User Manual

for

MSDB

A user-friendly program for reporting distribution and building databases of microsatellites from genome sequences

VERSION 2.4.1

Contents

1 Introduction

MSDB (Microsatellite Search and Building Database) is specially designed to offer you a user-friendly interface for finding microsatellite markers and recording their exact position and frequency of occurrence from genomic sequences. MSDB can accept a large number of sequences in GenBank, FASTA and EMBL formats as input and large number of motifs can be searched simultaneously. The program not only can search pure microsatellites but also can search compound and complex microsatellites. The outputs are Microsoft Excel statistics and SQLite database file (http://www.sqlite.org/) which is convenient for analysis and classification of microsatellites. The program is designed to run on Windows and Linux systems as a standalone application with a user-friendly interface.

2 License

Like most software, MSDB is distributed under a license, which means there are certain things that you are legally permitted (and not permitted) to do with MSDB software and source code. MSDB is distributed under a license called the **GNU General Public License** (http://www.gnu.org/licenses/gpl-3.0.html), a very popular license in the open source industry.

3 System Requirements

MSDB is a standalone Perl program and can be run on Windows and Linux operating systems. So you do not have Perl or other modules to be installed on your personal computer. But you should ensure that Microsoft Office Excel 2003 or higher have been installed to open the statistical file. On Linux, instead of Microsoft Office, the free software Open Office can be installed (http://www.openoffice.org/).

Since MSDB relies on R to generate a sliding window plot, you need to install R first. See this page for downloads: http://www.r-project.org/. Add the bin directory of the R package to the PATH environment variable. For example, on Windows systems, I used R-2.6.0-win32.exe to install R, hence I added c:\program files\R\R-2.6.0\bin to the end of the PATH environment variable. Note that this may be different depending on the version of R you've installed. Navigate to the directory to make sure you have it right. If R is in your PATH environment variable, then it should be available from a terminal. Otherwise, MSDB can’t call R. On Linux systems, R may have been installed.

4 Installation

4.1 Installation on windows

1. The preferred way to install MSDB is to directly download it as a single compressed ZIP file from http://msdb.biosv.com/. Then you must use a program, such as WinZip,

to uncompress this ZIP file into a specified directory.

2. Once downloaded and expanded on your hard drive, simply double-click on the Installer, and the appropriate files will be installed on your computer.
3. Try to run MSDB from the windows start menu.

4.2 Installation on Linux

You'll probably want to operate from inside your home directory. If your user is (for example) *username*, your home directory will be */home/username/*. For the rest of this section we will assume you have downloaded your zip file to */home/username/MSDB*. If you do not have a *MSDB* directory, you can create it with the following "mkdir" (make directory) command:

Code: mkdir /home/username/MSDB/

1. Download the package MSDB-2.4.1.tar.gz from http://msdb.biosv.com/.
2. Change to the /home/username/MSDB directory with the “cd” command like so:
   Code: cd /home/username/MSDB/
3. You now need to unzip the zipped file:
   Code: tar –xvfz MSDB-2.4.1.tar.gz
4. Next, you need to go into the new directory, so use the cd command:
   Code: cd MSDB-2.4.1
5. Make sure the file is set to “executable” by running this command:
   Code: chmod +x MSDB;
   chmod +x SWP;
   chmod +x SWR
6. Run the program like this:
   Code: ./MSDB
   Or double click the “MSDB” directly to run the program.

4.3 Compiling From Source Code

4.3.1 Perl Requirements

To test the source code, you should have activeperl5.10 or later to be installed on your computer. When activeperl is being installed, it should be installed as an Administrator and it must be made available to all users.

You can obtain the free express editions of activeperl from

http://www.activestate.com/activeperl

4.3.2 Dependencies

The dependencies required by MSDB source code are perl modules. All of the following

perl modules must be installed for MSDB script to work properly:

1. **Tkx:** provides yet another Tk interface for Perl. Tk is a GUI toolkit tied to the Tcl language, and Tkx provides a bridge to Tcl that allows Tk based applications to be written in Perl.
2. **File::Basename:** allow you to parse file paths into their directory, filename and suffix.
3. **File::Find:** search through directory trees doing work on each file found similar to the Unix find command.
4. **File::Copy:** provides two basic functions, copy and move, which are useful for getting the contents of a file from one place to another.
5. **DBI:** is a database access module for the Perl programming language. It defines a set of methods, variables, and conventions that provide a consistent database interface, independent of the actual database being used.
6. **ActiveState::Browser:** provides an interface to make a web browser pop up showing some URL or file.
7. **Spreadsheet::WriteExcel:** can be used to create a cross-platform Excel binary file. Multiple worksheets can be added to a workbook and formatting can be applied to cells. Text, numbers, formulas, hyperlinks, images and charts can be written to the cells.

More information on installing Perl modules can be found on CPAN (http://cpan.org). However, many of them will be installed as standard with Perl5.10 (if so, man or Perldoc should provide information).

4.3.3 Compiler

To compile from source code you will need PerlApp compiler for Windows. PerlApp can compile source code into executable file for Windows and Linux operating systems. The PerlApp compiler was contained in the Perl Dev Kit, which can be downloaded from http://www.activestate.com/perl-dev-kit.

5 Using MSDB

5.1 Overview

MSDB contains three programs: the MSDB main program used for finding and statistics of microsatellites; SWR (search within results) used for retrieving data from databases generated by the main program according to the end user's requirements and exporting to design primer; SWP (sliding window plot) used for automatically generating a sliding window plot of density or frequency of microsatellites versus sequence or chromosome position for sliding window analysis. MSDB has two search

modes: perfect search mode used to search pure microsatellites and imperfect search mode used to search all types of microsatellites, which allows analysis of compound and complex microsatellites. This program can accept FASTA, GenBank, EMBL and plain sequence files with any extensions. How to set the parameters for program running and more detail is given in this manual.

