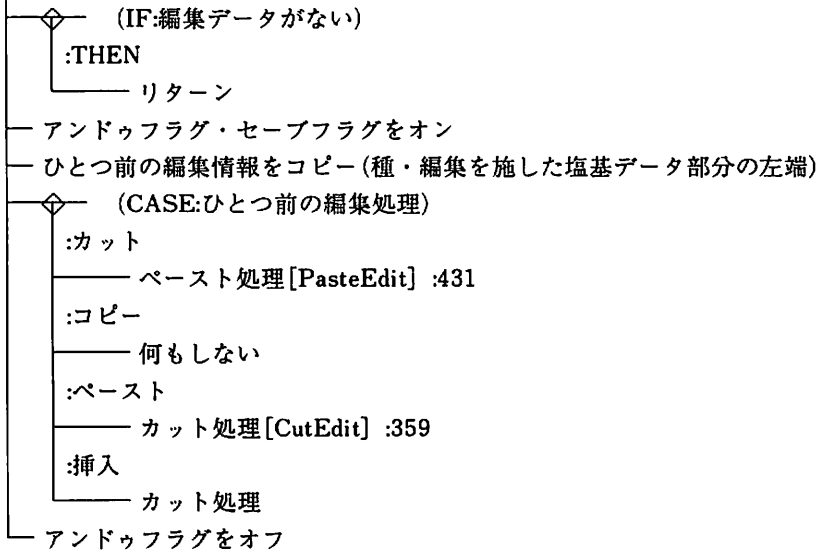


[FileClose] :258

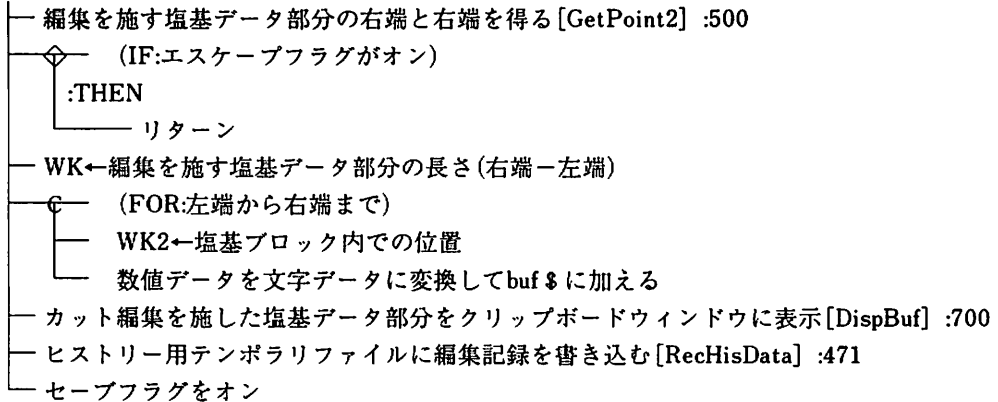


└─ 全ファイルを削除

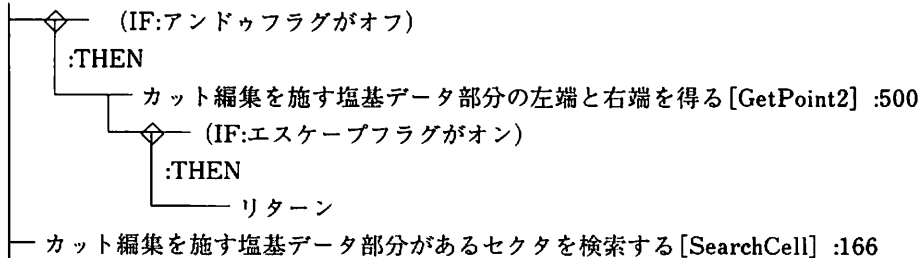
[UndoEdit] :343

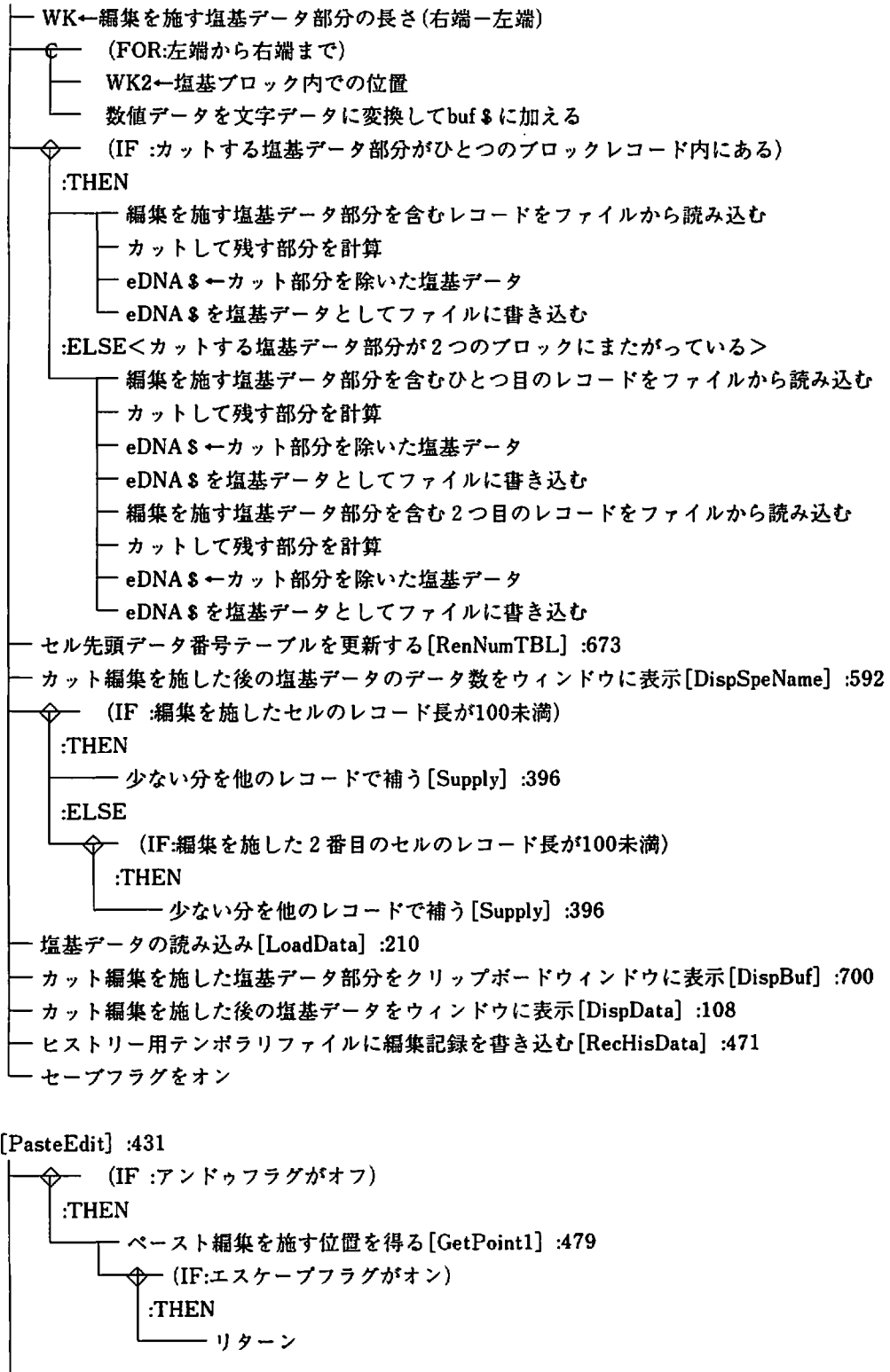


[CopyEdit] :420



[CutEdit] :359





- 挿入をする [InsRecord] :602
- 塩基データの読み込み [LoadData] :210
- カット編集を施した塩基データをクリップボードウィンドウに表示 [DispBuf] :700
- ペースト編集を施した後の塩基データをウィンドウに表示 [DispData] :108
- ヒストリー用テンポラリファイルに編集記録を書き込む [RecHisData] :471
- セーブフラグをオン

[InsertEdit] :441

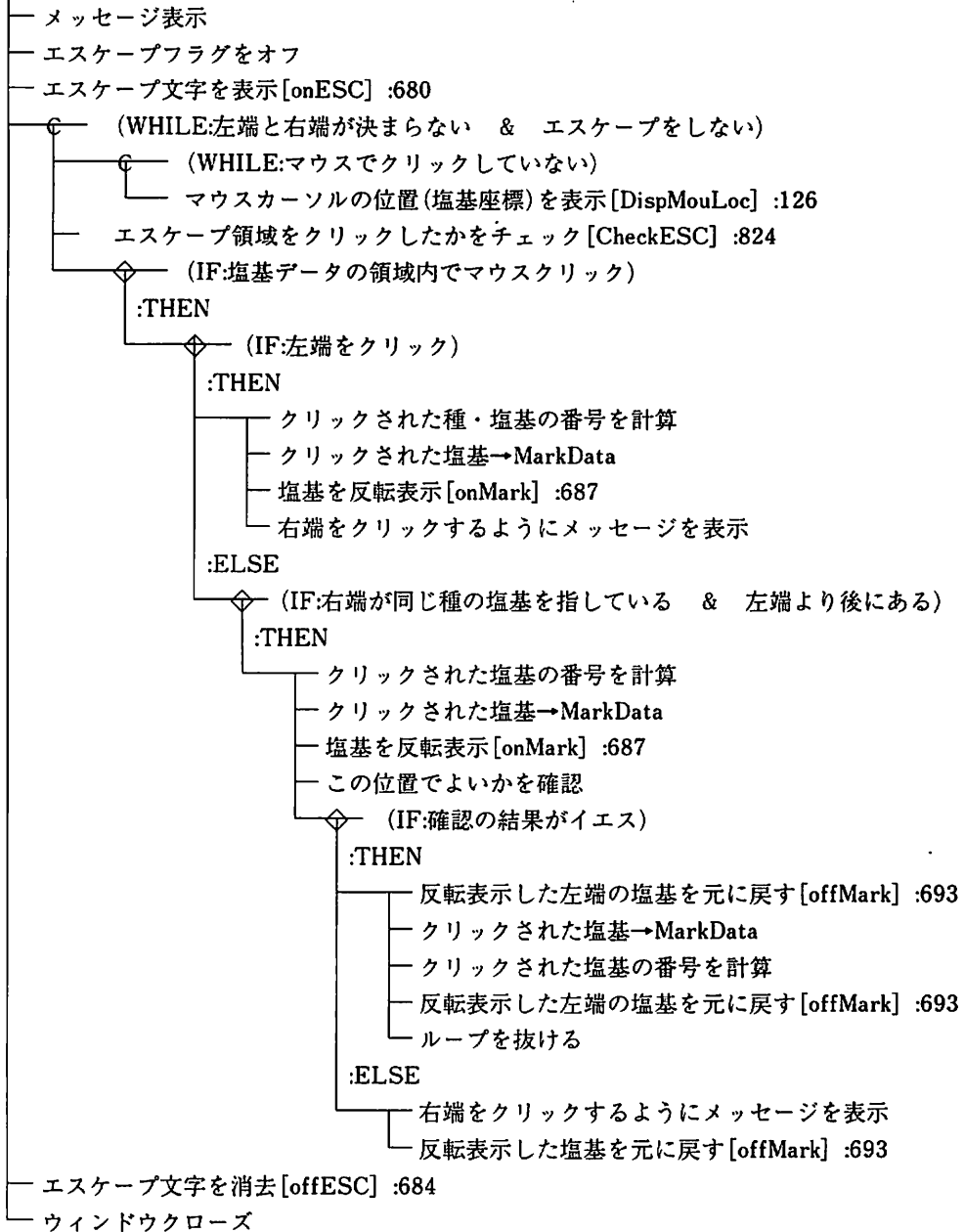
- 挿入編集を施す位置を得る [GetPoint1] :479
 - ◇ (IF:エスケープフラグがオン)
 - :THEN
 - リターン
- 塩基データを挿入する位置の塩基を反転表示する [onMark] :687
- 挿入する塩基データを得る [GetInsData] :624
- 挿入する [InsRecord] :602
- 塩基データの読み込み [LoadData] :210
- 挿入編集を施した塩基データをクリップボードウィンドウに表示 [DispBuf] :700
- 挿入編集を施した後の塩基データをウィンドウに表示 [DispData] :108
- テンポラリ用編集記録ファイルに編集記録を書き込む [RecHisData] :471
- セーブフラグをオン

[GetPoint1] :479

- メッセージ表示
- エスケープフラグをオフ
- エスケープ文字を表示 [onESC] :680
- ◇ (WHILE:編集を施す位置が決まらない & エスケープをしない)
 - ◇ (WHILE:マウスでクリックしていない)
 - マウスカーソルの位置(塩基座標)を表示 [DispMouLoc] :126
 - エスケープ領域をクリックしたかをチェック [CheckESC] :824
 - ◇ (IF:塩基データの領域内でマウスクリック)
 - :THEN
 - クリックされた種・塩基の番号を計算
 - クリックされた塩基→MarkData
 - 塩基を反転表示 [onMark] :687
 - この位置でよいかを確認
 - ◇ (IF:確認の結果がイエス)
 - :THEN
 - ループを抜ける
 - :ELSE
 - 反転表示した塩基を元に戻す [offMark] :693
- 反転表示した塩基を元に戻す [offMark] :693