5.2 Main Program Windows

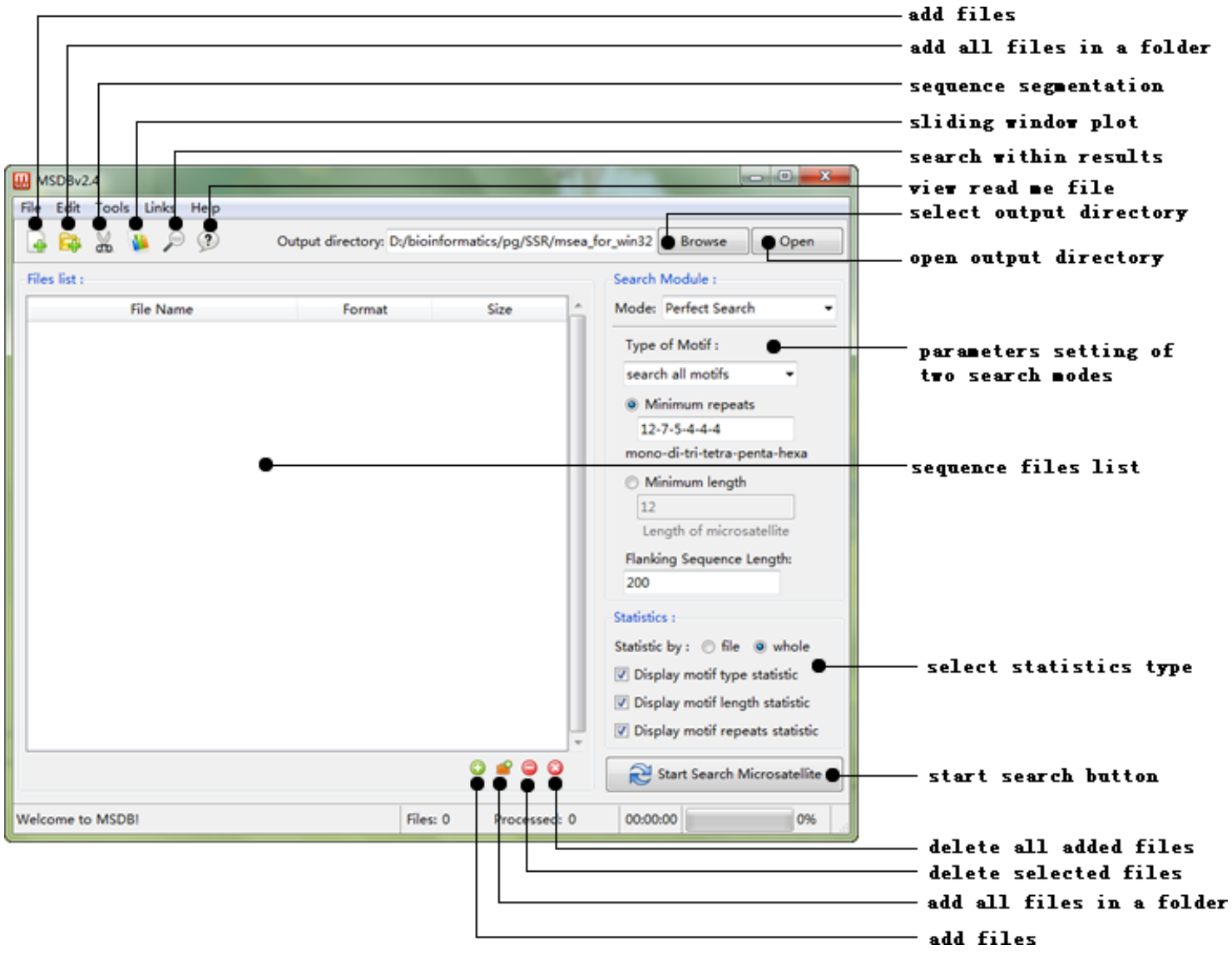

Fig. 1: The standard windows of main program of MSDB

5.3 Input and output files

MSDB can only process FASTA, GenBank, EMBL and plain files with one sequence; therefore, if a sequence file contains multiple sequences, it should be divided into multiple files such that each file contains one sequence.

5.3.1 Input files

Input file types are automatically detected, if the file format cannot be determined, it will be

considered as a plain sequence file. Allowed file types are GenBank, EMBL, FASTA, and plain sequence file with any extension.

Note that you should ensure the FASTA files start with “>” in first line, the GenBank files begin with word “LOCUS” in the first line and the EMBL files begin with the word “ID” in the first line. The number and space in the sequence file will be automatically deleted.

If the sequence file has more than one sequence, you should split it into multiple files before adding it to the program. The sequence segment window was shown in Figure 2. Note that multiple sequences were separated by “//” in GenBank and EMBL files.

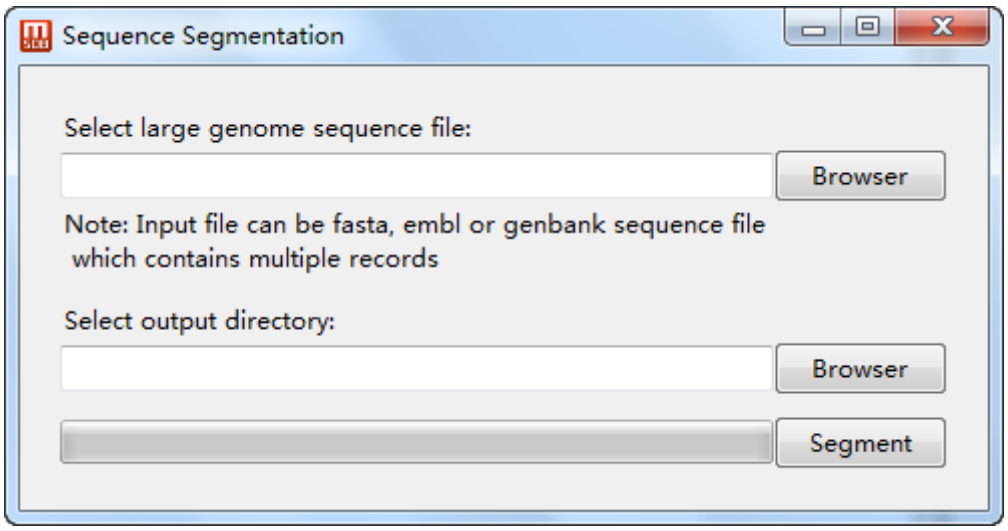

Fig. 2: The window of sequence segmentation

5.3.2 Add and delete files

There are two add file commands in the File menu: add files and add folder. One file or multiple files (select multiple files using the Ctrl or Shift keys) will be added into the program by using add files command. Using add folder command, a folder should be selected, and all the files in this folder will be added into the program. Files can also be added by clicking the buttons on the interface of program (FIG.1). You can also quickly drag files to files lists to add files.

There two delete file commands in the Edit menu: delete selected files and delete all files. You can select multiple files using the Ctrl or Shift keys from the files list. MSDB also provides two buttons on the interface of program to delete files (FIG.1).

5.3.3 Output files

Before the running of the program, an output directory must be specified, otherwise, nothing will be obtained when tasks complete. You will be allowed to select a directory by clicking “Browse” button as shown in Figure 1. During the running, the program will generate a database file and Excel statistical file. And the formats of these output files’

name were shown in Table 1.

Table 1. The formats of output file name.

File Type	Format
database	database_ + time + .db
statistics	statistics_ + time + .xls *(all files)*
	sequence name + statistics_ + time + .xls *(single file)*

5.4 Search Modes

Microsatellites can be grouped into six categories: pure microsatellites or perfect microsatellites consist of identical repeats; interrupted pure microsatellites consist of two or more individual pure microsatellites with the same motif; compound microsatellites consist of two adjacent pure microsatellites with different motifs; Interrupted compound microsatellites consist of two repetitive sequences of compound microsatellites interrupted by a short non-repetitive sequence; complex microsatellites contain several different perfect repetitive sequences; interrupted complex microsatellites contain several different perfect repetitive sequences interrupted by non-repetitive sequences.

Table2. Examples of the six classes of microsatellites

Class	Sequence
Pure	-$(AC)_{14}$-
Interrupted pure	-TA-$(CA)_4$-TA-$(CA)_7$-
Compound	-$(CT)_{22}$-$(CA)_6$-
Interrupted compound	-$(AC)_{14}$-AGAA-$(AG)_{12}$-
Complex	-$(TG)_{22}$-$(AG_{10}$-$(ACAG)_5$-
Interrupted complex	-$(CT)_{28}$-CGCACGCACG-$(CA)_{13}$-GACAG-$(AC)_9$-

MSDB has two main search modes: perfect search and imperfect search. Perfect search mode offers three different motif search methods. In this mode, you will be allowed to search perfect microsatellites. Imperfect search mode allows you to search compound and complex microsatellites.

5.4.1 Perfect Search Mode

This mode offers three different types of motifs search methods: search all motifs, search motifs of certain nucleotide number and search custom motifs. Search all motifs means searching all the potential motifs in genomic sequence. Search motifs of a certain nucleotide number means searching the motifs with a specified length. Custom motifs only search the motifs specified by user input.

5.4.1.1 Search All Motifs

In this search method, you can search microsatellite according to minimum repeats of microsatellite or minimum length of microsatellite. The former requires an input string of the form mono-di-tri-tetra-penta-hexa as shown in Figure 5. For example, the input string 12-7-5-4-4-4 stats that a mononucleotide microsatellite must have at least 12 repeats, a dinucleotide microsatellite must have at least 7 repeats and so on. The later requires an input minimum length number. The length of microsatellite found in sequence must be greater than the minimum length number.

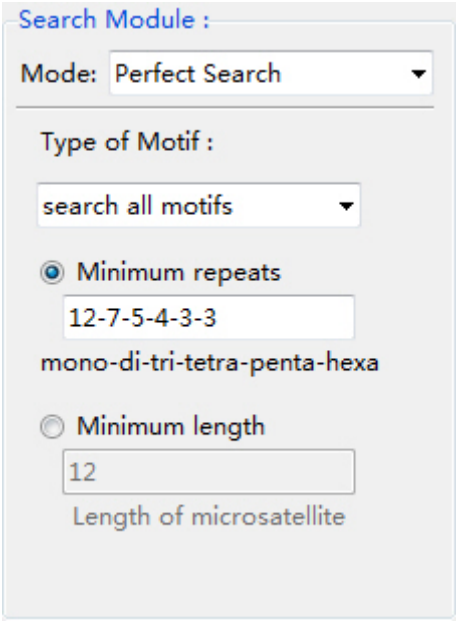

Fig. 3: Adjust the settings of perfect search with search all motifs

5.4.1.2 Search motifs of certain nucleotide number

Microsatellites are divided into six different types: mono-, di- , tri-, tetra-, penta- and hexanucleotide microsatellites according to the length of motif. So in this search method, you will be allowed to select one type from six to search microsatellites as shown in Figure 4. When you do not want to search all motif microsatellites, this is the best way for searching microsatellites and saving your time.

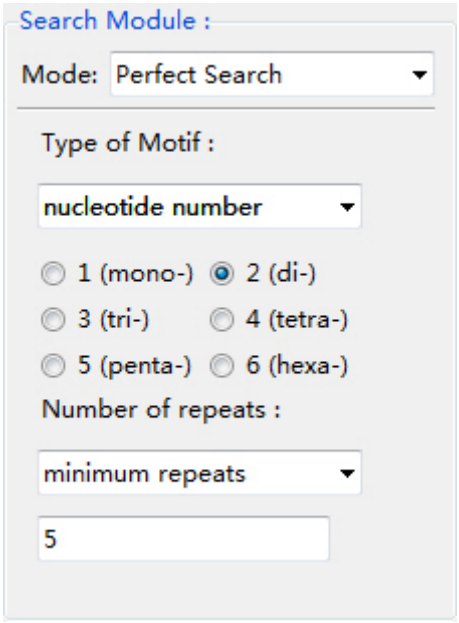

Fig. 4: Adjust the setting of search motifs of certain nucleotide number

Another important parameter for this search is the number of repeats. Three types of numbers of repeats can be selected to search microsatellites as shown in Figure 5. The minimum repeats allows to output microsatellites whose repeats number is greater than the minimum repeats user inputted (Fig. 5a). The precise repeats allows to output microsatellites whose repeats number equals to the repeats number user inputted (Fig. 5b). And interval repeats requires two parameters: minimum number of repeats and maximum number of repeats and allows to output microsatellites whose repeats number is between minimum repeats and maximum repeats (Fig. 5c).

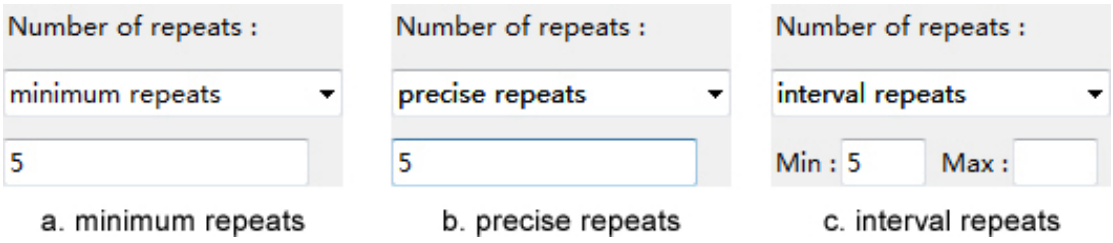

Fig. 5: Adjust the setting of repeats number

5.4.1.3 Custom Motifs

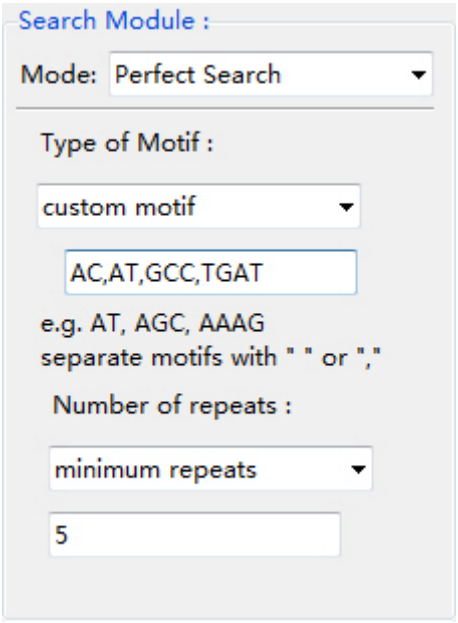

Fig. 6: custom motifs

Sometimes you only want to search some specific motifs rather than search all motifs. You can input all motifs of any length and separate motifs with commas or space (FIG.8). The setting of the number of repeats is the same as above, please see 5.4.1.2.