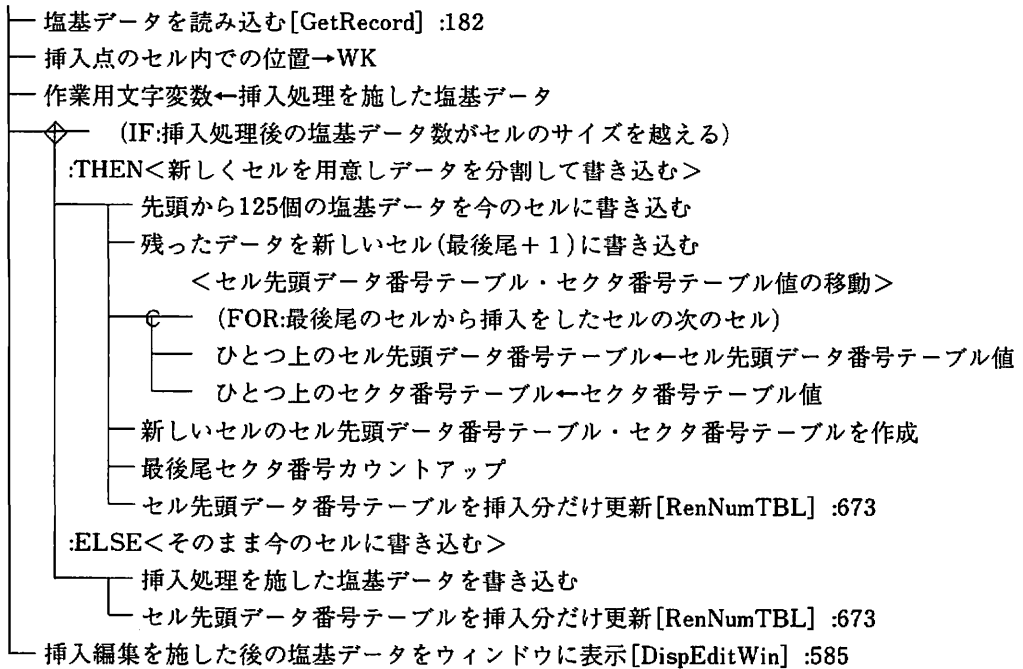
— エスケープ文字を消去 [offESC] :684
 — ウィンドウクローズ

[GetPoint2] :500

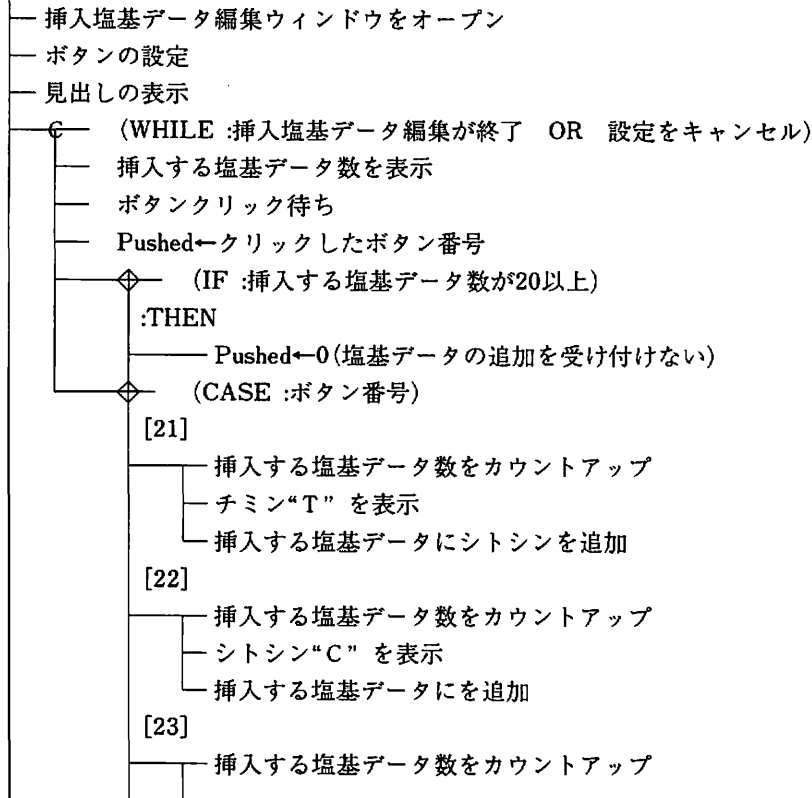


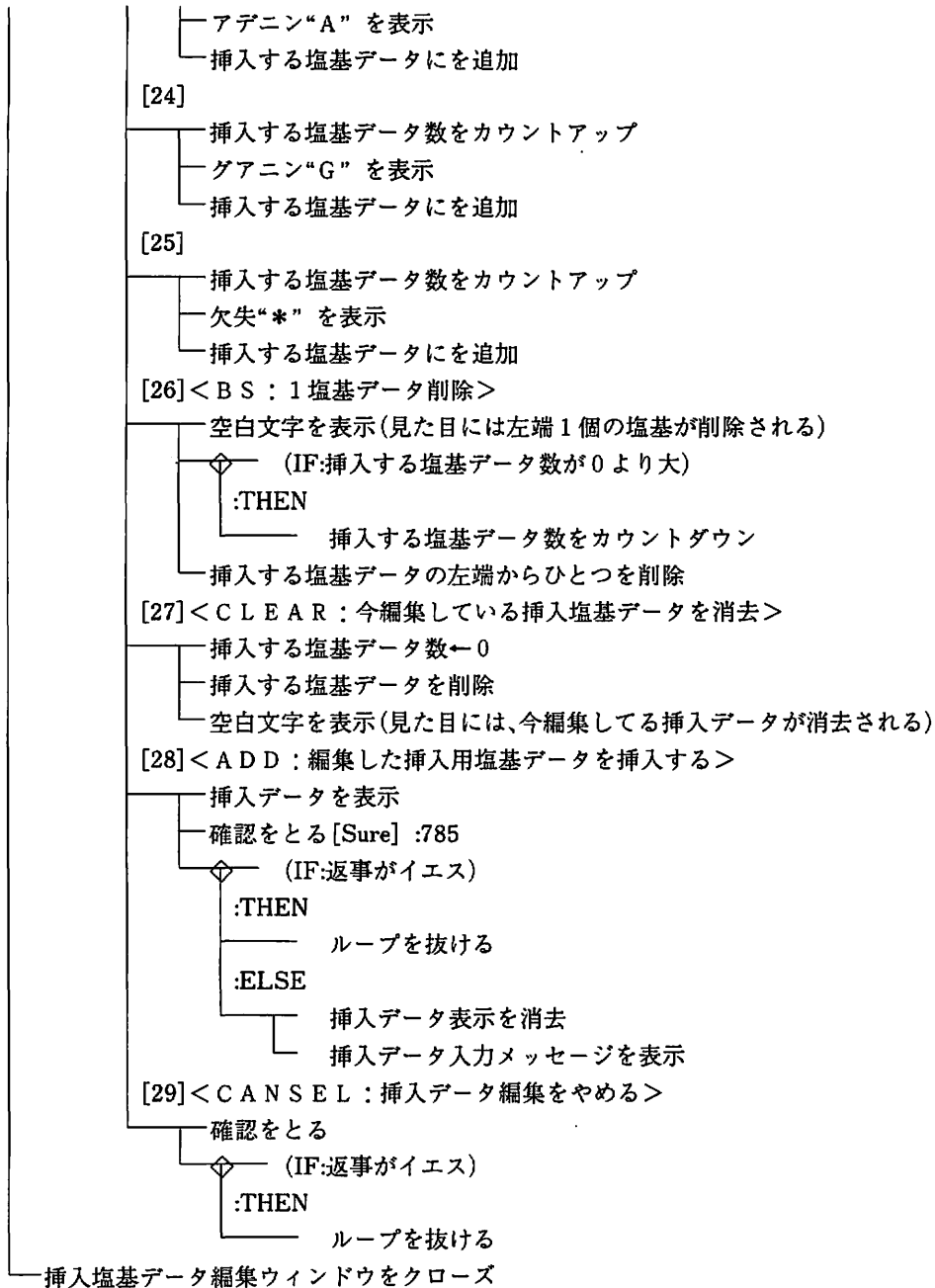
[InsRecord] :602

— 挿入処理を施す塩基データのあるセルを検索する [SerachCell] :166



[GetInsData] :624

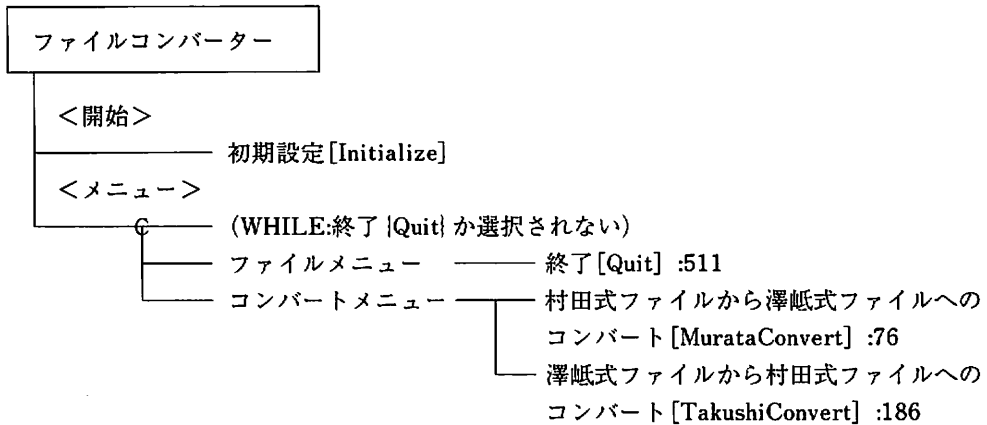




5.2 ファイルコンバーター

※ルーチン名の後ろの番号はプログラムリストの行番号をさす。

通常QuickBASICには行番号はないが、プログラムリストを追い易くするために便宜上行番号を添えた。

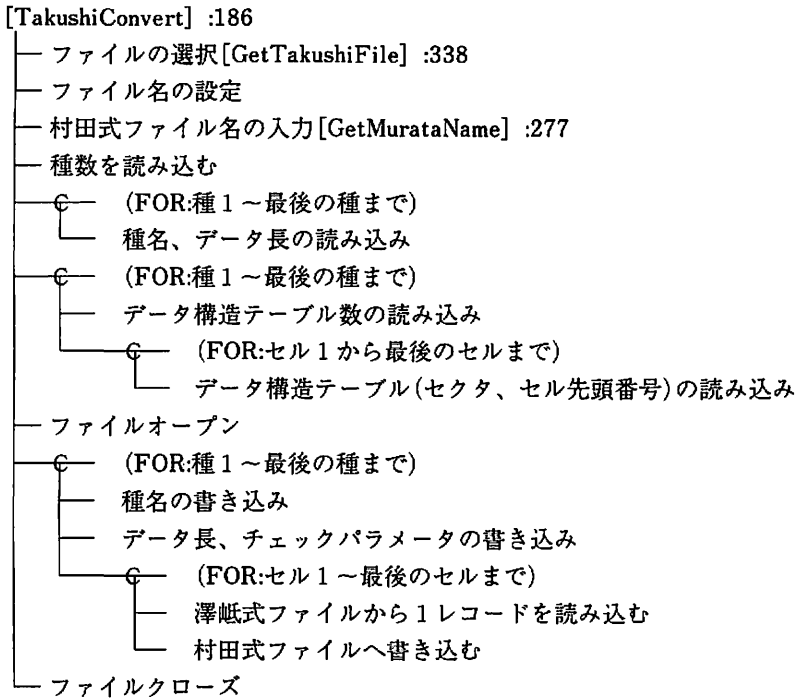


[Initialize]

- 定数の設定 :17-20
- 配列の設定 :22-24, :30-34
- ユーザー定義関数の設定 26-28
- メニューの設定 :36-44
- タイトルウィンドウの表示 :48-57

[MurataConvert] :76

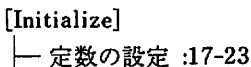
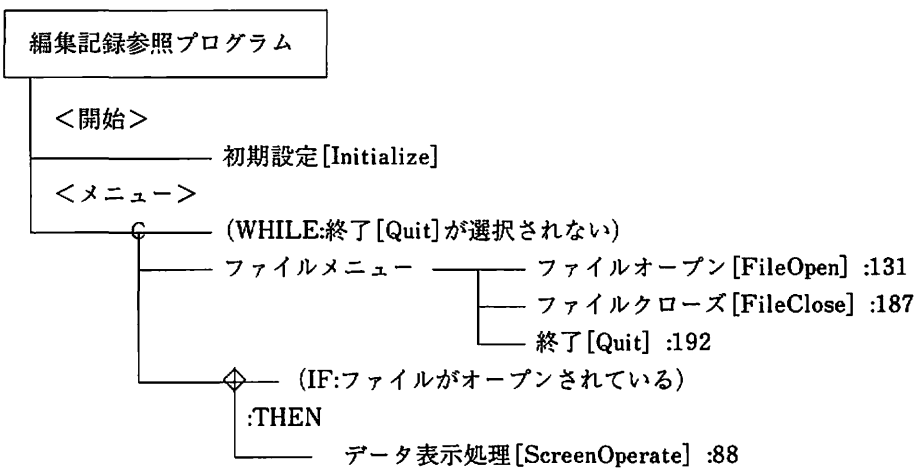
- ファイルの選択 [GetMurataFile] :288
- ファイル名の設定 [MurataSetFName] :266
- ファイルオープン
- 村田式ファイルの種数の読み込み
- (FOR:種 1 ~最後の種まで)
 - 種の名前、データ長、チェック用変数 1、チェック用変数 2 の読み込み
 - (WHILE :読み込んだデータの数とデータ長が一致しない)
 - 村田式ファイルの塩基データの読み込み
 - 澤峴式ファイルへの塩基データの書き込み
 - (IF :まだデータが残っている)
 - [THEN]
 - 残りのデータを書き込む
- データ構造を
 - 書き込む
 - (オリジナル)
 - 種の数、種名、データ長を書き込む
 - データ構造テーブル(セクタ、セル先頭番号)を書き込む
- 編集記録ファイルの作成
- データ構造を
 - 書き込む
 - (編集用 0 番)
 - 種の数、種名、データ長を書き込む
 - データ構造テーブル(セクタ、セル先頭番号)を書き込む
- データファイルのコピー(オリジナルから編集用 0 番)
- ファイルクローズ



5.3 編集記録参照プログラムのSPD

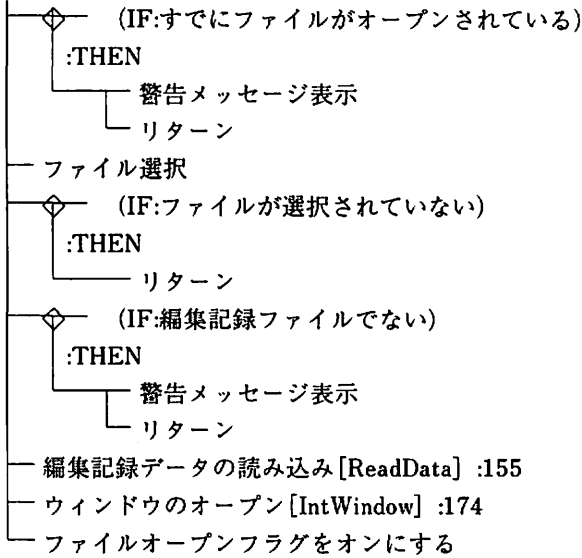
※ルーチン名の後ろの番号はプログラムリストの行番号をさす。

通常Quick BASICには行番号はないが、プログラムリストを追い易くするために便宜上行番号を添えた。



- 配列の設定 :25-27
- メニューの設定 :30-37
- タイトルウィンドウの表示 :41-49

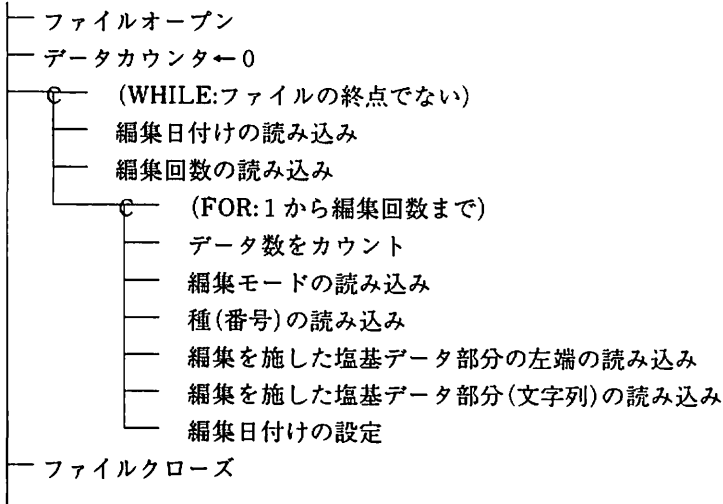
[FileOpen] :131



[FileClose] :187

- クローズメニューをオフにする
- ウィンドウをクローズ
- ファイルオープンフラグをオフにする
- データ表示フラグをオフにする

[ReadData] :155



└─ 編集回数トータル←データカウンタ

[IntWindow] :174

- └─ ウィンドウのオープン
- └─ ファイルクローズメニューをオンにする
- └─ ボタンの設定
- └─ 表示するデータ数の設定
- └─ データの表示 [DispData] :64

[DispData] :64

- └─ データの出力先をウィンドウ 1 にする
- └─ 文字のサイズ・フォント・バックグラウンドカラー・フォアグラウンドカラーを設定
- └─ ファイル名表示
- └─ 見出し表示
- └─ (FOR:指定した種から表示可能な数 {最大10} まで)
 - └─ 連番・編集モード・種番号・データ部分の左端・編集データ・日付けを出力

[ScreenOperate] :88

- └─ カーソルやマウスのアクティブウィンドウをウィンドウ 1 にする
- └─ 表示フラグをオンにする
- └─ (WHILE:表示フラグがオン)
 - └─ ボタン入力待ち
 - └─ (IF:表示フラグがオン)
 - └─ :THEN
 - └─ Pushed←押されたボタン
 - └─ (CASE:押されたボタン)
 - └─ :1
 - └─ 上方向にスクロール
 - └─ :2
 - └─ 下方向にスクロール
 - └─ :3
 - └─ 連番 1 の種を画面最上部にして表示
 - └─ :4
 - └─ 連番最後の種を画面最下部にして表示

変数の説明

変数名	種類	備考	説明			
6.1 DNA塩基配列編集プログラム				Count	整数型	1 一致
				CYAN	定数型	2 不一致
					273	forecolorやbackcolorで指定する色の値、水色
				CANSEL	定数型	3 フラグに利用する固定値、キャンセルの意味
Answer	整数型	-	確認をとった時の返事	CT	整数型	-
		YES(=2)	はい	CTs	整数型	-
		NO(=1)	いいえ			
AstarNum	整数型	-	境界線に使うアスタリスクの数	dtDNA%	配列型	DIM 50×500塩基配列データブロック
buf\$	文字型	-	クリップボード	DataFile\$	文字型	-
bTitle\$	文字型	-	タイトル	DelFile\$	文字型	-
bTitleColor	整数型	-	タイトルの色	DirPoint	整数型	-
BaseName\$	文字型	-	拡張子を除いたファイル名			
BorderTitle\$	文字型	-	境界線の中央に表示する小見出し			
BotSpe	整数型	-	画面最下端の種	DNC	配列型	DIM 7
BotNumDB	配列型	DIM 50	塩基データブロック(dtDNA%)中の塩基データ数			
BoxMes1\$	文字型	-	メッセージ1			1 T(チミン) → 緑
BoxMes2\$	文字型	-	メッセージ2			2 C(シトシン) → 水色
Box.MesColor1	整数型	-	Box.Mes1\$の色			3 A(アデニン) → 紫
Box.MesColor2	整数型	-	Box.Mes2\$の色			4 G(グアニン) → 黒
ButtonPushed	整数型	-	選択したボタン			5 * (欠損部分) → 黄色
ButtonSw	整数型	-	ボタン表示スイッチ			6 X (不一致) → 赤
		0	表示しない			7 + (一致) → 黒
		1	表示する	DT	整数型	-
BLACK	定数型	33	forecolorやbackcolorで指定する色の値、黒	eDNA\$	文字型	-
BLUE	定数型	409	forecolorやbackcolorで指定する色の値、青	EditDataFile\$	文字型	-
cflen\$	文字型	-	レコード中の塩基データ数(ファイルからファイルへのコピーの場合に使用)	EditFlag	整数型	-
check\$	文字型	-	潭紙式ファイルであるかをチェックする文字列			編集の処理待ちフラグ
eDNA\$	文字型	-	塩基データ(レコード単位・ファイルからファイルへのコピーの場合に使用)			FLAG.ON(=1) 処理待ち
Cell	整数型	-	セル番号			FLAG.OFF(=0) 処理終了
CheckPara1	整数型	-	村田式ファイルで使うチェック変数	EditStructFile\$	文字型	-
CheckPara2	整数型	-	村田式ファイルで使うチェック変数	EditMode	整数型	-
ChrCheck\$	文字型	-	村田式ファイルであるかをチェックする文字列			編集モード
Column	整数型	-	文字列を表示する列			1 カット処理
ComFile\$	文字型	-	村田式ファイル名			2 コピー処理
CopyDataFile\$	文字型	-	データファイルコピー先のファイル名			3 ベース処理
CopyNum	整数型	-	コピーするバージョン番号			4 挿入処理
CorDNA	配列型	DIM 60	塩基の一致不一致を示す	EscFlag	整数型	-
						編集処理エスケープフラグ
				ESCAPE	定数型	0
						FLAG.ON(=1) 編集処理を抜ける
						FLAG.OFF(=0) そのまま編集を続行
				ffPnt	整数型	-
				fBuf\$	整数型	-
				fCell	整数型	-
				fDNA\$	文字型	-
						編集を施した座標
						編集を施した塩基データ
						セル番号
						塩基データ(レコード単位)
						データファイルの読み書きに使用)

fEditMode	整数型	-	編集処理モード	HistoryFile \$	文字型	-	編集記録ファイル名
fHisSpe	整数型	-	編集処理を施した種	ins \$	文字型	-	挿入する塩基データ (文字列)
fLength \$	文字型	-	レコード中の塩基データ数 (データファイルの読み書きに使用)	StructFile \$	文字型	-	簿紙式ファイルのデータ構造ファイル名
fPnt	整数型	-	編集を施した塩基データ部分の左端	ItemName \$	文字型	-	表示項目
fPnt2	整数型	-	〃 の右端	ItemNum	整数型	-	項目数
FileCheck \$	文字型	-	簿紙式ファイルであるかをチェックする文字列	ItemTitle \$	文字型	-	項目の見出し
FileCount	整数型	-	簿紙式ファイルのファイル数	IN	定数型	1	フラグに利用する固定値、ループに入るの意味
FileExp \$	配列型	-	ファイル拡張子	jPnt	整数型	-	ジャンプ先のデータ番号 (編集エリアの横方向移動)
	1		編集用データ構造ファイル 0 番	JumpSpe	整数型	-	ジャンプ先の種 (編集エリアの移動)
	2		編集用データ構造ファイル 1 番	KSI	整数型	-	拡張子を区切る記号 (.) の位置 EX) FILENAME.DAT → 9
	3		編集用データ構造ファイル 2 番	LastDirPoint	整数型	-	ファイル名の中のディレクトリ (フォルダ) の仕切記号 (:) が最後にある位置 EX) 6.07:DNAGE:DATA → 11
	4		編集用データ構造ファイル 3 番	LastRecord	整数型	-	最終セクタ番号
	5		編集用データ構造ファイル 4 番	LeftSideData	整数型	-	画面左端の塩基データ番号
	6		オリジナル・データ構造ファイル	Length	整数型	-	レコード中の塩基データ数
	11		編集用データファイル 0 番	LenCell	整数型	-	現時点の塩基データ数
	12		編集用データファイル 1 番	LenDNA	配列型	DIM 50	各種のデータ数
	13		編集用データファイル 2 番	LOOP	整数型	-	ループ用フラグ
	14		編集用データファイル 3 番	Markdata	整数型	-	編集処理でマウス選択した塩基データ番号 (黒の反転表示でマーク)
	15		編集用データファイル 4 番	MasDataFile \$	文字型	-	コピーする場合のマスターになるデータファイル
	16		オリジナル・データファイル	MaxLength	整数型	-	最大のデータ数
File.Name \$	文字型	-	簿紙式ファイル名 (フォルダ名を除く)	MenuNumber	整数型	-	メニュー番号で選んだメニュー
FileID	整数型	-	ファイル番号 (0~4)	1			ファイルメニュー (File)
Folder \$	文字型	-	簿紙式ファイルが格納されているフォルダ名	2			編集メニュー (Edit)
FLAG.ON	定数型	-	フラグに利用する固定値、フラグオン	3			ジャンプメニュー (Jump)
FLAG.OFF	定数型	0	フラグに利用する固定値、フラグオフ	Menuitem	整数型	-	プルダウンメニューの中の項目
ginLOOP	整数型	-	挿入ルーチンで使うループ用フラグ	File-1			ファイルオープン
gRec	整数型	-	セクタ番号 (データファイルから読み込む場合)	File-2			ファイルクローズ
GetData \$	文字型	-	GetDataWinルーチンで得られた文字列	File-3			ファイルセーブ
GREEN	定数型	341	forecolorやbackcolorで指定する色の値、緑	File-4			ファイル名を変更してセーブ
HisCount	整数型	-	編集処理回数	File-5			ファイル削除
HisSpe	整数型	-	編集処理を受けた種 (編集記録に使用)	File-6			プログラム終了
HisTmpFile \$	文字型	-	作業用編集記録ファイル名	Edit-1			編集のアンドゥ
				Edit-2			カット編集
				Edit-3			コピー編集
				Edit-4			ペースト編集

		Edit-5	挿入編集				ブするかを確認する
		Area-1	編集エリア横方向移動 (左端にくる塩基データ番号で指定)				FLAG.OFF (=0) 編集処理は行っていない
		Area-2	編集エリア縦方向移動 (画面最上端にくる種番号で指定)	SeaSpe	整数型	-	検索する種
				SecTBL	配列型	DIM 50×100	セルのセクタ番号を示すテーブル
				SelectButton	整数型	-	選択したボタン
MouX	整数型	-	マウスのX座標	SelectFile	整数型	-	ファイル選択のフラグ
MouY	整数型	-	マウスのY座標			NO (=1)	ファイル未選択
Msg \$	文字型	-	メッセージ			YES (=0)	ファイル選択
MsgBoxType	整数型	-	MsgBoxWinルーチンのメッセージ表示タイプ	ShiftCount	整数型	-	セーブの時にバージョンをシフトするファイルの数
		1	確認のみ	Spe	整数型	-	種の番号 (FOR-NEXTループ用)
		2	返事を聞く (YES/NO)	SpeCellNum	配列型	DIM 50	各種のセル数
		3	々 (OK/CANCEL)	SpeLimit	定数型	50	エディタで一度に扱う種の数
		4	々 (OK/NO/CANCEL)	SpeName \$	配列型	DIM 50	各種の種名
MsgWindow	定数型	7	メッセージ表示用ウィンドウの番号	SpeNum	整数型	-	種の数
MAGENTA	定数型	137	forecolorやbackcolorで指定する色の値、紫	title \$	文字型	-	メッセージウィンドウのタイトル
NameFile \$	文字型	-	オープニングファイル	TitlePoint	整数型	-	挿入ルーチンでタイトルを表示するカラム座標
Num	整数型	-	ループ用カウンタ	TopNum	整数型	-	画面の最上部に表示の種の番号
NumCT	整数型	-	項目の番号	TopNumDB	配列型	DIM 50	塩基配列データブロック
NumTBL	配列型	DIM 50×100	セル先頭のデータ番号を示すテーブル	TopSpe	整数型	-	画面最上端の種 (番号)
NO	定数型	1	フラグに利用する固定値、いいえの意味	TASK \$	文字型	-	現在行っている作業
OpeSpe	整数型	-	現時点、処理の対象となっている種			NULL	なにも行っていない
pRec	整数型	-	セクタ番号 (ランダムファイルに書き込む場合)	FileOpen			編集ファイルを選択しオープンする
PntSpe	整数型	-	編集処理でマウス選択した種	Edit			編集
Program	整数型	-	プログラム実行のフラグ	UndoFlag	整数型	-	アンドゥフラグ
		IN (=1)	プログラム実行			FLAG.ON (=1)	アンドゥ呼び出された編集処理
		ESCAPE (=0)	プログラム終了			FLAG.OFF (=0)	通常の編集処理
Pushed	整数型	-	選択したボタン	wcopy \$	文字型	-	一時的にファイル名をコピーする変数
PushedButton	整数型	-	選択したボタン	wDNA \$	文字型	-	塩基データ (一時的に使う作業用変数)
RenNum	整数型	-	編集処理によって増減した塩基データ数	wLength \$	文字型	-	レコード中の塩基データ数 (一時的に使う作業用変数)
RenSpe	整数型	-	編集処理を施されてレコード中の塩基データが増減した種	Work	整数型	-	一時的に使う作業用変数
Row	整数型	-	文字列を表示する行	Work \$	文字型	-	一時的に使う作業用変数
RED	定数型	205	forecolorやbackcolorで指定する色の値、赤	WorkWindow	定数型	10	メッセージの表示など一時的に利用するウィンドウの番号
sfd	整数型	-	データの境界	WK	整数型	-	一時的に使う作業用変数
sCell	整数型	-	検索した種	WK2	整数型	-	一時的に使う作業用変数
sNum	整数型	-	検索する塩基データ番号	WHITE	定数型	30	forecolorやbackcolorで指定する色の値、白
SaveFlag	整数型	-	セーブフラグ	XP	整数型	-	マウス座標から割り出されるデータのX座標
		FLAG.ON (=1)	編集処理を1度以上行った → セーブをせずに終了しようとした場合にセー				

YELLOW	定数型	69	forecolorやbackcolorで指定する色の値、黄	eDNA \$	文字型	-	仕切記号 (:) がある位置塩基データ編集 (文字列) 作業用変数
YES	定数型	2	フラグに利用する固定値、はいの意味	ESCAPE	定数型	0	フラグに利用する固定値、ループを抜けるの意味
YP	整数型	-	マウス座標から割り出されるデータのY座標	fDNA \$	文字型	-	ランダムファイルの読み書き用変数 (塩基データ)
6.2 ファイルコンバータ				fLength \$	文字型	-	ランダムファイルの読み書き用変数 (塩基データ数)
変数名	種類	備考	説明	FileCheck \$	文字型	-	澤紙式ファイルであるかをチェックする文字列
AstarNum	整数型	-	境界線として使うアスタリスクの数	FileCount	整数型	-	澤紙式ファイルのファイル数
bTitle \$	文字型	-	タイトル	FileExp \$	配列型	-	ファイル拡張子
bTitleColor	整数型	-	タイトルの色			1	編集用データ構造ファイル0番
BaseName \$	文字型	-	拡張子を除いたファイル名			2	編集用データ構造ファイル1番
BorderTitle \$	文字型	-	境界線の中央に表示する小見出し			3	編集用データ構造ファイル2番
BoxMes1 \$	文字型	-	メッセージ			4	編集用データ構造ファイル3番
BoxMes2 \$	文字型	-	メッセージ			5	編集用データ構造ファイル4番
BoxMesColor1	整数型	-	BoxMes1 \$ の色			6	オリジナル・データ構造ファイル
BoxMesColor2	整数型	-	BoxMes2 \$ の色			11	編集用データファイル0番
ButtonPushed	整数型	-	選択したボタン			12	編集用データファイル1番
ButtonSw	整数型	-	ボタン表示スイッチ			13	編集用データファイル2番
		0	表示しない			14	編集用データファイル3番
		1	表示する			15	編集用データファイル4番
BLACK	定数型	33	forecolorやbackcolorで指定する色の値、黒			16	オリジナル・データファイル
BLUE	定数型	409	forecolorやbackcolorで指定する色の値、青	FileID	整数型	-	ファイル番号 (0~4)
check \$	文字型	-	澤紙式ファイルであるかをチェックする文字列	FileName \$	文字型	-	澤紙式ファイル名
Cell	整数型	-	セル番号	Folder \$	文字型	-	澤紙式ファイルが格納されているフォルダ名
CheckPara1	整数型	-	村田式ファイルで使うチェック変数	gRec	整数型	-	セクタ番号 (データファイルから読み込む場合)
CheckPara2	整数型	-	村田式ファイルで使うチェック変数	GREEN	定数型	341	forecolorやbackcolorで指定する色の値、緑
ChrCheck \$	文字型	-	村田式ファイルであるかをチェックする文字列	HistoryFile \$	文字型	-	編集記録ファイル名
ComFile \$	文字型	-	村田式ファイル名	IOFile \$	文字型	-	編集用データ構造ファイル0番のファイル名
ConvFile \$	文字型	-	コンバートするファイル名	ItemName \$	文字型	-	表示項目
Column	整数型	-	文字列を表示する列	ItemNum	整数型	-	項目数
Count	整数型	-	カウンタ	ItemTitle \$	文字型	-	項目の見出し
CANSEL	定数型	3	フラグに利用する固定値、キャンセルの意味	InfoFile \$	文字型	-	澤紙式ファイルのデータ構造ファイル名
CYAN	定数型	-73	forecolorやbackcolorで指定する色の値、水色	IN	定数型	1	フラグに利用する固定値、ループに入るの意味
dtDNA %	配列型	DIM 50×50	塩基配列データブロック	KSI	整数型	-	拡張子を区切る記号 (.) の位置
DOFile \$	文字型	-	編集用データファイル0番のファイル名				
DataFile \$	文字型	-	データファイル名				
DirPoint	整数型	-	ファイル名の中のディレクトリ (フォルダ) の				

			EX) FILENAME.DAT → 9	SpeNum	整数型	-	種の数
LastDirPoint	整数型	-	ファイル名の中のディレクトリ (フォルダ) の仕切記号 (:) が最後にある位置	TotalCount	整数型	-	現時点までに読み込んだ種ごとの塩基データ数
LastRecord	整数型	-	ランダムファイルの最終セクタ	wcopy \$	文字型	-	一時的にファイル名をコピーする変数
LenCell	整数型	-	現時点の塩基データ数	wDNA \$	文字型	-	塩基データの編集で一時的に使う作業用変数
LenDNA	配列型	DIM 50	各種のデータ数	wLength \$	文字型	-	ランダムファイル塩基データの1レコードサイズ (作業用)
Length	整数型	-	レコード中の塩基データ数	Work	整数型	-	一時的に使う作業用変数
MaxLength	整数型	-	最大のデータ数	Work \$	文字型	-	一時的に使う作業用変数
MenuItem	整数型	-	メニューの中の項目	WHITE	定数型	30	forecolorやbackcolorで指定する色の値、白
			File - 1 Quit	YELLOW	定数型	69	forecolorやbackcolorで指定する色の値、黄
			Convert - 1 Murata => Takushi	YES	定数型	2	フラグに利用する固定値、はいの意味
			Convert - 2 Takushi => Murata				
MenuNumber	整数型	-	メニューバーで選んだメニュー				
			1 ファイルメニュー				
			2 コンバートメニュー				
Msg \$	文字型	-	メッセージ				
MsgWindow	定数型	7	メッセージ表示用ウィンドウの番号				
MAGENTA	定数型	37	forecolorやbackcolorで指定する色の値、紫				
MODE \$	文字型	-	コンバートモード	BotSpe	整数型	-	画面の最下部に表示の種の番号
NameFile \$	文字型	-	オープニングファイル	BLACK	定数型	33	forecolorやbackcolorで指定する色の値、黒
Num	整数型	-	ループ用カウンタ	BLUE	定数型	409	forecolorやbackcolorで指定する色の値、青
NumCT	整数型	-	項目の番号	CANSEL	定数型	3	フラグに利用する固定値、キャンセルの意味
NumTBL	配列型	DIM 50×100	セル先頭のデータ番号を示すテーブル	CYAN	定数型	273	forecolorやbackcolorで指定する色の値、水色
NO	定数型	1	フラグに利用する固定値、いいえの意味	dp	整数型	-	データ番号
pRec	整数型	-	セクタ番号 (データファイルに書き込む場合)	DataLimit	整数型	-	編集記録参照プログラムで一度に扱う種の数
Program	整数型	-	プログラム実行のフラグ	DataPoint	整数型	-	編集を施した塩基データ番号
			IN(=1) プログラム実行	DispNum	整数型	-	表示しているデータの数
			ESCAPE(=0) プログラム終了	EditData \$	整数型	-	編集を施した塩基データ (文字列)
PushedButton	整数型	-	選択したボタン	EditMode	整数型	-	編集モード
RecTBL	配列型	DIM 50×100	セルのセクタを示すテーブル (セクタ番号テーブル)			1	カット処理
Row	整数型	-	文字列を表示する行			2	コピー処理
RED	定数型	205	forecolorやbackcolorで指定する色の値、赤			3	ペースト処理
SelectButton	整数型	-	選択したボタン			4	挿入処理
SelectFile	整数型	-	ファイル選択のフラグ	ESCAPE	定数型	0	フラグに利用する固定値、ループを抜けるの意味
			NO(=1) ファイル未選択				
			YES(=2) ファイル選択				
Spe	整数型	-	種の番号 (FOR-NEXTループ用)	File.Name \$	文字型	-	編集記録ファイル名
SpeCellNum	配列型	DIM 50	各種のセル数	GREEN	定数型	341	forecolorやbackcolorで指定する色の値、緑
SpeLimit	定数型	50	エディタで一度に扱う種の数	HisCount	整数型	-	編集記録回数 (全編集)
SpeName \$	配列型	DIM 50	各種の種名	HisData	整数型	-	編集処理回数 (1回の編集)

6.3 編集記録参照プログラム

変数名 種類 備考 説明

IN	定数型	1	フラグに利用する固定値、ループに入るの意味	0003 : '* TITLE : DNA Gap Editor Ver 2.1
MenuNumber	整数型	-	メニューバーで選んだメニュー	0004 : '*
		1	ファイルメニュー	0005 : '* FILE NAME : DNA Gap Editor Ver 2.1
MenuItem	整数型	-	メニューの中の項目	0006 : '*
		File-1	Open	0007 : '* REV 0.6 : Last modified on December 15th, 1992
		File-2	Close	0008 : '*
		File-3	Quit	0009 : '* AUTHOR : K.Takushi(Ryukyu univ.)
Msg \$	文字型	-	メッセージ	0010 : '*
MAGENTA	定数型	1 37	forecolorやbackcolorで指定する色の値、紫	0011 : '*****
Num	整数型	-	ループ用カウンタ	0012 :
NO	定数型	1	フラグに利用する固定値、いいえの意味	0013 : DEFINT A-Z : OPTION BASE 1
OpenFlag	整数型	-	ファイルオープンフラグ FLAG.ON(=1) ファイルオープン済み FLAG.OFF(=0)ファイルオープンしていない	0014 :
Program	整数型	-	プログラム実行のフラグ IN(=1) プログラム実行 ESCAPE(=0)プログラム終了	0015 : SpeLimit=50
Pushed	整数型	-	選択したボタン	0016 :
Rcount	整数型	-	編集記録データ数のカウンタ	0017 : DIM SpeName \$(SpeLimit),LenDNA(SpeLimit), NumTBL(SpeLimit,100)
Row	整数型	-	編集記録を表示する行座標	0018 : DIM SecTBL(SpeLimit,100),dtDNA%(SpeLimit, 500)
RED	定数型	205	forecolorやbackcolorで指定する色の値、赤	0019 : DIM SpeCellNum(SpeLimit),CorDNA(60)
ShowFlag	整数型	-	データ表示用フラグ FLAG.ON(=1) データを表示中にボタンの入力待ちをする FLAG.OFF(=0)ボタンの入力待ちをしない	0020 : DIM TopNumDB(SpeLimit),BotNumDB(Spe Limit),FileExp \$(16)
TopSpe	整数型	-	画面の最上部に表示している種の連番	0021 :
Update \$	整数型	-	アップデートした日付け	0022 : BLACK=33 : WHITE=30 : RED=205 : GREEN=341
WHITE	定数型	30	forecolorやbackcolorで指定する色の値、白	0023 : BLUE=409 : YELLOW=69 : MAGENTA=137 : CYAN=273
YELLOW	定数型	69	forecolorやbackcolorで指定する色の値、黄	0024 : NO=1 : YES=2 : CANCEL=3 : ESCAPE=0 : IN=-1 : FLAG.ON=1 : FLAG.OFF=0
YES	定数型	2	フラグに利用する固定値、はいの意味	0025 : WorkWindow=10 : MsgWindow=11 : OpenFlag= FLAG.OFF
				0026 :
				0027 : DNC(1)=GREEN : DNC(2)=CYAN : DNC(3)= MAGENTA
				0028 : DNC(4)=BLACK : DNC(5)=YELLOW : DN C(6)= BLACK : DNC(7)=RED
				0029 :
				0030 : DEF FNvData(Cchr \$,P)=INSTR("TCAG*", MID \$(Cchr \$,P,1))
				0031 : DEF FNpData(KoSpe,NoCell,dataPoint)=data Point-NumTBL(KoSpe,NoCell)
				0032 : DEF FNnData(KoSpe,NoCell)=NumTBL(KoS pe,NoCell+1)-NumTBL(KoSpe,NoCell)
				0033 :
				0034 : FileExp \$(1)="I0" : FileExp \$(2)="I1" : File Exp \$(3)="I2" : FileExp \$(4)="I3"
				0035 : FileExp \$(5)="I4" : FileExp \$(6)="I5"
				0036 : FileExp \$(11)="D0" : FileExp \$(12)="D1" : FileExp \$(13)="D2" : FileExp \$(14)="D3"
				0037 : FileExp \$(15)="D4" : FileExp \$(16)="DM"
				0038 : FileExp \$(10)="HIS"
				0039 :
				0040 : MENU RESET