5.4.2 Imperfect Search Mode

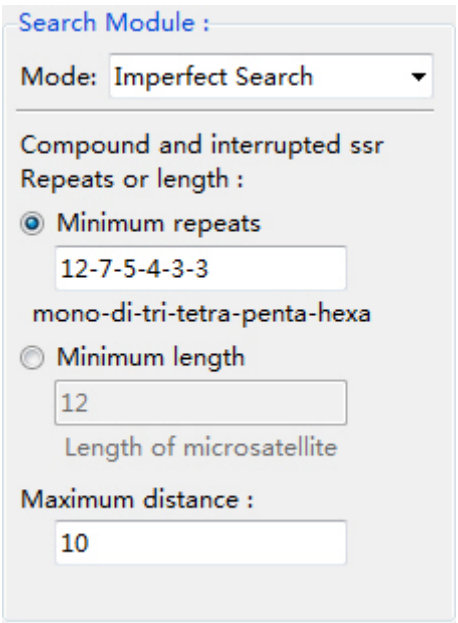

Fig. 7: Adjust settings for imperfect search mode

This search mode allows you to search pure, compound and complex microsatellites. In imperfect search mode, two parameters, minimum number of repeats and maximum distance (dMAX), must be set as shown in Figure 7. DMAX is the maximum distance allowed between two adjacent microsatellites. In this search mode, all microsatellites

found will be classified into six groups (pure, interrupted pure, compound, interrupted compound, complex and interrupted complex) according to the motifs and distance between two adjacent microsatellites. The setting of minimum repeats or length of microsatellite is the same as for the search all motifs of perfect search model, please see 5.4.1.1.

5.5 Microsatellite statistic

Microsatellite statistics is an important function of MSDB and the options of statistics were shown in Figure 8. First you should select a statistical method. The option “file” means the program will generate a statistical Excel file for each sequence file and the option “whole” means the program will generate one statistical Excel file for all sequence files as a whole.

a.statistics for perfect search mode **b.statistics for imperfect search mode**

Fig. 8: the options of statistics

The perfect search mode and imperfect search mode have different statistical options. These statistical options contain 5 common columns: total counts, total length (bp), average length (bp), frequency (loci/Mb) and density (bp/Mb). The SSR frequency was measured as the average number of SSRs per megabase:

$$F = \frac{N}{G} \times 1000000$$

Where *F* was the frequency of SSR, *N* was the total number of SSR, and G is the size of the whole genome sequence. The SSR density was measured by the total SSR length per megabase:

$$D = \frac{L}{G} \times 1000000$$

Where *D* was the density of the SSR, *L* was the total length of SSR.

5.5.1 The motif length statistic

The motif length statistic also contains nucleotide column. In this statistic, pure microsatellite was divided into six types: mono-, di-, tri-, tetra-, penta- and hexanucleotide microsatellite, according to the nucleotide number of motif of microsatellite. An example was shown in Figure 9.

	A	B	C	D	E	F
1	Nucleotide	Total Counts	Total Length(bp)	Average Length(bp)	Frequency(loci/Mb)	Density(bp/Mb)
2	mononucleotide	47	685	14.57	193.53	2820.602
3	dinucleotide	28	576	20.57	115.29	2371.776
4	trinucleotide	5	90	18	20.59	370.59
5	tetranucleotide	20	432	21.6	82.35	1778.832
6	pentanucleotide	4	100	25	16.47	411.767

Fig. 9: an example for the motif length statistic

5.5.2 The motif type statistic

The motif type statistic also contains motif (=type) and motif length columns. In this statistic, it will record statistical information according to the motif of the microsatellite and similar motifs will be grouped together. For instance, the microsatellite motifs “AC” and “CA” become identical during search, the counts of motif “CA” microsatellites will be added to the “AC” microsatellites. An example was shown in Figure 10.

	A	B	C	D	E	F	G
1	Motif	Motif Length	Total Counts	Total Length(bp)	Average Length(bp)	Frequency(loci/Mb)	Density(bp/Mb)
2	A	1	31	446	14.39	127.65	1836.48
3	AAAC	4	2	36	18	8.24	148.24
4	AAAT	4	8	168	21	32.94	691.77
5	AAG	3	2	36	18	8.24	148.24
6	AC	2	8	202	25.25	32.94	831.77
7	AG	2	7	116	16.57	28.82	477.65
8	AGG	3	2	39	19.5	8.24	160.59
9	ATCA	4	1	16	16	4.12	65.88
10	ATTT	4	2	52	26	8.24	214.12
11	CACCT	5	1	20	20	4.12	82.35
12	CT	2	6	108	18	24.71	444.71

Fig.10: an example for the motif type statistic

5.5.3 The motif repeats statistic

The motif repeats statistic is a more detailed statistic than the motif type statistic. This statistic also contains motif and repeats columns and statistical information will be recorded according to the motif and the repeats number of the microsatellites. An example was shown in Figure11.

	A	B	C	D	E	F	G
1	Motif	Repeats	Total Counts	Total Length(bp)	Average Length(bp)	Frequency(loci/Mb)	Density(bp/Mb)
2	A	12	5	60	12	20.59	247.06
3		13	12	156	13	49.41	642.36
4		14	1	14	14	4.12	57.65
5		15	4	60	15	16.47	247.06
6		16	5	80	16	20.59	329.41
7		17	2	34	17	8.24	140
8		19	1	19	19	4.12	78.24
9		23	1	23	23	4.12	94.71
10	AAAC	4	1	16	16	4.12	65.88
11	AAAT	4	2	32	16	8.24	131.77
12		5	2	40	20	8.24	164.71
13	AAG	5	1	15	15	4.12	61.76
14	AC	8	2	32	16	8.24	131.77
15		14	1	28	28	4.12	115.29
16		15	2	60	30	8.24	247.06

Fig. 11: an example for the motif repeats statistic

5.5.4 The SSR type statistic

The SSR type statistic also contains type column. In this statistic, microsatellites have been group into six classes: pure microsatellites (p), interrupted pure microsatellites (ip), compound microsatellites (cd), interrupted compound microsatellites (icd), complex microsatellites (cx) and interrupted complex microsatellites (icx). An example was shown in Figure 12.