プログラムリスト

通常QuickBASICには行番号はないが、プログラムリストを追いやすくするために便宜上行番号を添えた。

7.1 DNA塩基配列編集プログラム

0001 : '*****
0002 : '*


```

0041 : '*** menu bar ***
0042 : MENU 1,0,1,"File "
0043 : MENU 2,0,0,"Edit "
0044 : MENU 3,0,0,"Area "
0045 : '*** File menu ***
0046 : MENU 1,1,1,"Open..." : cmdkey 1,1,"O"
0047 : MENU 1,2,0,"Close"
0048 : MENU 1,3,0,"Save"
0049 : MENU 1,4,0,"Save As..."
0050 : MENU 1,5,1,"Delete"
0051 : MENU 1,6,1,"Quit " : cmdkey 1,6,"Q"
0052 : '*** Edit Menu ***
0053 : MENU 2,1,0,"Undo" : cmdkey 2,1,"Z"
0054 : MENU 2,2,0,"Cut" : cmdkey 2,2,"H"
0055 : MENU 2,3,0,"Copy" : cmdkey 2,3,"C"
0056 : MENU 2,4,0,"Paste" : cmdkey 2,4,"V"
0057 : MENU 2,5,0,"Insert" : cmdkey 2,5,"I"
0058 : '*** Area menu ***
0059 : MENU 3,1,0,"Horizontally"
0060 : MENU 3,2,0,"Uppermost position"
0061 :
0062 : ON MENU GOSUB MainMenu : MENU ON
0063 :
0064 : Program=IN
0065 : WHILE Program
0066 : SELECT CASE TASK $
0067 : CASE "FileOpen"
0068 : TASK $ = ""
0069 : IF FileName $ <> "" THEN GOSUB File
    Open
0070 : CASE "Edit"
0071 : GOSUB EditRoutine
0072 : CASE ELSE
0073 : END SELECT
0074 : WEND
0075 : WINDOW CLOSE 1 : WINDOW CLOSE 2 :
    WINDOW CLOSE 3
0076 : END
0077 :
0078 : MainMenu:
0079 : MenuNumber=MENU(0) : MenuItem=MENU
    (1)
0080 : MENU
0081 : ON MenuNumber GOSUB FileMenu,EditMenu,
    AreaMenu
0082 : RETURN
0083 :
0084 : FileMenu:
0085 : ON MenuItem GOSUB GetFileName,FileClose,
    FileSave,FileSaveAs,FileDelete,Quit
0086 : RETURN
0087 :
0088 : EditMenu:
0089 : ON MenuItem GOSUB UndoEdit,CutEdit,Copy
    Edit,PasteEdit,InsertEdit
0090 : RETURN
0091 :
0092 : AreaMenu:
0093 : ON MenuItem GOSUB ChangeHorizontally,
    ChangeUppermost
0094 : RETURN
0095 :
0096 : CorresDNA:
0097 : WK=LeftSideData-TopNumDB(1)
0098 : FOR I=1 TO 60
0099 : CorDNA(I)=1
0100 : FOR Spe=2 TO SpeNum
0101 : IF dtDNA%(1,WK+I) <> dtDNA%(Spe,
    WK+I) THEN
0102 : CorDNA(I)=2 : Spe=SpeNum
0103 : END IF
0104 : NEXT
0105 : NEXT
0106 : RETURN
0107 :
0108 : DispData:
0109 : TEXTSIZE 12 : TEXTFONT 3 : forecolor
    RED
0110 : LOCATE 3,11 : PRINT " "; ;
    LOCATE 3,70 : PRINT " "; ;
0111 : LOCATE 3,10 : PRINT " ";LeftSideData; ;
    LOCATE 3,69 : PRINT " ";LeftSideData+59
0112 : FOR Spe=TopSpe TO BotSpe
0113 : WK=LeftSideData-TopNumDB(Spe)
0114 : FOR DT=1 TO 60
0115 : forecolor DNC(dtDNA%(Spe,WK+DT))
0116 : LOCATE 3+Spe-TopSpe+1,9+DT :
    PRINT MID$("TCAG*",dtDNA%(Spe,
    WK+DT),1)
0117 : NEXT
0118 : NEXT
0119 : GOSUB CorresDNA : WK=BotSpe-TopSpe+1
0120 : FOR DT=1 TO 60
0121 : forecolor DNC(5+CorDNA(DT))
0122 : LOCATE 4+WK,9+DT : PRINT MID$
    ("+x",CorDNA(DT),1);
0123 : NEXT
0124 : RETURN
0125 :
0126 : DispMouLoc:
0127 : LOCATE 1,80 : PRINT MOUSE(0) : MouX=
    MOUSE(2) : MouY=MOUSE(1)
0128 : XP=(MouX-36) ¥ 16 : YP=(MouY-74) ¥ 8
0129 : IF (XP>BotSpe) OR (XP<1) THEN
0130 : LOCATE 2,40 : PRINT " spe : * * ";
0131 : ELSE

```

```

0132 : LOCATE 2,40 : PRINT " spe : ";XP+
      (TopSpe-1);
0133 : END IF
0134 : IF (YP>59) OR (YP<0) THEN
0135 :   LOCATE 2,50 : PRINT "loc : ** ";
0136 : ELSE
0137 :   LOCATE 2,50 : PRINT "loc : ";YP+(Le
      ftSideData);
0138 : END IF
0139 : RETURN
0140 :
0141 : EditRoutine:
0142 : WINDOW 1
0143 : EditFlag=FLAG.ON
0144 : WHILE EditFlag
0145 :   WHILE (DIALOG(0)<>1) AND EditFlag
0146 :     forecolor GREEN
0147 :     GOSUB DispMouLoc
0148 :   WEND
0149 :   Pushed=DIALOG(1)
0150 :   IF (Pushed<>0) THEN
0151 :     SELECT CASE Pushed
0152 :     CASE 10
0153 :       LeftSideData=LeftSideData+30 :
         GOSUB RightShift
0154 :     CASE 11
0155 :       LeftSideData=LeftSideData-30 :
         GOSUB LeftShift
0156 :     CASE 12
0157 :       LeftSideData=LeftSideData+5 :
         GOSUB RightShift
0158 :     CASE 13
0159 :       LeftSideData=LeftSideData-5 :
         GOSUB LeftShift
0160 :     CASE ELSE
0161 :       END SELECT
0162 :     END IF
0163 :   WEND
0164 : RETURN
0165 :
0166 : SearchCell:
0167 : IF sNum+1>NumTBL(SeaSpe,SpeCellNum
      (SeaSpe)) THEN
0168 :   fCell=SpeCellNum(SeaSpe) : RETURN
0169 : END IF
0170 : Cell=1
0171 : WHILE NumTBL(SeaSpe,Cell)<sNum+1 :
      Cell=Cell+1 : WEND
0172 : fCell=Cell-1
0173 : RETURN
0174 :
0175 : PutRecord:
0176 : Length=LEN(eDNA $)
0177 : LSET fDNA $ =eDNA $ +STRING $ (250-
      Length,"r")
0178 : LSET fLength $ =MKI $ (Length)
0179 : PUT #1,pRec
0180 : RETURN
0181 :
0182 : GetRecord:
0183 : GET #1,gRec
0184 : Length=CVI(fLength $)
0185 : eDNA $ =LEFT $ (fDNA $ ,Length)
0186 : RETURN
0187 :
0188 : RightShift:
0189 : FOR OpeSpe=1 TO SpeNum
0190 :   IF (LeftSideData+59)<LenDNA(OpeSpe)
      THEN
0191 :     IF (LeftSideData+59)>BotNumDB(Ope
      Spe) THEN GOSUB LoadData
0192 :   ELSE
0193 :     LeftSideData=LenDNA(OpeSpe)-59
0194 :   END IF
0195 : NEXT
0196 : GOSUB DispData
0197 : RETURN
0198 :
0199 : LeftShift:
0200 : FOR OpeSpe=1 TO SpeNum
0201 :   IF LeftSideData<1 THEN
0202 :     LeftSideData=1
0203 :   ELSE
0204 :     IF LeftSideData<TopNumDB(OpeSpe)
      THEN GOSUB LoadData
0205 :   END IF
0206 : NEXT
0207 : GOSUB DispData
0208 : RETURN
0209 :
0210 : LoadData:
0211 : sNum=LeftSideData : SeaSpe=OpeSpe :
      GOSUB SearchCell
0212 : gRec=SecTBL(OpeSpe,fCell) : GOSUB Get
      Record
0213 : FOR M=1 TO Length : dtDNA%(OpeSpe,
      M)=FNvData(eDNA $ ,M) : NEXT M
0214 : TopNumDB(OpeSpe)=NumTBL(OpeSpe,fCell)
      : sfd=Length
0215 : gRec=SecTBL(OpeSpe,fCell+1) : GOSUB
      GetRecord
0216 : FOR M=1 TO Length : dtDNA%(OpeSpe,sfd
      +M)=FNvData(eDNA $ ,M) : NEXT M
0217 : BotNumDB(OpeSpe)=TopNumDB(OpeSpe)
      +sfd+Length-1
0218 : RETURN

```

```

0219 :
0220 : FileOpen:
0221 :   Msg $ (1) = "Wait a minute." : Msg $ (2) = "" :
      GOSUB CommentWin
0222 :   '--- Set file name ---
0223 :   BaseName $ = FileName $
0224 :   StructFile $ = BaseName $ + FileExp $ (File
      Ver+1) : GOSUB GetStructData
0225 :   DataFile $ = BaseName $ + FileExp $ (FileVer
      +11)
0226 :   EditStructFile $ = BaseName $ + ".IT" : Edit
      DataFile $ = BaseName $ + ".DT"
0227 :   '--- Copy master file to temp file
0228 :   StructFile $ = EditStructFile $ : GOSUB Set
      StructData
0229 :   MasDataFile $ = DataFile $ : CopyDataFile
      $ = EditDataFile $ : GOSUB CopyDnaData
0230 :   '--- Open history record ---
0231 :   HisTmpFile $ = Folder $ + "HISTORY TEMP"
0232 :   OPEN HisTmpFile $ FOR OUTPUT AS #4
      : HisCount=0
0233 :   '--- Open data file
0234 :   OPEN EditDataFile $ AS #1 LEN=256
0235 :   FIELD #1,254 AS fDNA $,2 AS fLength $
0236 :   '--- set menu
0237 :   MENU 1,2,1 : MENU 1,4,1 : MENU 2,0,1 :
      MENU 2,1,1 : MENU 2,2,1 : MENU 2,3,1
0238 :   MENU 2,4,1 : MENU 2,5,1 : MENU 3,0,1 :
      MENU 3,1,1 : MENU 3,2,1
0239 :   IF FileVer=0 THEN MENU 1,3,1
0240 :   MENU
0241 :   '--- Set top species & bottom species ---
0242 :   TopSpe=1
0243 :   IF SpeNum<11 THEN BotSpe=SpeNum
      ELSE BotSpe=10
0244 :   '--- Open editing window & buffer window ---
0245 :   WINDOW 1,"DNA Gap Editor - Ver 2.1 -
      ",(5,45) - (635,295),6
0246 :   WINDOW 3,"Clipboard", (5,300) - (635,320),3
0247 :   forecolor RED : LOCATE 1,1 : PRINT "Clip
      board > ";
0248 :   WINDOW 1
0249 :   BUTTON 10,1,"<<",(7,20) - (37,32),1
0250 :   BUTTON 11,1,">>",(580,20) - (610,32),1
0251 :   BUTTON 12,1,"<",(40,20) - (80,32),1
0252 :   BUTTON 13,1,">",(537,20) - (577,32),1
0253 :   FOR Spe=1 TO SpeNum : TopNumDB(Spe)=
      2 : NEXT '--- TEQ
0254 :   LeftSideData=1 : GOSUB DispEditWin :
      TASK $ = "Edit" : OpenFlag=FLAG.ON
0255 :   WINDOW CLOSE WorkWindow
0256 :   RETURN
0257 :
0258 : FileClose:
0259 :   IF SaveFlag=FLAG.ON THEN
0260 :     GOSUB ConfirmSave
0261 :     IF Answer=IN THEN RETURN
0262 :   END IF
0263 :   '--- Set flag ---
0264 :   SaveFlag=FLAG.OFF : EditFlag=FLAG.
      OFF
0265 :   EditDataFile $ = "" : EditStructFile $ = "" :
      StructFile $ = "" : FileName $ = ""
0266 :   DataFile $ = "" : TASK $ = ""
0267 :   WINDOW CLOSE 1 : WINDOW CLOSE 3
0268 :   IF OpenFlag=FLAG.ON THEN
0269 :     CLOSE : KILL HisTmpFile $ : OpenFlag=
      FLAG.OFF
0270 :   END IF
0271 :   MENU 1,2,0 : MENU 1,3,0 : MENU 1,4,0 :
      MENU 3,1,0 : MENU 3,2,0
0272 :   MENU 2,1,0 : MENU 2,2,0 : MENU 2,3,0 :
      MENU 2,4,0 : MENU 2,5,0
0273 :   MENU
0274 :   RETURN
0275 :
0276 : FileSave:
0277 :   IF EditFlag=FLAG.OFF THEN
0278 :     Msg $ (1) = "No file is open." : MsgBoxType=
      1 : GOSUB MsgBoxWin
0279 :   END IF
0280 :   Msg $ (1) = "Wait a minute." : GOSUB Comm
      entWin
0281 :   '--- Seve structure file ---
0282 :   StructFile $ = EditStructFile $ : GOSUB Set
      StructData
0283 :   IF FileCount=5 THEN
0284 :     KILL BaseName $ + FileExp $ (5) : KILL
      BaseName $ + FileExp $ (15)
0285 :   END IF
0286 :   IF FileCount<5 THEN
0287 :     ShiftCount=FileCount
0288 :   ELSE
0289 :     ShiftCount=4
0290 :   END IF
0291 :   FOR CopyNum=ShiftCount TO 1 STEP -1
0292 :     NAME BaseName $ + FileExp $ (CopyNum)
      AS BaseName $ + FileExp $ (CopyNum+1)
0293 :     NAME BaseName $ + FileExp $ (CopyNum
      +10) AS BaseName $ + FileExp $ (CopyNu
      m+11)
0294 :   NEXT
0295 :   IF FileCount<5 THEN FileCount = File
      Count + 1
0296 :   '--- Save edited data ---
0297 :   StructFile $ = BaseName $ + ".IT" : GOSUB

```

```

GetStructData
0298 : StructFile $ =BaseName $ + ".IO" : GOSUB
SetStructData
0299 : MasDataFile $ =BaseName $ + ".DT" : Copy
DataFile $ =BaseName $ + ".D0"
0300 : GOSUB CopyDnaData : GOSUB SetHisData
: GOSUB SetFileStruct
0301 : '--- Open data file ---
0302 : OPEN EditDataFile $ AS #1 LEN=256
0303 : FIELD #1,254 AS fDNA $,2 AS fLength $
0304 : SaveFlag=FLAG.OFF : WINDOW CLOSE
WorkWindow
0305 : RETURN
0306 :
0307 : FileSaveAs:
0308 : Msg $ (1) = "Input a new file name.": GOSUB
GetDataWin
0309 : IF Answer=NO THEN RETURN
0310 : FileName $ =GetData $ : Msg $ (1) = "Wait
a minute.": GOSUB CommentWin
0311 : '--- Make an original file ---
0312 : StructFile $ =Folder $ +FileName $ + ".IM" :
GOSUB SetStructData
0313 : MasDataFile $ =EditDataFile $ : CopyData
File $ =Folder $ +FileName $ + ".DM"
0314 : GOSUB CopyDnaData
0315 : '--- make a data file
0316 : StructFile $ =Folder $ +FileName $ + ".IO" :
GOSUB SetStructData
0317 : CopyDataFile $ =Folder $ +FileName $ + ".D0"
0318 : '--- make a temp file
0319 : StructFile $ =Folder $ +FileName $ + ".IT" :
GOSUB SetStructData
0320 : NAME BaseName $ + ".DT" AS Folder $ +
FileName $ + ".DT"
0321 : BaseName $ =Folder $ +FileName $
0322 : EditDataFile $ =BaseName $ + ".DT" : Edit
StructFile $ =BaseName $ + ".IT"
0323 : OPEN BaseName $ + ".HIS" FOR OUTPUT
AS #3 : CLOSE #3 : GOSUB SetHisData
0324 : GOSUB SetFileStruct
0325 : '--- Open data file ---
0326 : FileVer=0 : MENU 1,3,1 : MENU
0327 : OPEN EditDataFile $ AS #1 LEN=256
0328 : FIELD #1,254 AS fDNA $,2 AS fLength $
0329 : SaveFlag=FLAG.OFF : WINDOW CLOSE
WorkWindow
0330 : RETURN
0331 :
0332 :
0333 :
0334 : Quit:
0335 : IF SaveFlag=FLAG.ON THEN GOSUB Co-

nfirmSave ELSE TASK $ = ""
0336 : IF OpenFlag=FLAG.ON THEN
0337 : CLOSE : KILL HisTmpFile $
0338 : KILL BaseName $ + ".DT" : KILL Base
Name $ + ".IT"
0339 : END IF
0340 : Program=ESCAPE : EditFlag=FLAG.OFF
: OpenFlag=FLAG.OFF
0341 : RETURN
0342 :
0343 : UndoEdit:
0344 : IF buf $ = "" THEN RETURN
0345 : UndoFlag=FLAG.ON : SaveFlag=FLAG.ON
0346 : PntSpe=HisSpe : fPnt=fPnt : fPnt2=fPnt
+ (LEN(buf $) - 1)
0347 : SELECT CASE EditMode
0348 : CASE 1 : 'CUT --> Paste
0349 : GOSUB PasteEdit
0350 : CASE 2 : 'COPY --> Do nothing
0351 : CASE 3 : 'PASTE --> Cut
0352 : GOSUB CutEdit
0353 : CASE 4 : 'INSERT --> Cut
0354 : GOSUB CutEdit
0355 : END SELECT
0356 : UndoFlag=FLAG.OFF
0357 : RETURN
0358 :
0359 : CutEdit:
0360 : IF UndoFlag=FLAG.OFF THEN
0361 : GOSUB GetPoint2
0362 : IF EscFlag=FLAG.ON THEN RETURN
0363 : END IF
0364 : SeaSpe=PntSpe : sNum=fPnt : GOSUB
SearchCell
0365 : WK=fPnt2-fPnt : buf $ = ""
0366 : FOR P=0 TO WK
0367 : WK2=fPnt+P-TopNumDB(PntSpe)+1
0368 : buf $ =buf $ +MID $ ("TCAG*",dtDNA%
(PntSpe,WK2),1)
0369 : NEXT P
0370 : IF NumTBL(PntSpe,fCell+1) >= fPnt2
THEN
0371 : gRec=SecTBL(PntSpe,fCell) : GOSUB
GetRecord
0372 : WK=FNpData(PntSpe,fCell,fPnt) : WK2=
Length-FNpData(PntSpe,fCell,fPnt2)-1
0373 : wDNA $ =eDNA $ : eDNA $ =LEFT $
(wDNA $,WK)+RIGHT $(wDNA $,WK2)
0374 : pRec=gRec : GOSUB PutRecord : pass=1
0375 : ELSE
0376 : gRec=SecTBL(PntSpe,fCell) : GOSUB Get
Record
0377 : WK=FNpData(PntSpe,fCell,fPnt) : wDNA $ =

```

```

eDNA $ : eDNA $ =LEFT $ (wDNA $ ,WK)
0378 : pRec=gRec : GOSUB PutRecord
0379 : gRec=SecTBL(PntSpe,fCell+1) : GOSUB
GetRecord
0380 : WK=Length-FNpData (PntSpe,fCell+1,fPnt
2)-1: wDNA $ =eDNA $
0381 : eDNA $ =RIGHT $ (wDNA $ ,WK)
0382 : pRec=gRec : GOSUB PutRecord : pass=2
0383 : END IF
0384 : RenNum=-LEN(buf $ ) : RenSpe=PntSpe :
GOSUB RenNumTBL : GOSUB DispSpeName
0385 : IF FNpData (PntSpe,fCell) < 100 THEN
0386 : sCell=fCell : GOSUB Supply
0387 : ELSE
0388 : IF FNpData (PntSpe,fCell+1) < 100 THEN
0389 : sCell=fCell+1 : GOSUB Supply
0390 : END IF
0391 : END IF
0392 : OpeSpe=PntSpe : GOSUB LoadData :
GOSUB DispBuf : GOSUB DispData
0393 : HisSpe=PntSpe : EditMode=1 : GOSUB
RechisData : SaveFlag=FLAG.ON
0394 : RETURN
0395 :
0396 : Supply:
0397 : Work=FNpData (PntSpe,sCell) + FNpData
(PntSpe,sCell+1)
0398 : IF Work < 250 THEN
0399 : ' combination
0400 : gRec=SecTBL(PntSpe,sCell) : GOSUB
GetRecord : wDNA $ =eDNA $
0401 : gRec=SecTBL(PntSpe,sCell+1) : GOSUB
GetRecord : wDNA $ =wDNA $ +eDNA $
0402 : eDNA $ =wDNA $
0403 : pRec=SecTBL(PntSpe,sCell) : GOSUB
PutRecord
0404 : FOR I=sCell+1 TO SpeCellNum(PntSpe)
-1
0405 : NumTBL(PntSpe,I)=NumTBL(PntSpe)-1
0406 : NEXT
0407 : SpeCellNum(PntSpe)=SpeCellNum(PntSpe)
-1
0408 : ELSE
0409 : 'cut up
0410 : gRec=SecTBL(PntSpe,sCell) : GOSUB
GetRecord : wDNA $ =eDNA $
0411 : gRec=SecTBL(PntSpe,sCell+1) : GOSUB
GetRecord : wDNA $ =wDNA $ +eDNA $
0412 : eDNA $ =LEFT $ (wDNA $ ,125) : pRec=
SecTBL(PntSpe,sCell)
0413 : GOSUB PutRecord
0414 : eDNA $ =RIGHT $ (wDNA $ ,LEN(wDNA
$ )-125) : pRec=SecTBL(PntSpe,sCell+1)
0415 : GOSUB PutRecord
0416 : NumTBL(PntSpe,sCell+1) = NumTBL(Pnt
Spe,sCell) + 125
0417 : END IF
0418 : RETURN
0419 :
0420 : CopyEdit:
0421 : GOSUB GetPoint2
0422 : IF EscFlag=FLAG.ON THEN RETURN
0423 : WK=fPnt2-fPnt : buf $ =""
0424 : FOR P=0 TO WK
0425 : WK2=fPnt+P-TopNumDB(PntSpe)+1
0426 : buf $ =buf $ +MID $ ("TCAG*",dtDNA %
(PntSpe,WK2),1)
0427 : NEXT P
0428 : GOSUB DispBuf : HisSpe=PntSpe : Edit
Mode=2 : GOSUB RechisData : SaveFlag=
FLAG.ON
0429 : RETURN
0430 :
0431 : PasteEdit:
0432 : IF UndoFlag=FLAG.OFF THEN
0433 : GOSUB GetPoint1
0434 : IF EscFlag=FLAG.ON THEN RETURN
0435 : END IF
0436 : ins $ =buf $ : GOSUB InsRecord
0437 : OpeSpe=PntSpe : GOSUB LoadData :
GOSUB DispBuf : GOSUB DispData
0438 : HisSpe=PntSpe : EditMode=3 : GOSUB
RechisData : SaveFlag=FLAG.ON
0439 : RETURN
0440 :
0441 : InsertEdit:
0442 : GOSUB GetPoint1
0443 : IF EscFlag=FLAG.ON THEN RETURN
0444 : GOSUB onMark : GOSUB GetInsData :
GOSUB InsRecord
0445 : OpeSpe=PntSpe : GOSUB LoadData
0446 : buf $ =ins $ : GOSUB DispBuf : GOSUB
DispData
0447 : HisSpe=PntSpe : EditMode=4 : GOSUB
RechisData : SaveFlag=FLAG.ON
0448 : RETURN
0449 :
0450 : ChangeHorizontally:
0451 : title $ ="Change editing area horizontally."
0452 : Msg $ (1)="Input the left end of an area " :
GOSUB GetDataWin
0453 : jPnt=VAL(GetData $ )
0454 : IF (jPnt > 0) AND (jPnt < MaxLength+1)
THEN
0455 : LeftSideData=jPnt : GOSUB DispEditWin
0456 : END IF

```

```

0457 : RETURN
0458 :
0459 : ChangeUppermost:
0460 : IF SpeNum<11 THEN RETURN
0461 : title $="Change editing area vertically."
0462 : Msg $ (1)="Input the number of species for
the uppermost position " : GOSUB GetData
Win
0463 : JmpSpe=VAL(GetData $)
0464 : IF (JmpSpe>0) AND (JmpSpe<SpeNum) T
HEN
0465 : IF JmpSpe+9>SpeNum THEN TopSpe=
SpeNum-9 ELSE TopSpe=JmpSpe
0466 : BotSpe=TopSpe+1
0467 : GOSUB DispEditWin
0468 : END IF
0469 : RETURN
0470 :
0471 : RecHisData:
0472 : PRINT #4,EditMode
0473 : PRINT #4,HisSpe ' edited species number
0474 : PRINT #4,fPnt ' DNA's point
0475 : PRINT #4,buf $ ' edited data
0476 : HisCount = HisCount + 1
0477 : RETURN
0478 :
0479 : GetPoint1:
0480 : Msg $ (1)="Click insertion point." : GOSUB
CommentWin : WINDOW 1
0481 : LOOP=IN : EscFlag=FLAG.OFF : GOSUB
onESC
0482 : WHILE LOOP
0483 : WHILE MOUSE(0)<>1 : GOSUB Disp
MouLoc : WEND
0484 : GOSUB CheckESC
0485 : IF (XP>0) AND (YP>0) AND (YP<60)
THEN
0486 : PntSpe=XP+ (TopSpe-1) : fPnt=YP+
LeftSideData
0487 : MarkData=dtDNA% (PntSpe,fPnt-Top
Num DB(PntSpe)+1)
0488 : GOSUB onMark
0489 : GOSUB Sure
0490 : IF Answer=YES THEN
0491 : LOOP=ESCAPE
0492 : ELSE
0493 : GOSUB offMark
0494 : END IF
0495 : END IF
0496 : WEND
0497 : GOSUB offMark : GOSUB offESC : WINDOW
CLOSE WorkWindow
0498 : RETURN
0499 :
0500 : GetPoint2:
0501 : Msg $ (1)="Click the left end" : GOSUB
CommentWin : WINDOW 1
0502 : LOOP=IN : CTs=0 : MarkData=0 : EscFlag=
FLAG.OFF : GOSUB onESC
0503 : WHILE LOOP
0504 : WHILE MOUSE(0)<>1: GOSUB DispMou
Loc : WEND
0505 : GOSUB CheckESC
0506 : IF (XP>0) AND (YP>0) AND (YP<60)
THEN
0507 : IF CTs=0 THEN
0508 : PntSpe=XP+ (TopSpe-1) : fPnt=YP
+ LeftSideData
0509 : MarkData=dtDNA% (PntSpe,fPnt-Top
NumDB(PntSpe)+1)
0510 : GOSUB onMark : CTs=1
0511 : Msg $ (1)="Click the right end" :
GOSUB CommentWin : WINDOW 1
0512 : ELSE
0513 : IF PntSpe=XP+ (TopSpe-1) AND
fPnt<= YP+LeftSideData THEN
0514 : fPnt=YP+LeftSideData
0515 : MarkData=dtDNA% (PntSpe,fPnt-
TopNumDB(PntSpe)+1)
0516 : GOSUB onMark : GOSUB Sure
0517 : IF Answer=YES THEN
0518 : LOOP=ESCAPE : GOSUB off
Mark
0519 : MarkData=dtDNA% (PntSpe,fPnt-
TopNumDB(PntSpe)+1)
0520 : XP=PntSpe : YP=fPnt-LeftSide
Data : GOSUB offMark
0521 : ELSE
0522 : GOSUB CommentWin : WINDOW
1
0523 : GOSUB offMark
0524 : END IF
0525 : END IF
0526 : END IF
0527 : END IF
0528 : WEND
0529 : GOSUB offESC : WINDOW CLOSE Work
Window
0530 : RETURN
0531 :
0532 : SetStructData:
0533 : OPEN StructFile $ FOR OUTPUT AS #2
0534 : PRINT #2,SpeNum
0535 : FOR Spe=1 TO SpeNum
0536 : PRINT #2,SpeName $ (Spe)
0537 : PRINT #2,LenDNA (Spe)

```

```

0538 : NEXT
0539 :   FOR Spe=1 TO SpeNum
0540 :     PRINT #2,SpeCellNum(Spe)
0541 :     FOR Cell=1 TO SpeCellNum(Spe)
0542 :       PRINT #2,NumTBL(Spe,Cell),SecTBL
         (Spe,Cell)
0543 :     NEXT
0544 :   NEXT
0545 :   PRINT #2,LastRecord
0546 :   CLOSE #2
0547 : RETURN
0548 :
0549 : GetStructData:
0550 : OPEN StructFile $ FOR INPUT AS #2
0551 :   MaxLength=0 : INPUT #2,SpeNum
0552 :   FOR Spe=1 TO SpeNum
0553 :     INPUT #2,SpeName $ (Spe),LenDNA(Spe)
0554 :     IF LenDNA(Spe)>MaxLength THEN Max
         Length=LenDNA(Spe)
0555 :   NEXT
0556 :   FOR Spe=1 TO SpeNum
0557 :     INPUT #2,SpeCellNum(Spe)
0558 :     FOR Cell=1 TO SpeCellNum(Spe)
0559 :       INPUT #2,NumTBL(Spe,Cell),SecTBL
         (Spe,Cell)
0560 :     NEXT
0561 :   NEXT
0562 :   INPUT #2,LastRecord
0563 :   CLOSE #2
0564 : RETURN
0565 :
0566 : CopyDnaData:
0567 :   CLOSE #1 : CLOSE #2
0568 :   OPEN CopyDataFile $ FOR OUTPUT AS
         #2 : CLOSE #2
0569 :   OPEN "R",#2,CopyDataFile $ ,256
0570 :   FIELD #2,254 AS cDNA $ ,2 AS cflen $
0571 :   CLOSE #2
0572 :   OPEN CopyDataFile $ AS #2 LEN=256
0573 :   FIELD #2,254 AS cDNA $ ,2 AS cflen $
0574 :   OPEN MasDataFile $ AS #1 LEN=256
0575 :   FIELD #1,254 AS fDNA $ ,2 AS flength $
0576 :   FOR I=1 TO LastRecord
0577 :     gRec=I : GOSUB GetRecord
0578 :     LSET cDNA $ =eDNA $ +STRING $(254
         -Length,"r")
0579 :     LSET cflen $ =MKI $(Length)
0580 :     PUT #2,I
0581 :   NEXT I
0582 :   CLOSE #1 : CLOSE #2
0583 : RETURN
0584 :
0585 : DispEditWin:
0586 : WINDOW 1 : TEXTFONT 3 : TEXTSIZE
         12 : forecolor RED
0587 : LOCATE 1,1 : PRINT BaseName $ ;
0588 : forecolor BLUE : PRINT " Ver = ";File
         Ver;
0589 : GOSUB DispSpeName : GOSUB LeftShift
0590 : RETURN
0591 :
0592 : DispSpeName:
0593 : WINDOW 1
0594 : TEXTFONT 3 : TEXTSIZE 12
0595 : forecolor BLACK
0596 : FOR Spe=TopSpe TO BotSpe
0597 :   LOCATE 3+Spe,1 : PRINT Spe;MID $
         (SpeName $ (Spe),4,5);
0598 :   LOCATE 3+Spe,71 : PRINT LenDNA
         (Spe);
0599 : NEXT
0600 : RETURN
0601 :
0602 : InsRecord:
0603 :   sNum=fPnt : SeaSpe=PntSpe : GOSUB
         SearchCell
0604 :   gRec=SecTBL(PntSpe,fCell) : GOSUB Get
         Record
0605 :   WK=FNpData(PntSpe,fCell,fPnt)
0606 :   wDNA $ =LEFT $(eDNA $ ,WK)+ins $ +MID
         $(eDNA $ ,WK+1,Length-WK)
0607 :   IF LEN(wDNA $ )>254 THEN
0608 :     eDNA $ =LEFT $(wDNA $ ,125) : pRec=
         SecTBL(PntSpe,fCell) : GOSUB PutRecord
0609 :     eDNA $ =RIGHT $(wDNA $ ,LEN(wDNA
         $ )-125) : pRec=LastRecord+1 : GOSUB
         PutRecord
0610 :     FOR K=mssp(PntSpe) TO fCell+1 STEP
         -1
0611 :       NumTBL(PntSpe,K+1)=NumTBL(Pnt
         Spe,K) : SecTBL(PntSpe,K+1)=SecTBL
         (PntSpe,K)
0612 :     NEXT K
0613 :     NumTBL(PntSpe,fCell+1)=NumTBL(Pnt
         Spe,fCell)+125 : SecTBL(PntSpe,fCell+1)=
         LastRecord+1
0614 :     LastRecord=LastRecord+1
0615 :     RenNum=LEN(ins $ ) : RenSpe=PntSpe :
         GOSUB RenNumTBL
0616 :   ELSE
0617 :     eDNA $ =wDNA $ : pRec=SecTBL(Pnt
         Spe,fCell) : GOSUB PutRecord
0618 :     RenNum=LEN(ins $ ) : RenSpe=PntSpe :
         GOSUB RenNumTBL
0619 :   END IF
0620 : GOSUB DispEditWin

```

```

0621 : RETURN
0622 :
0623 :
0624 : GetInsData:
0625 : WINDOW 4,"Insert Data Setting Window",
      (185,345) - (625,475),6
0626 : BUTTON 21,1,"T", (10,15) - (32,37),1 : BUTT
      ON 22,1,"C", (40,15) - (62,37),1
0627 : BUTTON 23,1,"A", (70,15) - (92,37),1 : BUTT
      ON 24,1,"G", (100,15) - (122,37),1
0628 : BUTTON 25,1,"*", (130,15) - (152,37),1 :
      BUTTON 26,1,"BS", (170,15) - (205,37),1
0629 : BUTTON 27,1,"CLEAR", (215,15) - (280,37),1
      :
      BUTTON 28,1,"ADD", (290,15) - (340,37),1
0630 : BUTTON 29,1,"CANCEL", (350,15) - (425,37),1
0631 : TEXTFONT 3 : LINE (48,50) - (270,80) ,,B :
      CT=0 : ins $ = ""
0632 : forecolor BLUE : CALL MOVETO (300,
      70) : PRINT "Count : "
0633 : forecolor MAGENTA : CALL MOVETO (10,
      110)
0634 : PRINT "Create insertion data by clicking the
      upper button."
0635 : ginLOOP=IN
0636 : WHILE ginLOOP
0637 : forecolor BLACK
0638 : CALL MOVETO ( 10,70) : PRINT " "
      : CALL MOVETO (340,70) : PRINT CT;
0639 : WHILE DIALOG (0) <> 1 : WEND
0640 : Pushed=DIALOG (1) : IF CT>=20 THEN
      Pushed=0
0641 : SELECT CASE Pushed
0642 : CASE 21 : forecolor GREEN : CT=CT
      +1 : CALL MOVETO (50+CT*10,70)
0643 : PRINT "T"; : ins $ =ins $ + "T"
0644 : CASE 22 : forecolor CYAN : CT=CT
      +1 : CALL MOVETO (50+CT*10,70)
      PRINT "C"; : ins $ =ins $ + "C"
0645 : CASE 23 : forecolor MAGENTA : CT=
      CT+1 : CALL MOVETO (50+CT*10,70)
      PRINT "A"; : ins $ =ins $ + "A"
0646 : CASE 24 : forecolor BLACK : CT=CT
      +1 : CALL MOVETO (50+CT*10,70)
      PRINT "G"; : ins $ =ins $ + "G"
0647 : CASE 25 : forecolor YELLOW : CT=CT
      +1 : CALL MOVETO (50+CT*10,70)
      PRINT "*"; : ins $ =ins $ + "*"
0648 : CASE 26 : CALL MOVETO (50+CT*
      10,70) : PRINT " ";
0649 : IF CT>0 THEN CT=CT-1
0650 : ins $ =LEFT $ (ins $ ,CT)
0651 : CASE 27 : CT=0 : ins $ = "" : CALL
      MOVETO (53,70) : PRINT STRING $
      (51," ")
0652 : CASE 28 : forecolor MAGENTA : CALL
      MOVETO (50,110)
0653 : PRINT STRING $ (51," ") :
      CALL MOVETO (50,110)
0654 : PRINT "Insertion data : ";ins $
0655 : GOSUB Sure
0656 : IF Answer=YES THEN
      ginLOOP=ESCAPE
0657 : ELSE
      CALL MOVETO (50,110) : PRINT
      STRING $ (51," ")
0658 : CALL MOVETO (10,110)
0659 : PRINT "Create insertion data by
      clicking the upper button."
0660 : END IF
0661 : CASE 29 : GOSUB Sure : IF Answer=
      YES THEN ginLOOP=ESCAPE
0662 : END SELECT
0663 : WEND
0664 : WINDOW CLOSE 4 : WINDOW 1
0665 : RETURN
0666 : RenNumTBL:
0667 : FOR K=fCell+1 TO SpeCellNum(RenSpe)
0668 : NumTBL(RenSpe,K) =NumTBL(RenSpe,K)
      +RenNum
0669 : NEXT K
0670 : LenDNA (RenSpe) =LenDNA (RenSpe) +Ren
      Num
0671 : RETURN
0672 :
0673 : onESC:
0674 : forecolor BLUE : LOCATE 2,20 : PRINT "
      ESC" : forecolor GREEN
0675 : RETURN
0676 :
0677 : offESC:
0678 : LOCATE 2,20 : PRINT " "
0679 : RETURN
0680 : onMark:
0681 : forecolor WHITE : backcolor BLACK
0682 : LOCATE 3+XP,10+YP : PRINT MID $ ("
      TCAG*",MarkData,1)
0683 : forecolor GREEN : backcolor WHITE
0684 : RETURN
0685 :
0686 : offMark:
0687 : IF MarkData=0 THEN RETURN
0688 : forecolor DNC(MarkData)
0689 : LOCATE 3+XP,10+YP : PRINT MID $ ("
      TCAG*",MarkData,1)
0690 : forecolor GREEN