	A	B	C	D	E	F
1	Type	Total Counts	Total Length(bp)	Average Length(bp)	Frequency(loci/Mb)	Density(bp/Mb)
2	cd	4172	276031	66.16	25.03	1656.35
3	cx	628	73356	116.81	3.77	440.18
4	icd	2649	165340	62.42	15.9	992.14
5	icx	1746	200915	115.07	10.48	1205.61
6	ip	1426	78400	54.98	8.56	470.45
7	p	59243	1520106	25.66	355.49	9121.53

Fig. 12: an example for SSR type statistic

5.5.5 The pure SSR statistic

The pure SSR statistic also contains motif and motif length columns. This statistic will be made only for pure microsatellite (pure and interrupted pure) and is the same as the motif type statistic (Fig. 10).

5.5.6 The compound SSR statistic

The compound SSR statistic also contains SSR Couple column. This statistic will be made only for compound microsatellite (compound and interrupted compound). An example was shown in Figure 13.

	A	B	C	D	E	F
1	SSR Couple	Total Counts	Total Length(bp)	Average Length(bp)	Frequency(loci/Mb)	Density(bp/Mb)
2	A-AAAC	3	116	38.67	0.02	0.7
3	A-AAACA	1	48	48	0.01	0.29
4	A-AAACC	1	52	52	0.01	0.31
5	A-AAAGA	4	286	71.5	0.02	1.72
6	A-AAC	2	70	35	0.01	0.42
7	A-AACA	1	33	33	0.01	0.2
8	A-AACAA	1	47	47	0.01	0.28
9	A-AAGA	1	64	64	0.01	0.38
10	A-AAGG	2	139	69.5	0.01	0.83
11	A-AATA	1	31	31	0.01	0.19
12	A-AC	3	193	64.33	0.02	1.16
13	A-ACAA	3	105	35	0.02	0.63
14	A-AG	3	154	51.33	0.02	0.92
15	A-AGAC	1	45	45	0.01	0.27

Fig. 13: an example for the compound SSR statistic

5.5.7 The complex SSR statistic

The complex SSR statistic also contains SSR Cluster column. This statistic will be made only for complex microsatellite (complex and interrupted complex). An example was

shown in Figure 14.

	A	B	C	D	E	F
1	SSR Cluster	Total Counts	Total Length(bp)	Average Length(bp)	Frequency(loci/Mb)	Density(bp/Mb)
2	A-A-AAC-A	1	78	78	0.01	0.47
3	A-A-GA	1	41	41	0.01	0.25
4	A-A-GAAA	1	60	60	0.01	0.36
5	A-AAAGAA-AAGAA	1	86	86	0.01	0.52
6	A-AAG-GAA	1	194	194	0.01	1.16
7	AAGA-AG-GAAG-AAA	5	781	156.2	0.03	4.69
8	A-AAGAA-AGGAA	1	110	110	0.01	0.66
9	A-AAGG-AAAG	1	95	95	0.01	0.57
10	A-AG-TC	1	67	67	0.01	0.4
11	GGAA-AGGGGA-AGG	1	392	392	0.01	2.35
12	-CAA-AAACC-ACC-AA	1	116	116	0.01	0.7
13	A-CAA-CAA	1	58	58	0.01	0.35
14	A-CAAA-AAAC	1	53	53	0.01	0.32
15	A-CAAAA-ACAAA	1	63	63	0.01	0.38

Fig. 14: an example for the complex SSR statistic

5.5.8 The sequence file information statistic

Except for above types of statistic, the sequence file information statistic will be automatically made in perfect search mode and imperfect mode. This statistic contains seven columns: sequence name, sequence length, total SSR counts, total SSR length, average length, frequency and density. An example was show in Figure 15.

	A	B	C	D	E	F	G
1	Sequence Name	Sequence Length(bp)	Total SSR Counts	Total SSR Length(bp)	Average Length(bp)	Frequency(loci/Mb)	Density(bp/Mb)
2	seq3	647	1	14	14	1545.6	21638.33
3	seq11	22959	5	75	15	217.78	3266.69
4	seq7	87055	30	535	17.83	344.61	6145.54
5	seq6	3450	1	12	12	289.86	3478.26
6	seq9	52165	25	456	18.24	479.25	8741.49
7	seq4	27661	11	207	18.82	397.67	7483.46
8	seq5	4787	3	49	16.33	626.7	10236.06
9	seq2	24662	13	262	20.15	527.13	10623.63
10	seq8	17876	14	245	17.5	783.17	13705.53
11	seq10	1594	1	28	28	627.35	17565.87

Fig. 15: an example for the sequence file information statistic

5.6 Database for microsatellites

MSDB will automatically generate a SQLite database file to save microsatellite information. There are two tables, “ssr” and “file”, within the database used to store data record. The table “ssr” used to store information of each microsatellite has 12 fields: uid, motif, type, complexity, repeats, length, seq, start, end, left, right, source. The description of each field was shown in Table 3.

Table 3. The description of each field in table "ssr" of database

Field	Description
uid	An automatically increased number as the identifier of microsatellite
motif	The motif for pure microsatellite or the SSR couple for compound microsatellite or the SSR cluster for complex microsatellite
type	The type of microsatellite (value: p, ip, cd, icd, cx or icx)
complexity	The number of individual SSRs within a microsatellite (pure microsatellite = 1, compound microsatellite = 2 and complex microsatellite > 2)
repeats	The number of repeats of motif (the sum of the repeats of all motifs for interrupted pure, compound and complex microsatellite)
length	The length of microsatellite
seq	The sequence of microsatellite (example: (AC)7-GTCC-(AG)8)
start	The start site of microsatellite
end	The end site of microsatellite
left	The left flanking sequence of microsatellite
right	The right flanking sequence of microsatellite
source	The name of sequence file where microsatellite was found

And the table "file" used to store information of each sequence file has four fields: filename, size, count, length. The description of each field was shown in Table 4.

Table 4. The description of each field in table "file" of database

Field	description
filename	The name of sequence file
size	The length of sequence in sequence file
count	Total counts of microsatellites found in sequence file
length	Total length of microsatellites found in sequence file

Table "ssr" and "file" build relationship with each other by fields source and filename. Each microsatellite has its own features: motif, repeats, type, complexity, length, location, flanking sequence and source. The program will insert these features as a record into table "ssr" of database. And for each sequence file, its file name, total length of sequence, total counts of microsatellites will be inserted into table "file" of database.