```



```

0698 : RETURN
0699 :
0700 : DispBuf:
0701 : WINDOW 3 : TEXTFONT 3 : TEXTSIZE
      12
0702 : LOCATE 1,8 : PRINT STRING$(100," ");
0703 : FOR K=1 TO LEN(buf$)
0704 :   forecolor DNC(FNVData(buf$,K))
0705 :   LOCATE 1,7+K : PRINT MID$(buf$,K,
      1);
0706 : NEXT K
0707 : WINDOW 1
0708 : RETURN
0709 :
0710 : MsgBoxWin:
0711 : WINDOW WorkWindow,"", (150,100) - (550,230),
      2
0712 : SELECT CASE MsgBoxType
0713 : CASE 1
0714 :   BUTTON 31,1,"OK", (150,110) - (250,125),1
0715 : CASE 2
0716 :   BUTTON 31,1,"YES", (110,110) - (200,125),1
0717 :   BUTTON 32,1,"NO", (210,110) - (300,125),1
0718 : CASE 3
0719 :   BUTTON 31,1,"OK", (110,110) - (200,125),1
0720 :   BUTTON 32,1,"CANSEL", (210,110) - (300,
      125),1
0721 : CASE 4
0722 :   BUTTON 31,1,"YES", (260,110) - (350,125),1
0723 :   BUTTON 32,1,"NO", (160,110) - (250,125),1
0724 :   BUTTON 33,1,"CANSEL", ( 60,110) - (150,
      125),1
0725 : END SELECT
0726 : forecolor RED : backcolor WHITE : TEXTF-
      ONT 2 : TEXTSIZE 12
0727 : TitlePoint=29-LEN(title$)/2
0728 : LOCATE 2,TitlePoint : PRINT title$
0729 : forecolor BLUE
0730 : LOCATE 4,3 : PRINT Msg$(1) : LOCATE
      5,3 : PRINT Msg$(2)
0731 : LOCATE 6,3 : PRINT Msg$(3) : LOCATE
      7,3 : PRINT Msg$(4)
0732 : LOCATE 8,3 : PRINT Msg$(5)
0733 : WHILE DIALOG(0) <> 1 : WEND
0734 : PushedButton=DIALOG(1)
0735 : SELECT CASE PushedButton
0736 : CASE 31 : Answer=FLAG.ON
0737 : CASE 32 : Answer=FLAG.OFF
0738 : CASE 33 : Answer=IN
0739 : END SELECT
0740 : FOR Count=1 TO 5 : Msg$(Count)="" :
      NEXT
0741 : WINDOW CLOSE WorkWindow : WINDOW
      1
0742 : MsgOption=NO
0743 : RETURN
0744 :
0745 : GetDataWin:
0746 : WINDOW WorkWindow,"", (150,100) - (550,
      230),2
0747 : BUTTON 31,1,"OK", (110,110) - (200,125),1
0748 : BUTTON 32,1,"CANSEL", (210,110) - (300,
      125),1
0749 : forecolor RED : backcolor WHITE :
      TEXTFONT 2 : TEXTSIZE 12
0750 : TitlePoint=29-LEN(title$)/2
0751 : LOCATE 2,TitlePoint : PRINT title$
0752 : forecolor BLUE
0753 : LOCATE 4,3 : PRINT Msg$(1)
0754 : LOCATE 5,3 : PRINT Msg$(2)
0755 : LOCATE 6,3 : PRINT Msg$(3)
0756 : LOCATE 8,10 : INPUT GetData$
0757 : WHILE DIALOG(0) <> 1 : WEND
0758 : PushedButton=DIALOG(1)
0759 : SELECT CASE PushedButton
0760 : CASE 31 : Answer=FLAG.ON
0761 : CASE 32 : Answer=FLAG.OFF
0762 : END SELECT
0763 : WINDOW CLOSE WorkWindow : WINDOW
      1
0764 : FOR Count=1 TO 3 : Msg$(Count)="" :
      NEXT
0765 : MsgOption=NO
0766 : RETURN
0767 :
0768 : CommentWin:
0769 : WINDOW WorkWindow,"", (110,403) - (580,
      440),2
0770 : TEXTFONT 2 : TEXTSIZE 18 : forecolor
      BLACK : backcolor CYAN : CLS
0771 : LOCATE 1,1 : PRINT Msg$(1);
0772 : LOCATE 2,1 : PRINT Msg$(2);
0773 : FOR Count=1 TO 2 : Msg$(Count)="" :
      NEXT
0774 : RETURN
0775 :
0776 : SetFileStruct:
0777 : OPEN FileName$ FOR OUTPUT AS #5
0778 : PRINT #5,"DNA GAP EDITOR"
0779 : PRINT #5,FileName$
0780 : PRINT #5,Folder$
0781 : PRINT #5,FileCount
0782 : CLOSE #5
0783 : RETURN
0784 :
0785 : Sure:

```

```

0786 : WINDOW 5,"sure", (20,345) - (155,445),2
0787 : BUTTON 21,1,"Yes", (20,55) - (60,77),1
0788 : BUTTON 22,1,"No", (75,55) - (115,77),1
0789 : TEXTFONT 2 : TEXTSIZE 24 : forecolor
      BLACK : backcolor CYAN : CLS
0790 : CALL MOVETO (25,40) : PRINT "OK ?"
0791 : WHILE DIALOG(0)<>1 : WEND
0792 : PushedButton=DIALOG(1)
0793 : SELECT CASE PushedButton
0794 : CASE 21 : Answer=YES
0795 : CASE 22 : Answer=NO
0796 : END SELECT
0797 : WINDOW CLOSE 5
0798 : RETURN
0799 :
0800 : SetHisData:
0801 : CLOSE #4 : OPEN HisTmpFile $ FOR
      INPUT AS #4
0802 : OPEN BaseName $ + ".HIS" FOR APPEND
      AS #3
0803 : PRINT #3,DATE $ : PRINT #3,HisCount
0804 : FOR Count=1 TO HisCount
0805 :   INPUT #4,fEditMode : PRINT #3,fEdit
      Mode
0806 :   INPUT #4,fHisSpe : PRINT #3,fHisSpe
0807 :   INPUT #4,ffPnt : PRINT #3,ffPnt
0808 :   INPUT #4,fBuf $ : PRINT #3,fBuf $
0809 : NEXT
0810 : CLOSE #3 : CLOSE #4 : HisCount=0
0811 : KILL HisTmpFile $
0812 : OPEN HisTmpFile $ FOR OUTPUT AS #4
0813 : RETURN
0814 :
0815 : ConfirmSave:
0816 : Msg $ (1) = "Current file has not been saved."
0817 : Msg $ (2) = "Do you want to save it before
      proceeding."
0818 : MsgBoxType=4 : GOSUB MsgBoxWin
0819 : IF Answer=YES THEN
0820 :   GOSUB FileSave
0821 : END IF
0822 : RETURN
0823 :
0824 : CheckESC:
0825 :   IF (XP=0) AND (YP>9) AND (YP<
      12) THEN
0826 :     IF MarkData<>0 THEN GOSUB Disp
      Data
0827 :     WINDOW CLOSE WorkWindow : LOOP=
      ESCAPE
0828 :     YP=0 : EscFlag=FLAG.ON
0829 :     END IF
0830 : RETURN
0831 :
0832 : FileDelete:
0833 :   GOSUB FileClose : IF Answer = IN THEN
      RETURN
0834 :   CLOSE
0835 :   Msg $ (1) = "File deletion" : Msg $ (2) =
      "Select a file with no extension." : GOSUB
      Messenger
0836 :   DelFile $ = FILES $ (1,"TEXT") : WINDOW
      CLOSE MsgWindow
0837 :   IF DelFile $ = "" THEN RETURN
0838 :   OPEN DelFile $ FOR INPUT AS #1
0839 :   INPUT #1,check $
0840 :   IF check $ <> "DNA GAP EDITOR" THEN
0841 :     Msg $ (1) = "This is not in Takushi's format."
0842 :     Msg $ (2) = "Click this window." : GOSUB
      Messenger
0843 :     WHILE MOUSE(0)<>1 : WEND
0844 :     CLOSE #1
0845 :     WINDOW CLOSE MsgWindow : RETURN
0846 :   ELSE
0847 :     INPUT #1,FileName $ : INPUT #1,Folder
      $ : INPUT #1,FileCount : CLOSE #1
0848 :   END IF
0849 :   BaseName $ = FileName $
0850 :   WINDOW 3,"", (100,100) - (600,400),3
0851 :   BUTTON 1,1,"OK", (150,260) - (240,275),1
0852 :   BUTTON 2,1,"CANCEL", (250,260) - (340,275),
      1
0853 :   TEXTFONT 2 : TEXTSIZE 12 : forecolor
      BLACK : backcolor CYAN : CLS
0854 :   LOCATE 1,26 : PRINT "FILE DELETE
      MODE"
0855 :   LOCATE 3,1 : PRINT "Delet file list"
0856 :
0857 :   forecolor BLUE
0858 :   LOCATE 5,5 : PRINT DelFile $ : LOCATE
      6,5 : PRINT BaseName $ + ".HIS"
0859 :   Count=0
0860 :   FOR I=1 TO FileCount+1
0861 :     LOCATE Count+7,5 : PRINT BaseName
      $ + FileExp $ (I+1);
0862 :     Count=Count+1
0863 :     LOCATE Count+7,5 : PRINT BaseName
      $ + FileExp $ (I+1);
0864 :     Count=Count+1
0865 :   NEXT
0866 :   forecolor RED
0867 :   LOCATE 21,22 : PRINT "Will you delete
      file."
0868 :   WHILE DIALOG(0)<>1 : WEND
0869 :   PushedButton=DIALOG(1)
0870 :   WINDOW CLOSE 3

```

```

0871 : IF PushedButton = 2 THEN RETURN
0872 : KILL DelFile $ : KILL BaseName $ + ".HIS"
      : KILL BaseName $ + ".DM"
0873 : KILL BaseName $ + ".IM"
0874 : FOR I=1 TO FileCount
0875 :   KILL BaseName $ + FileExp $ (I) : KILL
      BaseName $ + FileExp $ (I+10)
0876 : NEXT
0877 : RETURN
0878 :
0879 : GetFileName:
0880 : IF OpenFlag=FLAG.ON THEN
0881 :   Msg $ (1)="File already exist." : Msg $ (2)=
      "" : GOSUB Messenger
0882 :   FOR I=1 TO 8000 : NEXT
0883 :   WINDOW CLOSE MsgWindow : RETURN
0884 : END IF
0885 : '--- Select a file ---
0886 : Msg $ (1)="Select a data folder, then open
      a file with no extension." : Msg $ (2)="" :
      GOSUB Messenger
0887 : FileName $ =FILES $ (1,"TEXT")
0888 : WINDOW CLOSE MsgWindow
0889 : IF FileName $ ="" THEN
0890 :   SelectFile=YES : RETURN
0891 : END IF
0892 : '--- ERROR CHECK
0893 : OPEN FileName $ FOR INPUT AS #1
0894 : INPUT #1,check $
0895 : IF check $ <> "DNA GAP EDITOR" THEN
0896 :   CLOSE #1
0897 :   Msg $ (1)="This is not in Takushi's format"
0898 :   Msg $ (2)="Click this window." : GOSUB
      Messenger
0899 :   WHILE MOUSE(0) <> 1 : WEND
0900 :   WINDOW CLOSE MsgWindow : RETURN
0901 : ELSE
0902 :   INPUT #1,FileName $ : INPUT #1,Folder
      $ : INPUT #1,FileCount
0903 :   CLOSE #1
0904 : END IF
0905 : '--- Get file version ---
0906 : GOSUB GetFileVer
0907 : IF FileVer>4 THEN
0908 :   SelectFile=YES : RETURN
0909 : END IF
0910 : BaseName $ =FileName $
0911 : '--- Open structure file ---
0912 : StructFile $ =BaseName $ + FileExp $ (File
      Ver+1)
0913 : OPEN StructFile $ FOR INPUT AS #1
0914 : MaxLength=0 : INPUT #1,SpeNum
0915 : FOR Count=1 TO SpeNum
0916 :   INPUT #1,SpeName $ (Count),LenDNA
      (Count)
0917 :   IF LenDNA (Count)>MaxLength THEN
      MaxLength=LenDNA (Count)
0918 : NEXT
0919 : CLOSE
0920 : '--- COPY
0921 : FOR Spe=1 TO SpeNum
0922 :   ChrCheck $ ="" : Count=0 : Work $ =
      SpeName $ (Spe)
0923 :   WHILE ChrCheck $ =""
0924 :     ChrCheck $ =MID $ (Work $,2,1)
0925 :     Count=Count+1
0926 :     Work $ =RIGHT $ (Work $,LEN(Work $)-
      1)
0927 :   WEND
0928 :   ItemName $ (Spe)=LEFT $ (Work $,5)
0929 : NEXT
0930 : '--- Confirm ---
0931 : ItemNum=SpeNum
0932 : BoxMesColor1=BLUE : BoxMes1 $ ="File
      name-" + BaseName $
0933 : BoxMesColor2=BLACK : BoxMes2 $ =
      "Species => " + STR $ (SpeNum)
0934 : bTitleColor =RED : bTitle $ ="Sure ?"
0935 : BorderTitle $ ="File infomation" : ItemTitle
      $ ="No Species BP"
0936 : GOSUB SelectBox
0937 : IF SelectButton=YES THEN
0938 :   SelectFile=YES
0939 :   TASK $ ="FileOpen"
0940 : END IF
0941 : RETURN
0942 :
0943 :
0944 : SelectBox:
0945 : WINDOW 10,"Menu Box", (25,45) - (595,350),3
0946 : TEXTFONT 3 : TEXTSIZE 12
0947 : backcolor CYAN : CLS
0948 : BUTTON 1,1,"Yes", (40,40) - (72,60),1
0949 : BUTTON 2,1,"No", (90,40) - (122,60),1
0950 : IF ItemNum>30 THEN
0951 :   BUTTON 3,1,"Up", (360,40) - (392,60),1
0952 :   BUTTON 4,1,"Down", (410,40) - (455,60),1
0953 :   LOCATE 2,45 : forecolor 33 : PRINT "P
      AGE";
0954 : END IF
0955 : forecolor BoxMesColor1 : LOCATE 1,1 :
      PRINT BoxMes1 $
0956 : forecolor BoxMesColor2 : LOCATE 1,48 :
      PRINT BoxMes2 $
0957 : forecolor bTitleColor : LOCATE 2,5 :
      PRINT bTitle $

```

```

0958 : Work=LEN(BorderTitle $) + 4 : AstarNum=      (Count)=1 : NEXT
        INT((76-Work)/2)
0959 : IF ( AstarNum*2+Work ) < 75 THEN
        AstarNum=AstarNum+1
0960 : forecolor BLACK      : LOCATE 6,1
0961 : PRINT STRING $ (AstarNum,"*");"
        ";BorderTitle $;" ";STRING $ (AstarNum,"*");
0962 : TopNum=1 : GOSUB Listing
0963 : LOOP=1
0964 : WHILE LOOP
0965 :   WHILE (DIALOG(0)<>1) : WEND
0966 :   ButtonPushed=DIALOG(1)
0967 :   SELECT CASE ButtonPushed
0968 :     CASE 1 : LOOP=0 : SelectButton=YES
0969 :     CASE 2 : LOOP=0 : SelectButton=NO
0970 :     CASE 3 : TopNum=31 : GOSUB Listing
0971 :     CASE 4 : TopNum=1 : GOSUB Listing
0972 :   END SELECT
0973 : WEND
0974 : WINDOW CLOSE 10
0975 : RETURN
0976 :
0977 :
0978 : Listing:
0979 :   Column=5 : Row=9 : NumCT=TopNum
0980 :   backcolor WHITE
0981 :   FOR J=1 TO 11 : LOCATE 7+J,4 : PRINT
        STRING $ (130," "); : NEXT J
0982 :   FOR J=1 TO 30
0983 :     IF Row=9 THEN
0984 :       forecolor GREEN : LOCATE 8,Column :
        PRINT ItemTitle $
0985 :       forecolor BLACK
0986 :     END IF
0987 :     LOCATE Row,Column : PRINT USING
        "# #";NumCT;
0988 :     LOCATE Row,Column+3 : PRINT Item
        Name $ (NumCT);
0989 :     LOCATE Row,Column+10 : PRINT Len
        DNA (NumCT);
0990 :     Row=Row+1
0991 :     IF Row=19 THEN
0992 :       Row=9 : Column=Column+23
0993 :     END IF
0994 :     NumCT=NumCT+1 : IF NumCT>ItemNum
        THEN J=30
0995 :   NEXT J
0996 : RETURN
0997 :
0998 : GetFileVer:
0999 : WINDOW CLOSE 1
1000 : WINDOW 7,"",(100,100)-(550,230),3
1001 : FOR Count=1 TO FileCount : ButtonSw
        (Count)=1 : NEXT
1002 : TEXTFONT 2 : TEXTSIZE 24 : forecolor
        33 : backcolor 341 : CLS
1003 : CALL MOVETO (127,24) : PRINT "Select
        file Version"
1004 : CALL MOVETO (25,90) : PRINT "Ver";
1005 : TEXTSIZE 12
1006 : CALL MOVETO (25,50) : PRINT "File =";
        FileName $
1007 : BUTTON 21,ButtonSw(1),"0",(12*6,12*6)-
        (12*8,12*8),1
1008 : BUTTON 22,ButtonSw(2),"1",(12*9,12*6)-
        (12*11,12*8),1
1009 : BUTTON 23,ButtonSw(3),"2",(12*12,12*6)-
        (12*14,12*8),1
1010 : BUTTON 24,ButtonSw(4),"3",(12*15,12*6)-
        (12*17,12*8),1
1011 : BUTTON 25,ButtonSw(5),"4",(12*18,12*6)-
        (12*20,12*8),1
1012 : BUTTON 26,1,"CANCEL",(12*21,12*6)-
        (12*26,12*8),1
1013 : WHILE DIALOG(0)<>1 : WEND
1014 : PushedButton=DIALOG(1)
1015 : FileVer=PushedButton-21
1016 : WINDOW CLOSE 7
1017 : RETURN
1018 :
1019 : Messenger:
1020 : WINDOW MsgWindow,"",(140,330)-(500,400),
        2
1021 : backcolor GREEN : forecolor WHITE : CLS
1022 : TEXTSIZE 12 : TEXTFONT 2 : PRINT
1023 : FOR Num=1 TO 2 : PRINT Msg $ (Num) :
        NEXT
1024 : PRINT Msg $ (3);
1025 : RETURN
1026 :
7.2 ファイルコンバータ
0001 : ' * * * * *
0002 : ' *
0003 : ' * TITLE : Converter Ver.2.1 for DNA Gap
        Editor
0004 : ' *
0005 : ' * File Name : DGE-Converter Ver.2.1
0006 : ' *
0007 : ' * rev 0.0 : Last modified on December 15th,
        1992
0008 : ' *
0009 : ' * AUTHOR : K. Takushi ( Ryukyu univ. )
0010 : ' *
0011 : ' * * * * *

```

```

0012 : '
0013 : '
0014 : '==== INITIALIZE =====
0015 : DEFINT A-Z : OPTION BASE 1
0016 :
0017 : BLACK = 33 : WHITE = 30 : RED = 205 :
      GREEN = 341
0018 : BLUE = 409 : YELLOW = 69 : MAGENTA
      = 137 : CYAN = 273
0019 : NO = 1 : YES = 2 : CANCEL = 3 :
      ESCAPE = 0 : IN = 1 : LOOP = 1
0020 : SpeLimit=50 : MsgWindow=7
0021 :
0022 : DIM SpeName $ (SpeLimit),LenDNA(SpeLimit),
      NumTBL(SpeLimit,100)
0023 : DIM RectTBL(SpeLimit,100),dtDNA%(SpeLimit,
      500),SpeCellNum(SpeLimit)
0024 : DIM TopNum(SpeLimit),FileExp $ (16)
0025 :
0026 : DEF FNvData(cChr $,Pt) = INSTR("TCAG*",
      MID$(cChr $,Pt,1))
0027 : DEF FNpData(KoSpe,NoCell,DataPoint) =
      DataPoint - NumTBL(KoSpe,NoCell)
0028 : DEF FNnData(KoSpe,NoCell) = NumTBL(Ko
      Spe,NoCell + 1) - NumTBL(KoSpe,NoCell)
0029 :
0030 : FileExp $ (1) = ".I0" : FileExp $ (2) = ".I1" : File
      Exp $ (3) = ".I2" : FileExp $ (4) = ".I3"
0031 : FileExp $ (5) = ".I4" : FileExp $ (6) = ".IM"
0032 : FileExp $ (11) = ".D0" : FileExp $ (12) = ".D1" :
      FileExp $ (13) = ".D2" : FileExp $ (14) = ".D3"
0033 : FileExp $ (15) = ".D4" : FileExp $ (16) = ".DM" :
0034 : FileExp $ (10) = ".HIS" :
0035 :
0036 : MENU RESET
0037 : '*** MENU BAR ***
0038 : MENU 1,0,1,"File "
0039 : MENU 2,0,1,"Convert"
0040 : '*** FILE MENU ***
0041 : MENU 1,1,1,"Quit" : cmdkey 1,1,"Q"
0042 : '*** CONVERT MENU ***
0043 : MENU 2,1,1,"Murata => Takushi"
0044 : MENU 2,2,1,"Takushi => Murata"
0045 :
0046 : ON MENU GOSUB MainMenu : MENU ON
0047 :
0048 : WINDOW 1,"", (50,50) - (600,120),3
0049 : backcolor CYAN : forecolor BLACK :
      TEXTFONT 1: TEXTSIZE 24
0050 : CLS
0051 : PRINT " CONVERTER Ver.2.1"
0052 : PRINT " for DNA
      GAP EDITOR";
0053 : FOR J=1 TO 100
0054 : FOR K=1 TO 600 : NEXT
0055 : NEXT
0056 : WINDOW CLOSE 1
0057 :
0058 : Program = IN
0059 : WHILE Program : WEND
0060 : END
0061 :
0062 : MainMenu:
0063 : MenuNumber=MENU(0) : MenuItem=MENU
      (1)
0064 : MENU : ON MenuNumber GOSUB FileMenu,
      ConvertMenu
0065 : RETURN
0066 :
0067 : FileMenu:
0068 : ON MenuItem GOSUB Quit
0069 : RETURN
0070 :
0071 : ConvertMenu:
0072 : ON MenuItem GOSUB MurataConvert,
      TakushiConvert
0073 : RETURN
0074 :
0075 : '==== Convert Murata's format into
      Takushi's one =====
0076 : MurataConvert:
0077 : '--- Get file name ---
0078 : SelectFile=NO
0079 : WHILE SelectFile=NO : GOSUB GetMurata
      File : WEND
0080 : IF ConvFile $ = "" THEN RETURN
0081 : '--- Make data file ---
0082 : GOSUB MurataSetFName
0083 : OPEN DataFile $ FOR OUTPUT AS #1 :
      CLOSE #1
0084 : OPEN ConvFile $ FOR INPUT AS #2
0085 : OPEN "R", #1,DataFile $ ,256
0086 : FIELD #1,254 AS fDNA $ ,2 AS fLength $
0087 : INPUT #2,SpeNum
0088 : MODE $ = "Murata's format => Takushi's
      format" : Work $ = "data file" : GOSUB Conv-
      MsgOpen
0089 : Rec=1 : eDNA $ = "" : RectTBL(1,1)=1
0090 : FOR Spe=1 TO SpeNum
0091 : INPUT #2,SpeName $ (Spe)
0092 : INPUT #2,LenDNA(Spe),CheckPara1,
      CheckPara2
0093 : TotalCount=0 : LenCell=0 : NumTBL
      (Spe,1) = 1 : Cell=1
0094 : GOSUB ConvMsgWrite
0095 : WHILE (TotalCount <> LenDNA(Spe))