The generated SQLite database can be used in subsequent analyses. The microsatellites in database can be exported into Excel or TXT file according end users' requirements. More importantly, you can get more detailed statistics using generated SQLite database by executing SQL statement. The more detailed information about how to use SQLite

database will be stated below.

5.7 Example

In this section, we use the X chromosome of mouse as an example. The data file for this example is available online and can be downloaded from website (http://hgdownload.cse.ucsc.edu/goldenPath/mm9/chromosomes/). The X chromosome of mouse has a length of 166.65Mb. We found 83913 microsatellite loci on chromosome X of the mouse by using perfect search mode with default parameters setting. This whole process has taken 7min 8s. The results were shown in Table 5.

Table 5. Summary of microsatellite loci on chromosome X of mouse

Nucleotide	SSR Counts	SSR Length (bp)	Average Length (bp)	Frequency (loci/Mb)	Density (bp/Mb)
mononucleotide	20886	342414	16.39	125.33	2054.686
dinucleotide	31255	974436	31.18	187.55	5847.19
trinucleotide	7065	194721	27.56	42.39	1168.441
tetranucleotide	18337	530908	28.95	110.03	3185.761
pentanucleotide	4611	163340	35.42	27.67	980.136
hexanucleotide	1759	82314	46.8	10.56	493.933
Total	83913	2288133	27.27	503.53	13730.15

The detailed introduction on how to use generated SQLite database will be shown in below.

6 Search within results

In addition to the main program functions, MSDB also contains SWR (search within results) program which can be used to search microsatellite within the database generated by the main program. This program allows you to export microsatellites into an Excel file or export into a file for primer3 to design primers.

6.1 Window of SWR

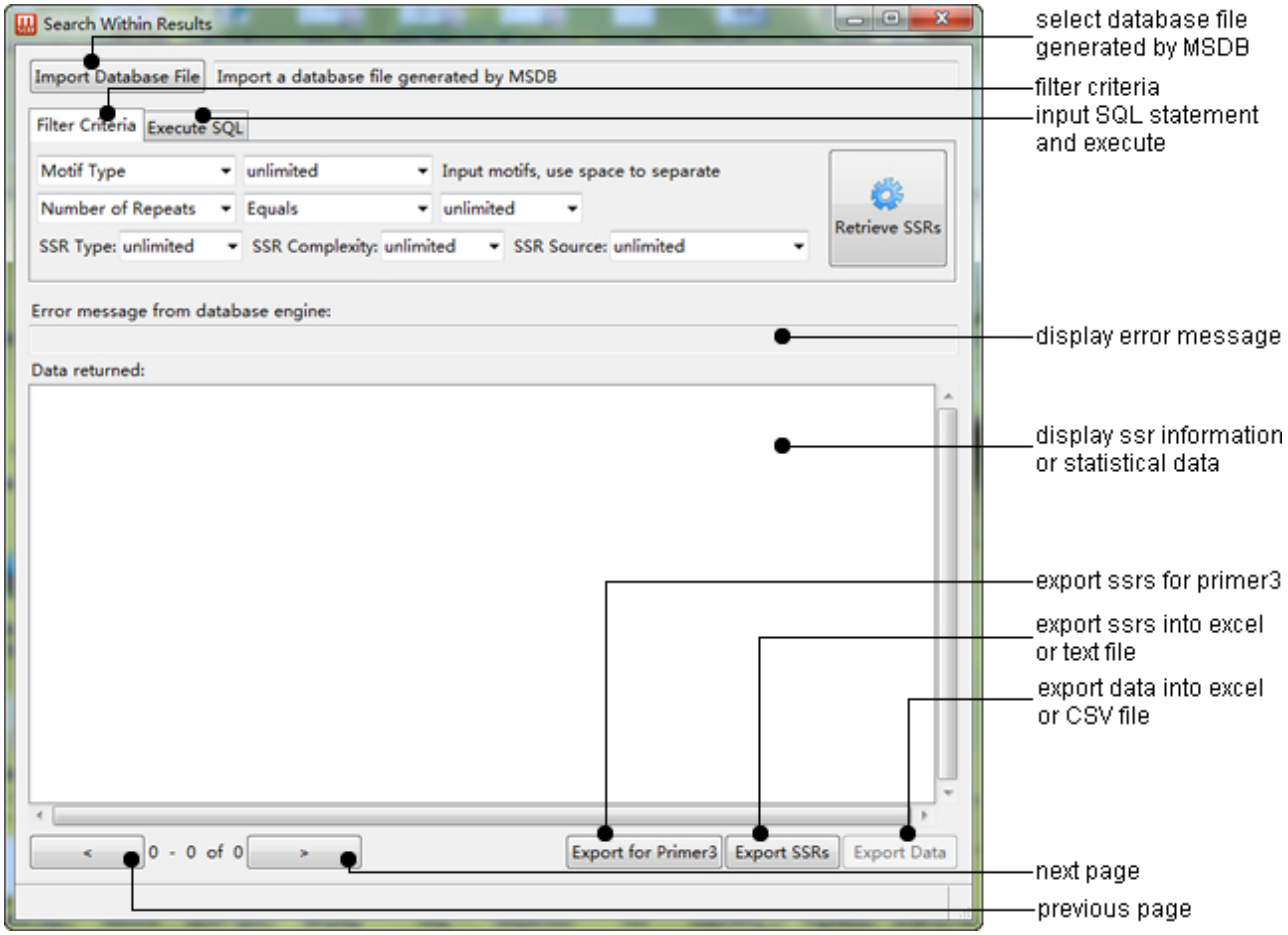

Fig. 16: the window of SWR

6.2 Export SSRs

When the main program has completed its tasks, the generated database can be used to export microsatellites by SWR. You can set different filter criteria to select microsatellites and then export the selected microsatellites. In addition to the filter criteria setting, SWR also provides advanced feature. SWR supports the creation of SQL statements to select microsatellites or count the number of microsatellites.

6.2.1 Export SSRs to Excel or TXT file

SWR can help you to export SSRs to Excel or TXT file from the SQLite database file. The window of exporting SSRs to Excel file was shown in Figure 17.

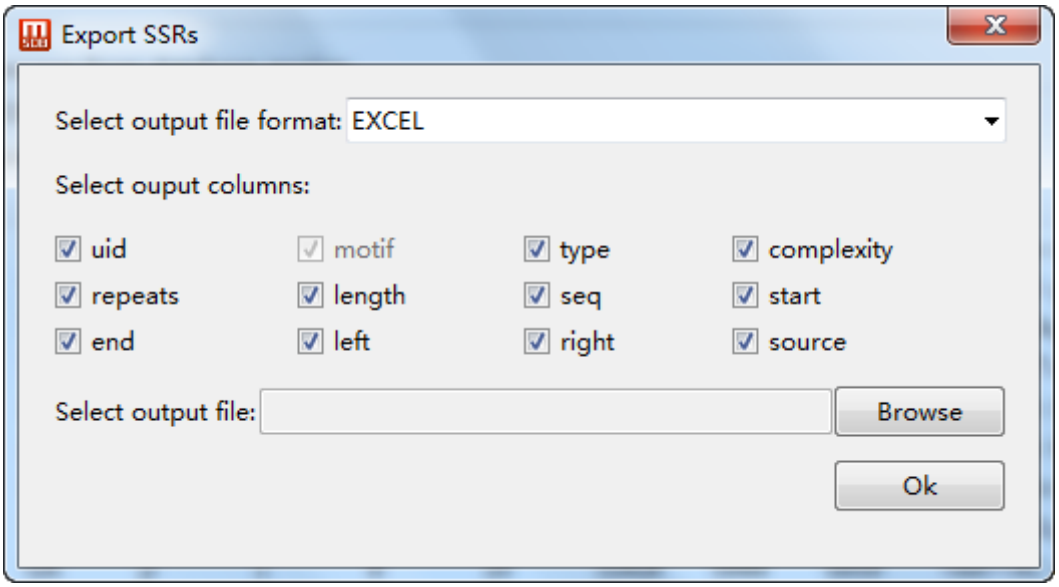

Fig.17: the window of exporting SSRs to Excel file

Twelve columns have been listed in Figure 17. The meaning of each column was described in Table 3. You can specify any columns to output according to your requirements.

The window of exporting SSRs to TXT file was shown in Figure 18. In which case, the program allows you to edit the output format by using the tags which are listed in Figure 18. The meaning of each tag is the same as the column above. When outputting the TXT file, the tags in format will be replaced by relevant content.

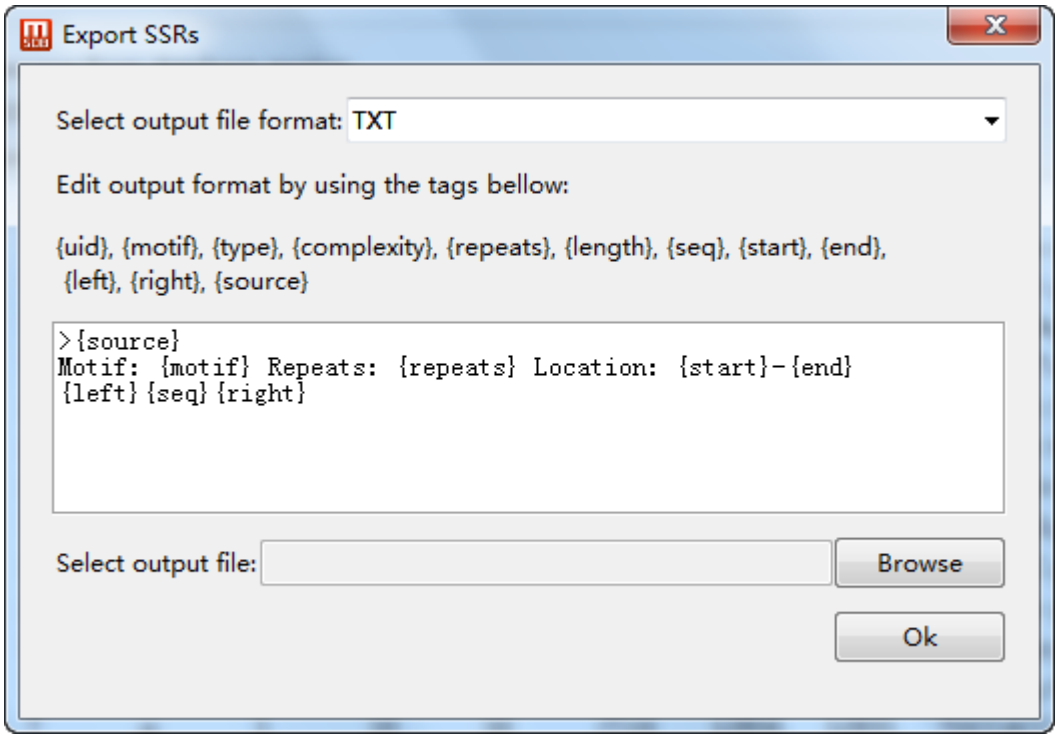

Fig.18: the window of exporting SSRs to TXT file

6.2.2 Export SSRs for Primer3

SWR also can help you to export SSRs from SQLite database to generate primer3 formatted input file. The window of exporting SSRs for primer3 was shown in Figure 19.