```

```

0096 : INPUT #2,wDNA $
0097 : TotalCount=TotalCount+LEN(wDNA $)
0098 : LenCell=LenCell+LEN(wDNA $)
0099 : IF LenCell>200 THEN
0100 : Rest=200-LEN(eDNA $)
0101 : eDNA $ =eDNA $ +LEFT $(wDNA $,
Rest)
0102 : Cell=Cell+1 : NumTBL(Spe,Cell)=
NumTBL(Spe,Cell-1)+200
0103 : pRec=Rec : GOSUB PutRecord
0104 : RecTBL(Spe,Cell-1)=Rec : Rec=Rec
+1
0105 : eDNA $ =RIGHT $(wDNA $,LEN(wD
NA $)-Rest)
0106 : LenCell=LEN(eDNA $)
0107 : ELSE
0108 : eDNA $ =eDNA $ +wDNA $
0109 : END IF
0110 : LOCATE 4,19 : PRINT USING "###
##";Cell;
0111 : LOCATE 5,19 : PRINT USING "###
##";TotalCount;
0112 : WEND
0113 : IF LEN(eDNA $)>0 THEN
0114 : pRec=Rec : GOSUB PutRecord
0115 : Cell=Cell+1 : NumTBL(Spe,Cell)=Num
TBL(Spe,Cell-1)+200
0116 : RecTBL(Spe,Cell-1)=Rec : Rec=Rec+1
0117 : LenCell=0 : eDNA $ =""
0118 : END IF
0119 : SpeCellNum(Spe)=Cell-1
0120 : GOSUB ConvMsgWrite
0121 : NEXT
0122 : LastRecord=Rec-1 : CLOSE
0123 : '--- Make structure file ---
0124 : '--- No. of Species , Names, Data Length /
Species ---
0125 : Work $ ="structure file" : GOSUB ConvMsg
Open
0126 : OPEN InfoFile $ FOR OUTPUT AS #1
0127 : PRINT #1,SpeNum
0128 : FOR Spe=1 TO SpeNum
0129 : PRINT #1,SpeName $(Spe)
0130 : PRINT #1,LenDNA (Spe)
0131 : NEXT
0132 : '--- Make data allocation table ---
0133 : FOR Spe=1 TO SpeNum
0134 : PRINT #1,SpeCellNum(Spe)
0135 : FOR Cell=1 TO SpeCellNum(Spe)
0136 : PRINT #1,NumTBL(Spe,Cell),RecTBL
(Spe,Cell)
0137 : NEXT
0138 : NEXT
0139 : PRINT #1,LastRecord
0140 : CLOSE
0141 : '--- Make opening file ---
0142 : FileCount=1
0143 : OPEN NameFile $ FOR OUTPUT AS #1
0144 : PRINT #1,"DNA GAP EDITOR"
0145 : PRINT #1,FileName $
0146 : PRINT #1,Folder $
0147 : PRINT #1,FileCount
0148 : CLOSE #1
0149 : '--- Make history record ---
0150 : OPEN HistoryFile $ FOR OUTPUT AS #1:
CLOSE
0151 : '--- Make structure file ver 0 ---
0152 : Work $ ="structure file 0" : GOSUB Conv
MsgOpen
0153 : OPEN IOFile $ FOR OUTPUT AS #1
0154 : PRINT #1,SpeNum
0155 : FOR Spe=1 TO SpeNum
0156 : PRINT #1,SpeName $(Spe)
0157 : PRINT #1,LenDNA (Spe)
0158 : NEXT
0159 : '--- Make data allocation table ---
0160 : FOR Spe=1 TO SpeNum
0161 : PRINT #1,SpeCellNum(Spe)
0162 : FOR Cell=1 TO SpeCellNum(Spe)
0163 : PRINT #1,NumTBL(Spe,Cell),RecTBL
(Spe,Cell)
0164 : NEXT
0165 : NEXT
0166 : PRINT #1,LastRecord
0167 : CLOSE
0168 : '--- Make data file ---
0169 : Work $ ="data file 0" : GOSUB ConvMsgOpen
0170 : OPEN D0File $ FOR OUTPUT AS #2 :
CLOSE #2
0171 : OPEN "R",#2,D0File $,256
0172 : FIELD #2,254 AS wDNA $,2 AS wLength
$
0173 : OPEN DataFile $ AS #1 LEN=256
0174 : FIELD #1,254 AS fDNA $,2 AS fLength $
0175 : FOR Rec=1 TO LastRecord
0176 : gRec=Rec : GOSUB GetRecord
0177 : LSET wDNA $ =eDNA $ +STRING $(254
-Length,"r")
0178 : LSET wLength $ = MKI $(Length)
0179 : PUT #2,Rec
0180 : NEXT
0181 : CLOSE
0182 : GOSUB ConvMsgClose
0183 : RETURN
0184 :
0185 : '==== Convert Takushi's format into

```

```

Murata's format =====
0186 : TakushiConvert:
0187 : '--- GetTakushi's format file ---
0188 : SelectFile=NO
0189 : WHILE SelectFile=NO : GOSUB Get
      TakushiFile : WEND
0190 : IF ConvFile $ ="" THEN RETURN
0191 : '--- Make Murata's format file ---
0192 : InfoFile $ =BaseName $ +FileExp $ (FileVer
      +1)
0193 : DataFile $ =BaseName $ +FileExp $ (FileVer
      +11)
0194 : GOSUB GetMurataName
0195 : MurataFile $ =Folder $ +Work $
0196 : '--- Read No. of Species, Names, Data Length
      / Species ---
0197 : OPEN InfoFile $ FOR INPUT AS #1
0198 : MaxLength=0 : INPUT #1,SpeNum
0199 : FOR Spe=1 TO SpeNum
0200 :   INPUT #1,SpeName $ (Spe),LenDNA(Spe)
0201 :   IF LenDNA(Spe)>MaxLength THEN Max
      Length=LenDNA(Spe)
0202 : NEXT
0203 : '--- Read data allocation table ---
0204 : FOR Spe=1 TO SpeNum
0205 :   INPUT #1,SpeCellNum(Spe)
0206 :   FOR Cell=1 TO SpeCellNum(Spe)
0207 :     INPUT #1,NumTBL(Spe,Cell),RecTBL
      (Spe,Cell)
0208 :   NEXT
0209 : NEXT
0210 : CLOSE
0211 : '--- Open Takushi's format file ---
0212 : OPEN DataFile $ AS #1 LEN = 256
0213 : FIELD #1,254 AS fDNA $,2 AS fLength $
0214 : '--- Open Murata's format file ---
0215 : OPEN MurataFile $ FOR OUTPUT AS #2
0216 : MODE $ ="Takushi's format => Murata's
      format" : Work $ ="data file": GOSUB Conv
      MsgOpen
0217 : GOSUB ConvMsgWrite
0218 : CheckPara1=1
0219 : PRINT #2,SpeNum
0220 : FOR Spe=1 TO SpeNum
0221 :   PRINT #2,SpeName $ (Spe)
0222 :   PRINT #2,LenDNA(Spe),CheckPara1,Len
      DNA(Spe)
0223 :   TotalCount=0
0224 :   GOSUB ConvMsgWrite
0225 :   FOR Cell=1 TO SpeCellNum(Spe)
0226 :     gRec=RecTBL(Spe,Cell)
0227 :     GOSUB GetRecord
0228 :     PRINT #2,eDNA $
0229 :     TotalCount=TotalCount+Length
0230 :     LOCATE 4,19 : PRINT USING "# ##
      ##";Cell;
0231 :     LOCATE 5,19 : PRINT USING "# ##
      ##";TotalCount;
0232 :   NEXT
0233 : NEXT
0234 : CLOSE
0235 : GOSUB ConvMsgClose
0236 : RETURN
0237 :
0238 :
0239 : PutRecord:
0240 :   Length=LEN(eDNA $)
0241 :   eDNA $ =eDNA $ +STRING $(254-Length,
      "r")
0242 :   LSET fDNA $ =eDNA $
0243 :   LSET fLength $ =MKI $(Length)
0244 :   PUT #1,pRec
0245 : RETURN
0246 :
0247 : GetRecord:
0248 :   GET #1,gRec
0249 :   Length=CVI(fLength $)
0250 :   eDNA $ =LEFT $(fDNA $,Length)
0251 : RETURN
0252 :
0253 : PickUpFName:
0254 :   DirPoint=100 : LastDirPoint=0 : wcopy $ =
      Work $
0255 :   WHILE DirPoint
0256 :     DirPoint=INSTR(Work $,":")
0257 :     LastDirPoint=LastDirPoint+DirPoint
0258 :     Work $ =RIGHT $(Work $,LEN(Work $)-
      DirPoint)
0259 :   WEND
0260 :   Folder $ =LEFT $(wcopy $,LastDirPoint)
0261 :   KSI=INSTR(Work $, ".")-1
0262 :   KSI=KSI+LastDirPoint
0263 :   IF KSI>-1 THEN Work $ =LEFT $(wcopy
      $,KSI)
0264 : RETURN
0265 :
0266 : MurataSetFName:
0267 :   Work $ =ConvFile $ : GOSUB PickUpFName
0268 :   FileName $ =Work $
0269 :   DataFile $ =Work $ +".DM"
0270 :   InfoFile $ =Work $ +".IM"
0271 :   HistoryFile $ =Work $ +".HIS"
0272 :   D0File $ =Work $ +".D0"
0273 :   I0File $ =Work $ +".I0"
0274 :   NameFile $ =Work $
0275 : RETURN

```

```

0276 :                               SpeName $ (Spe)
0277 : GetMurataName:                0321 :   WHILE ChrCheck $ =""
0278 :   WINDOW 4,"", (50,100) - (450,200),3  0322 :     ChrCheck $ =MID $ (Work $,2,1)
0279 :   forecolor BLACK : backcolor WHITE :  0323 :     Count=Count+1
   TEXTFONT 2 : TEXTSIZE 12          0324 :     Work $ =RIGHT $ (Work $,LEN(Work $)
0280 :   Work $ =""                      -1)
0281 :   WHILE Work $ =""                0325 :   WEND
0282 :   CLS                               0326 :     ItemName $ (Spe)=LEFT $ (Work $,5)
0283 :   LOCATE 2,1 : INPUT "name of destined  0327 :   NEXT
   file ";Work $                      0328 :   '--- Confirm ---
0284 :   WEND                               0329 :   ItemNum=SpeNum
0285 :   WINDOW CLOSE 4                   0330 :   BoxMesColor1=BLUE : BoxMes1 $ ="File
0286 : RETURN                               name - "+ConvFile $
0287 :                                     0331 :   BoxMesColor2=BLACK : BoxMes2 $ =
0288 : GetMurataFile:                       "Species => "+STR $ (SpeNum)
0289 :   '--- Select file name ---          0332 :   bTitleColor =RED : bTitle $ ="Sure ?"
0290 :   Msg $ (1)="Select a data folder, then open  0333 :   BorderTitle $ ="File Infomation" : ItemTitle
   a file with extension "+CHR $ (34)+"mrt"    $ ="No Species BP"
   +CHR $ (34)+"."                          0334 :   GOSUB SelectBox
0291 :   Msg $ (2)="" : GOSUB Messenger      0335 :   IF SelectButton=YES THEN SelectFile=
0292 :   ConvFile $ =FILES $ (1,"TEXT") : WINDOW  YES
   CLOSE MsgWindow                          0336 : RETURN
0293 :   IF ConvFile $ ="" THEN            0337 :
0294 :     SelectFile=YES : RETURN          0338 : GetTakushiFile:
0295 :   END IF                               0339 :   '--- Select file name ---
0296 :   '--- Error check ---                0340 :   Msg $ (1)="Select a data folder, then open
0297 :   OPEN ConvFile $ FOR INPUT AS #1      a file with no extension."
0298 :   INPUT #1,SpeNum : INPUT #1,FileCheck  0341 :   Msg $ (2)="" : GOSUB Messenger
   $ : CLOSE #1                              0342 :   ConvFile $ =FILES $ (1,"TEXT") : WINDOW
0299 :   IF LEFT $ (FileCheck $,2) <> "==" THEN  CLOSE MsgWindow
0300 :     Msg $ (1)="This is not in Murata's format."  0343 :   IF ConvFile $ ="" THEN
     Msg $ (2)="Click this window."          0344 :     SelectFile=YES : RETURN
0301 :     GOSUB Messenger                    0345 :   END IF
0302 :     WHILE MOUSE(0) <> 1 : WEND          0346 :   '--- Error check ---
0303 :     WINDOW CLOSE MsgWindow : RETURN    0347 :   OPEN ConvFile $ FOR INPUT AS #1
0304 :   END IF                               0348 :   INPUT #1,check $
0305 :   '--- Open Murata's format file ---    0349 :   IF check $ <> "DNA GAP EDITOR" THEN
0306 :   OPEN ConvFile $ FOR INPUT AS #1      Msg $ (1)="This is not a data file." : Msg
0307 :   INPUT #1,SpeNum                      $ (2)="Click this window." : GOSUB
0308 :   FOR Spe=1 TO SpeNum                  Messenger
0309 :     INPUT #1,SpeName $ (Spe)            0351 :   WHILE MOUSE(0) <> 1 : WEND
0310 :     INPUT #1,LenDNA (Spe),ChechPara1,  0352 :   WINDOW CLOSE MsgWindow : RETURN
     CheckPara2                              0353 :   ELSE
0311 :     TotalCount=0                       INPUT #1,FileName $ : INPUT #1,Folder
0312 :     WHILE (TotalCount <> LenDNA (Spe))    $ : INPUT #1,FileCount
0313 :       INPUT #1,wDNA $                   0355 :   END IF
0314 :       TotalCount=TotalCount+LEN(wDNA $)  0356 :   CLOSE #1
0315 :     WEND                               0357 :   '--- Get file version ---
0316 :   NEXT                               0358 :   GOSUB GetFileVer
0317 :   CLOSE #1                             0359 :   IF FileVer > 4 THEN
0318 :   '--- Copy ---                          SelectFile=YES : RETURN
0319 :   FOR Spe=1 TO SpeNum                  0360 :   END IF
0320 :     ChrCheck $ ="" : Count=0 : Work $ =  0361 :   END IF
   0362 :   BaseName $ =FileName $

```



```

0363 : '--- Open structure file ---
0364 : InfoFile $ =BaseName $ +FileExp $ (FileVer
+1)
0365 : OPEN InfoFile $ FOR INPUT AS #1
0366 : MaxLength=0 : INPUT #1,SpeNum
0367 : FOR Count=1 TO SpeNum
0368 : INPUT #1,SpeName $ (Count),LenDNA
(Count)
0369 : IF LenDNA(Count)>MaxLength THEN
MaxLength=LenDNA(Count)
0370 : NEXT
0371 : CLOSE #1
0372 : '--- Copy ---
0373 : FOR Spe=1 TO SpeNum
0374 : ChrCheck $ ="=" : Count=0 : Work $ =
Spe.Name $ (Spe)
0375 : WHILE ChrCheck $ ="="
0376 : ChrCheck $ =MID $ (Work $ ,2,1)
0377 : Count=Count+1
0378 : Work $ =RIGHT $ (Work $ ,LEN(Work $ )
-1)
0379 : WEND
0380 : ItemName $ (Spe)=LEFT $ (Work $ ,5)
0381 : NEXT
0382 : '--- Confirm ---
0383 : ItemNum=SpeNum
0384 : BoxMesColor1=BLUE : BoxMes1 $ ="File
name - "+BaseName $
0385 : BoxMesColor2=BLACK : BoxMes2 $ =
"Species => "+STR $ (SpeNum)
0386 : bTitleColor =RED : bTitle $ ="Sure ?"
0387 : BorderTitle $ ="File Infomation" : ItemTitle
$ ="No Species BP"
0388 : GOSUB SelectBox
0389 : IF SelectButton=YES THEN SelectFile=
YES
0390 : RETURN
0391 :
0392 : SelectBox:
0393 : WINDOW 10,"Menu Box", (25,45)-(595,350),3
0394 : TEXTFONT 3 : TEXTSIZE 12
0395 : backcolor CYAN : CLS
0396 : BUTTON 1,1,"Yes", (40,40) - (72,60),1
0397 : BUTTON 2,1,"No", (90,40) - (122,60),1
0398 : IF ItemNum>30 THEN
0399 : BUTTON 3,1,"Up", (360,40) - (392,60),1
0400 : BUTTON 4,1,"Down", (410,40) - (455,60),1
0401 : LOCATE 2,45 : forecolor 33 : PRINT
"PAGE";
0402 : END IF
0403 : forecolor BoxMesColor1 : LOCATE 1,1 :
PRINT BoxMes1 $
0404 : forecolor BoxMesColor2 : LOCATE 1,48 :
PRINT BoxMes2 $
0405 : forecolor bTitleColor : LOCATE 2,5 : PRI
NT bTitle $
0406 : Work=LEN(BorderTitle $ )+4 : AstarNum=
INT((76-Work)/2)
0407 : IF ( AstarNum*2+Work ) < 75 THEN
AstarNum=AstarNum+1
0408 : forecolor BLACK : LOCATE 6,1
0409 : PRINT STRING $ (AstarNum,"*"); " ";Border
Title $ ; " ";STRING $ (AstarNum,"*");
0410 : TopNum=1 : GOSUB Listing
0411 : LOOP=1
0412 : WHILE LOOP
0413 : WHILE (DIALOG (0)<>1) : WEND
0414 : ButtonPushed=DIALOG (1)
0415 : SELECT CASE ButtonPushed
0416 : CASE 1 : LOOP=0 : SelectButton=YES
0417 : CASE 2 : LOOP=0 : SelectButton=NO
0418 : CASE 3 : TopNum=31 : GOSUB Listing
0419 : CASE 4 : TopNum=1 : GOSUB Listing
0420 : END SELECT
0421 : WEND
0422 : WINDOW CLOSE 10
0423 : RETURN
0424 :
0425 : Listing:
0426 : Column=5 : Row=9 : NumCT=TopNum
0427 : backcolor WHITE
0428 : FOR J=1 TO 11 : LOCATE 7+J,4 : PRINT
STRING $ (130," "); : NEXT J
0429 : FOR J=1 TO 30
0430 : IF Row=9 THEN
0431 : forecolor GREEN : LOCATE 8,Column :
PRINT ItemTitle $
0432 : forecolor BLACK
0433 : END IF
0434 : LOCATE Row,Column : PRINT USING "
##";NumCT;
0435 : LOCATE Row,Column+3 : PRINT Item
Name $ (NumCT);
0436 : LOCATE Row,Column+10 : PRINT Len
DNA(NumCT);
0437 : Row=Row+1
0438 : IF Row=19 THEN
0439 : Row=9 : Column=Column+23
0440 : END IF
0441 : NumCT=NumCT+1 : IF NumCT>ItemNum
THEN J=30
0442 : NEXT J
0443 : RETURN
0444 :
0445 : ConvMsgOpen:
0446 : WINDOW 2,"", (50,50) - (400,100),3

```

```

0447 : backcolor YELLOW : TEXTFONT 3 :
      TEXTSIZE 12 : CLS
0448 : forecolor RED : PRINT "MODE : ";MODE
      $
0449 : forecolor BLUE : PRINT "WORK : ";Work
      $
0450 : forecolor BLACK : PRINT "Species = ";
      SpeNum;
0451 : WINDOW 3,"", (50,100) - (450,200),3
0452 : forecolor BLUE : backcolor WHITE :
      TEXTFONT 3 : TEXTSIZE 12 : CLS
0453 : RETURN
0454 :
0455 : ConvMsgClose:
0456 : WINDOW CLOSE 2 : WINDOW CLOSE 3 :
      WINDOW CLOSE 1
0457 : WINDOW 3,"", (50,100) - (450,200),3
0458 : forecolor GREEN : CLS
0459 : LOCATE 2,1 : PRINT "COMPLETE !";
0460 : LOCATE 3,1 : PRINT "Click this window.";
0461 : WHILE MOUSE(0) < > 1 : WEND
0462 : WINDOW CLOSE 3
0463 : RETURN
0464 :
0465 : ConvMsgWrite:
0466 : WINDOW 3 : CLS
0467 : LOCATE 2,1 : PRINT "No. ";Spe;
0468 : LOCATE 2,8 : PRINT SpeName $ (Spe);
0469 : LOCATE 2,23 : PRINT "data length =";
0470 : LOCATE 2,33 : PRINT USING "###";
      LenDNA (Spe);
0471 : LOCATE 4,8 : PRINT "cell count =";
0472 : LOCATE 4,19 : PRINT USING "###";
      Cell;
0473 : LOCATE 5,8 : PRINT "data count =";
0474 : LOCATE 5,19 : PRINT USING "###";
      TotalCount;
0475 : IF SpeNum < 6 THEN
0476 :     FOR K=1 TO 1000 : NEXT
0477 :     END IF
0478 : RETURN
0479 :
0480 : GetFileVer:
0481 : WINDOW CLOSE 1
0482 : WINDOW 7,"", (100,100) - (550,230),3
0483 : FOR Count=1 TO FileCount : ButtonSw
      (Count)=1 : NEXT
0484 : TEXTFONT 2 : TEXTSIZE 24 : forecolor
      33 : backcolor 341 : CLS
0485 : CALL MOVETO (127,24) : PRINT "Select
      file Version"
0486 : CALL MOVETO (25,90) : PRINT "Ver";
0487 : TEXTSIZE 12
0488 : CALL MOVETO (25,50) : PRINT "File = ";
      FileName $
0489 : BUTTON 21,ButtonSw (1),"0", (12*6,12*6) -
      (12*8,12*8),1
0490 : BUTTON 22,ButtonSw (2),"1", (12*9,12*6) -
      (12*11,12*8),1
0491 : BUTTON 23,ButtonSw (3),"2", (12*12,12*6) -
      (12*14,12*8),1
0492 : BUTTON 24,ButtonSw (4),"3", (12*15,12*6) -
      (12*17,12*8),1
0493 : BUTTON 25,ButtonSw (5),"4", (12*18,12*6) -
      (12*20,12*8),1
0494 : BUTTON 26,1,"CANCEL", (12*21,12*6) -
      (12*26,12*8),1
0495 : WHILE DIALOG (0) < > 1 : WEND
0496 : PushedButton=IALOG (1)
0497 : FileVer=PushedButton-21
0498 : WINDOW CLOSE 7
0499 : RETURN
0500 :
0501 : Messenger:
0502 : WINDOW MsgWindow,"", (140,330) - (500,400),
      2
0503 : backcolor GREEN : forecolor BLACK : CLS
0504 : TEXTSIZE 12 : TEXTFONT 2
0505 : PRINT Msg $ (1)
0506 : forecolor WHITE : PRINT Msg $ (2) : PRINT
      Msg $ (3)
0507 : forecolor BLACK : PRINT Msg $ (4);
0508 : FOR Num=1 TO 4: Msg $ (Num)="": NEXT
0509 : RETURN
0510 :
0511 : Quit:
0512 : END
0513 :
7.3 編集記録参照プログラム
0001 : ' *****
0002 : ' *
0003 : ' * TITLE : Tracer of editing
0004 : ' *
0005 : ' * File Name : Tracer of editing Ver 1.0
0006 : ' *
0007 : ' * rev 0.2 : Last modified on December 15th,
      1992
0008 : ' *
0009 : ' * AUTHOR : K. Takushi ( Ryukyu univ. )
0010 : ' *
0011 : ' *****
0012 : '
0013 : '
0014 : '==== INITIALIZE =====

```

```

0015 : DEFINT AZ : OPTION BASE 1
0016 :
0017 : BLACK = 33 : WHITE = 30 : RED = 205 :
      GREEN = 341
0018 : BLUE = 409 : YELLOW = 69 : MAGENTA
      = 137 : CYAN = 273
0019 : NO = 1 : YES = 2 : CANSEL = 3 :
      ESCAPE = 0 : IN = 1 : LOOP = 1 : FLAG.
      OFF=0 : FLAG.ON=1
0020 : DNC(1) = GREEN : DNC(2) = CYAN : DNC
      (3) = MAGENTA
0021 : DNC(4) = BLACK : DNC(5) = YELLOW :
      DNC(6) = BLACK
0022 : DNC(7) = RED
0023 : MsgWindow=7 : DataLimit=500
0024 :
0025 : DIM EditMode(DataLimit),Spe(DataLimit),Data
      Point(DataLimit),EditData$(DataLimit)
0026 : DIM EditDate$(DataLimit)
0027 : EditMode$(1)="CUT" : EditMode$(2)=
      "COPY" : EditMode$(3)="PASTE" :
      EditMode$(4)="INSERT"
0028 : OpenFlag=FLAG.OFF
0029 :
0030 : MENU RESET
0031 : '*** MENU BAR ***
0032 : MENU 1,0,1,"File"
0033 : MENU 2,0,0,"Edit"
0034 : '*** FILE MENU ***
0035 : MENU 1,1,1,"Open" : cmdkey 1,1,"O"
0036 : MENU 1,2,0,"Close"
0037 : MENU 1,3,1,"Quit" : cmdkey 1,3,"Q"
0038 :
0039 : ON MENU GOSUB MainMenu : MENU ON
0040 :
0041 : WINDOW 1,"", (50,50) - (600,120),3
0042 : backcolor CYAN : forecolor BLACK :
      TEXTFONT 1 : TEXTSIZE 24
0043 : CLS
0044 : PRINT "   TRACER OF EDITING Ver.1.0"
0045 : PRINT "                               for
      DNA GAP EDITOR ";
0046 : FOR J=1 TO 1000
0047 : FOR L=1 TO 90: NEXT
0048 : NEXT
0049 : WINDOW CLOSE 1
0050 :
0051 : Program=IN
0052 : WHILE Program
0053 : IF OpenFlag=FLAG.ON THEN
0054 : GOSUB ScreenOperate
0055 : END IF
0056 : WEND

0057 : END
0058 :
0059 : MainMenu:
0060 : MenuNumber=MENU(0) : MenuItem=MENU
      (1)
0061 : MENU : ON MenuItem GOSUB FileOpen,
      FileClose,Quit
0062 : RETURN
0063 :
0064 : DispData:
0065 : WINDOW 1 : TEXTSIZE 12 : TEXTFONT
      2 : backcolor WHITE
0066 : CLS : forecolor RED
0067 : LOCATE 1,1 : PRINT FileName$;
0068 : forecolor GREEN : LOCATE 1,50 : PRINT
      "Total = ";HisData
0069 : forecolor BLUE
0070 : LOCATE 5,4 : PRINT "No."
0071 : LOCATE 5,10 : PRINT "Mode"
0072 : LOCATE 5,17 : PRINT "Speices"
0073 : LOCATE 5,25: PRINT "Data point"
0074 : LOCATE 5,35 : PRINT "Data"
0075 : LOCATE 5,65:PRINT "Date"
0076 : forecolor BLACK
0077 : FOR i=0 TO DispNum-1
0078 : dp=TopSpe+i : low=6+i
0079 : LOCATE low,2: PRINT USING "###";
      dp
0080 : LOCATE low,10 : PRINT EditMode$(Edit
      Mode(dp))
0081 : LOCATE low,20 : PRINT USING "##";
      Spe(dp)
0082 : LOCATE low,25 : PRINT USING "###
      ###";DataPoint(dp)
0083 : LOCATE low,35 : PRINT EditData$(dp)
0084 : LOCATE low,65 : PRINT EditDate$(dp)
0085 : NEXT
0086 : RETURN
0087 :
0088 : ScreenOperate:
0089 : WINDOW 1
0090 : ShowFlag=FLAG.ON
0091 : WHILE ShowFlag
0092 : WHILE DIALOG(0)<>1 AND ShowFlag=
      FLAG.ON : WEND
0093 : IF ShowFlag=FLAG.ON THEN
0094 : Pushed = DIALOG(1)
0095 : SELECT CASE Pushed
0096 : CASE 1 : GOSUB upscroll
0097 : CASE 2 : GOSUB downscroll
0098 : CASE 3 : GOSUB topscroll
0099 : CASE 4 : GOSUB botscroll
0100 : END SELECT

```

```

0101 :   END IF
0102 :   WEND
0103 : RETURN
0104 :
0105 : upscroll:
0106 :   IF HisData <11 THEN RETURN
0107 :   TopSpe=TopSpe+5 : BotSpe=TopSpe+9
0108 :   IF BotSpe>HisData THEN TopSpe=HisData-
0109 :     9
0109 :   BotSpe=TopSpe+DispNum-1
0110 :   GOSUB DispData
0111 : RETURN
0112 :
0113 : downscroll:
0114 :   IF HisData<11 THEN RETURN
0115 :   TopSpe=TopSpe-5
0116 :   IF TopSpe<1 THEN TopSpe=1
0117 :   BotSpe=TopSpe+DispNum-1
0118 :   GOSUB DispData
0119 : RETURN
0120 :
0121 : topscroll:
0122 :   IF HisData<11 THEN RETURN
0123 :   TopSpe=1 : BotSpe=TopSpe+DispNum :
0124 :     GOSUB DispData
0124 : RETURN
0125 :
0126 : botscroll:
0127 :   IF HisData<11 THEN RETURN
0128 :   BotSpe=HisData : TopSpe=BotSpe-9 :
0129 :     GOSUB DispData
0129 : RETURN
0130 :
0131 : FileOpen:
0132 :   IF OpenFlag=FLAG.ON THEN
0133 :     Msg $ (1)="File already exist." : Msg $ (2)=
0134 :       "" : GOSUB Messenger
0134 :     FOR J=1 TO 8000 : NEXT
0135 :     WINDOW CLOSE MsgWindow : RETURN
0136 :   END IF
0137 :   Msg $ (1)="Select a data folder, then open
0138 :     a file with extension "+CHR $ (34)+"HIS"
0139 :     +CHR $ (34)+"."
0138 :   Msg $ (2)="" : GOSUB Messenger
0139 :   FileName $ =FILES $ (1,"TEXT") : WINDOW
0140 :     CLOSE MsgWindow
0140 :   IF FileName $="" THEN RETURN
0141 :   FileExp $ =RIGHT $ (FileName $ ,3)
0142 :   IF FileExp $ <<>"HIS" THEN
0143 :     Msg $ (1)="This file is not available." :
0144 :       Msg $ (2)="" : GOSUB Messenger
0144 :   BEEP
0145 :   FOR J=1 TO 900
0146 :     FOR L=1 TO 80: NEXT
0147 :     NEXT
0148 :     WINDOW CLOSE MsgWindow : RETURN
0149 :   END IF
0150 :   GOSUB ReadData
0151 :   GOSUB IntWindow
0152 :   OpenFlag=FLAG.ON
0153 : RETURN
0154 :
0155 : ReadData:
0156 :   OPEN FileName $ FOR INPUT AS #1
0157 :   Rcount = 0
0158 :   WHILE NOT EOF(1)
0159 :     INPUT #1,Update $
0160 :     INPUT #1,HisCount
0161 :     FOR i=1 TO HisCount
0162 :       Rcount=Rcount+1
0163 :       INPUT #1,EditMode (Rcount)
0164 :       INPUT #1,Spe (Rcount)
0165 :       INPUT #1,DataPoint (Rcount)
0166 :       INPUT #1,EditData $ (Rcount)
0167 :       EditDate $ (Rcount)=Update $
0168 :     NEXT
0169 :   WEND
0170 :   CLOSE
0171 :   HisData=Rcount
0172 : RETURN
0173 :
0174 : IntWindow:
0175 :   WINDOW 1,"Tracer of editing Ver.1.0", (5,45)
0176 :     - (635,325),6
0176 :   MENU 1,2,1 : MENU
0177 :   BUTTON 1,1,"UP", (10,20) - (110,42),1
0178 :   BUTTON 2,1,"DOWN", (120,20) - (220,42),1
0179 :   BUTTON 3,1,"TOP", (230,20) - (340,42),1
0180 :   BUTTON 4,1,"BOTTUM", (350,20) - (460,42),1
0181 :   BotSpe=10 : TopSpe=1
0182 :   IF HisData<10 THEN BotSpe=HisData
0183 :   DispNum=BotSpe
0184 :   GOSUB DispData
0185 : RETURN
0186 :
0187 : FileClose:
0188 :   MENU 1,2,0 : MENU
0189 :   WINDOW CLOSE 1: OpenFlag=FLAG.OFF
0190 :     : ShowFlag=FLAG.OFF
0190 : RETURN
0191 :
0192 : Quit:
0193 :   IF OpenFlag=FLAG.ON THEN
0194 :     OpenFlag=FLAG.OFF
0195 :     WINDOW CLOSE 1
0196 :   END IF

```

```

0197 : Program=ESCAPE : ShowFlag=FLAG.OFF      0203 : TEXTSIZE 12: TEXTFONT 22 : PRINT
0198 : RETURN                                    0204 : FOR Num=1 TO 3 : PRINT Msg$(Num) :
0199 :                                             NEXT
0200 : Messenger:                                0205 : PRINT Msg$(4);
0201 : WINDOW MsgWindow,"", (140,330)-(500,400), 0206 : RETURN
        2                                         0207 :
0202 : bgcolor GREEN : forecolor WHITE : CLS

```

あとがき

(1) 向上点

今回、DNA塩基配列編集プログラムの改良・拡張・移植を行った。操作性・取り扱うデータ量についてはかなり向上した。具体的には、キーボードだけの操作からマウスを使っての操作をも可能にしたこと、扱えるデータの量が10万個（10生物種×1万個の塩基）から12.5倍の125万個（50生物種×2万5千個の塩基）になったことである。このデータ量の拡張が可能になったのは、ランダムファイルを扱うアルゴリズムを確立したことにある。また、村田氏の『DNAGE』では各生物種を変数として置く事ができなかったのを可能にした。その結果、各ルーチンでの記述を単純化することができた。例えば、いままでループ処理ができなかった処理が、ループ処理できるようになった。

今回の改良で、大量データの扱いとマッキントッシュへの移植という目的は達成できたといえよう。また、マッキントッシュへの移植から必然的に操作性が向上し、幸いであったといえる。

(2) 課題点

今回のマッキントッシュへの移植によってすべての機能が移植できたわけではなく、一部の機能が省略されている。その中でも「生物種間のDNAの一致率を表示する機能」は必要性が高く、つぎの拡張では最初に行われるべき事項である。

また今回の改良により、大量のデータが扱えるようになったのだが、この量でもミトコンドリアのDNAなどといった特別なDNAまでしか対応できない。さらに大量のデータを扱えるように改良・拡張がなされるべきである。

今後の改良・拡張に当たっては、扱えるメモリの拡大が必要になる。そうすると使用言語を現在のBASICからC言語に移行したほうが無難であるといえよう。その理由は、容易にメモリが確保できること、これからのユーザーズメモリの拡大に対応がしやすいことがあげられる。

そうなると、プログラムの生産性を考えた場合に現行のマッキントッシュでは不適切である。マッキントッシュでのソフトの開発は容易ではなく、専門的な知識をかなり必要とする。マッキントッシュのOSはマルチタスクが主流になりつつある点からしても、今後さらに難解なものとなることが予想される。特別な専門家でない限り「PCでの開発の方が容易である。」ということになる。ただ、マッキントッシュでの開発に比べて、マウスを利用したプログラムの開発は逆に難しくなる。しかし大局的にみれば、操作性よりも実用性の方が必要であるので、PCによるC言語での開発が現時点では最善であろう。

これまでは、大量のデータを扱うことについて述べたが、たとえ大量のデータが扱えなくとも、次の機能を備えればかなり実用性があるエディタになる。

ひとつには、自動一致の機能である。現在手動で行われている編集を自動的に行えるようにすれば、このプログラムを利用する者にとって有益なことになる。

もうひとつは最尤法の導入である。現在は、このエディタなどによって得られた相同な塩基配列の違いを最尤法などで計算する事によって、遺伝的な距離や分岐年代を得る研究が進められている。(木村、根井や長谷川らをはじめとした多くの学者が取り組んでいる。) この最尤法による計算もいっしょに処理できるような拡張が行えればかなり実用性の高いプログラムとなる。

引用文献

Apple Computer Japan, "Macintosh™ System software User's Guide"