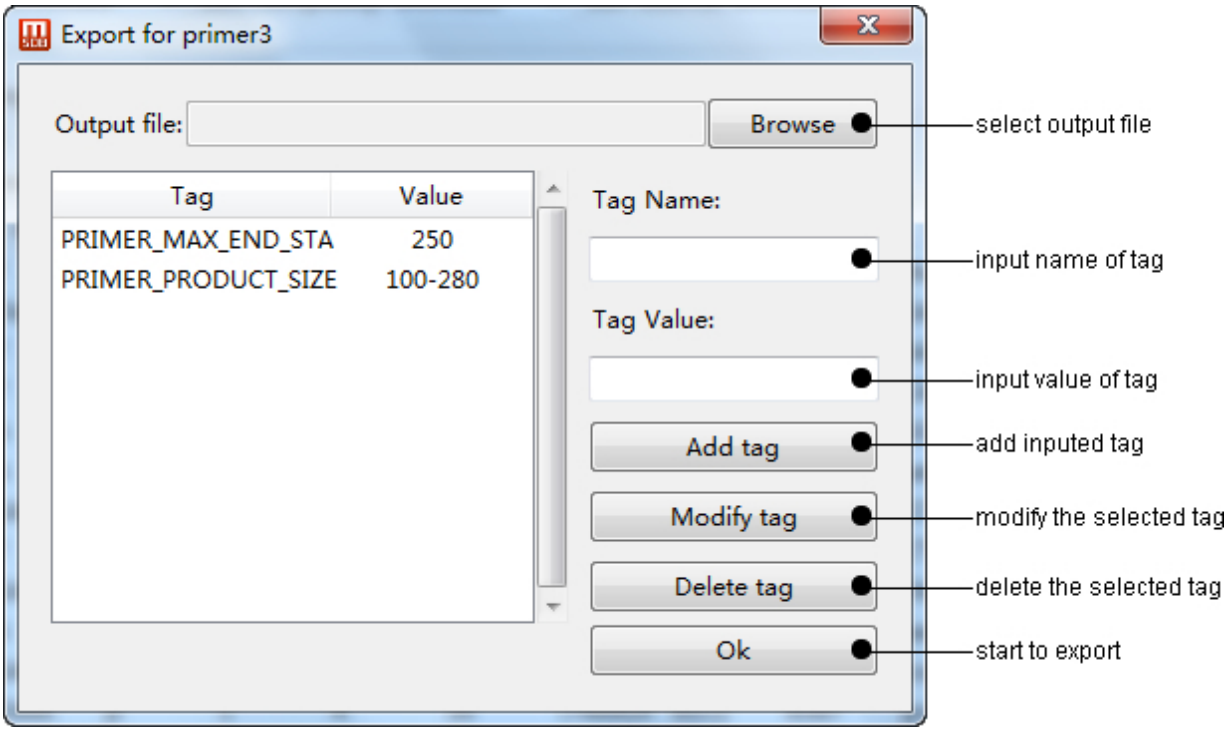

Fig.19: the window of exporting SSRs for primer3

The input file of primer3 consists of a sequence of records. A record consists of a sequence of (TAG, VALUE) pairs, each terminated by a newline character (\n). A record is terminated by '=' appearing by itself on a line. The tags SEQUENCE_ID, SEQUENCE_TEMPLATE and SEQUENCE_TARGET will be added automatically by the program. You are also allowed to add other tags as shown in Figure19.

6.3 Examples

An example for SWR was shown in Figure 20, when importing the database file of the X chromosome of the mouse generated by MSDB.

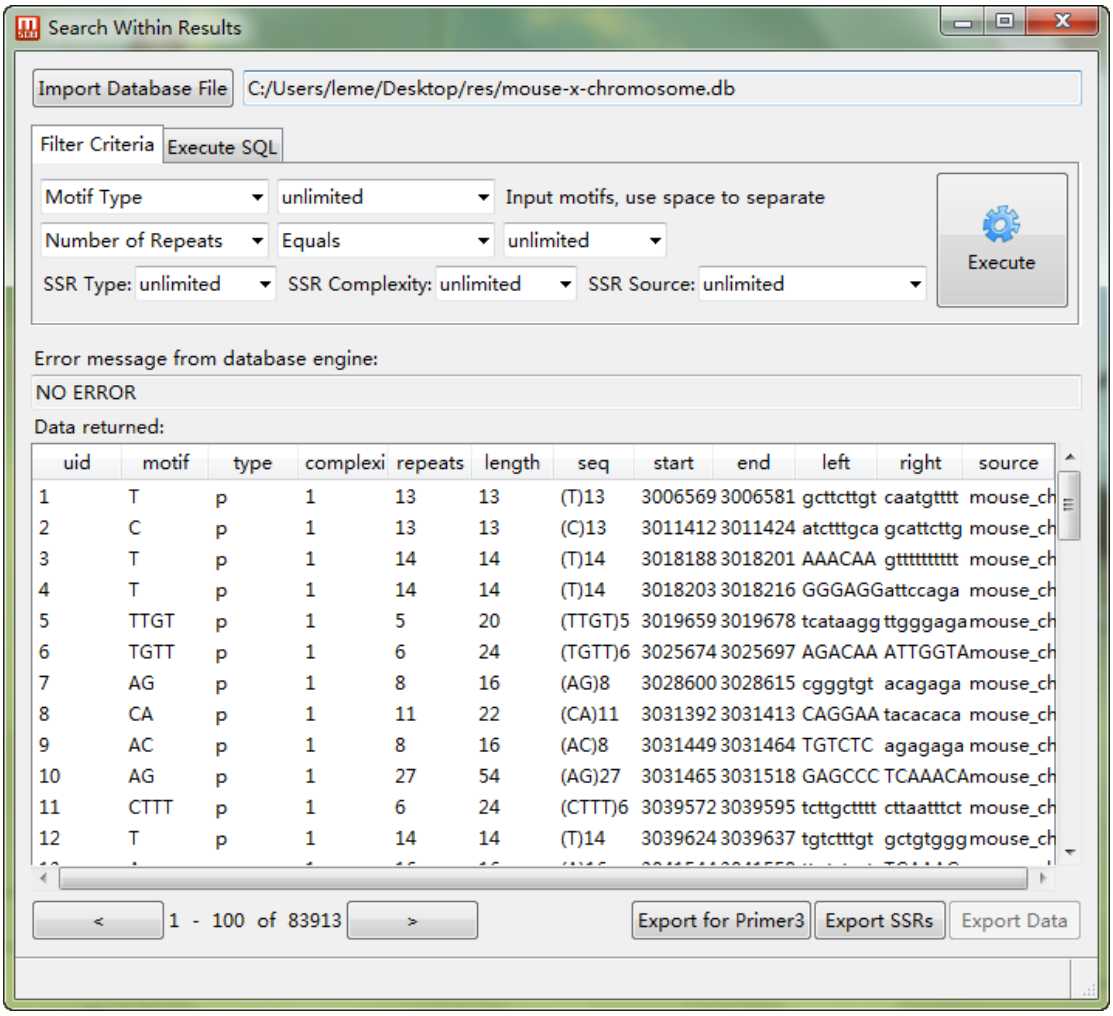

Fig. 20: an example for SWR

An example for output for primer3 was shown in Figure 21.

```
SEQUENCE_ID=74
SEQUENCE_TEMPLATE=aaacacaaaaaagtaaatagatatgtcttagataactgagagccctatgtgtttaacaagcaattggggcataccag
ggcttacaaataaatgctctatatgatccatgtttcaccttctactttccttcattgttatatcctcccccatcacataaaaaaatatcacacac
acacacacacagacagacagatagacagACACACACACACACACACACACAgagagagaaagagagagagagagagagagagagtcaaacaacat
ttaattcctgtgtagcaacaaggacatcttgcatgcagttcagtctggttaaggaatgggagaagcaaaatgaggctttagtcaatgatagagga
aggaagggatttagatgggtcttgaggctttgaagatgaagtattaaccaggctgcttgaaa
SEQUENCE_TARGET=201,24
PRIMER_MAX_END_STABILITY=250
PRIMER_PRODUCT_SIZE_RANGE=100-280
=
SEQUENCE_ID=75
SEQUENCE_TEMPLATE=taactgagagccctatgtgtttaacaagcaattggggcataccagggcttacaaataaatgctctatatgatccatg
tttcaccttctactttccttcattgttatatcctcccccatcacataaaaaaatatcacacacacacacacacagacagacagatagacagacac
acacacacacacacacacacagagagaaAGAGAGAGAGAGAGAGAGAGAGtcaaacaacatttaattcctgtgtagcaacaaggacatcttgc
atgcagttcagtctggttaaggaatgggagaagcaaaatgaggctttagtcaatgatagaggaaggaagggatttagatgggtcttgaggctttg
aagatgaagtattaaccaggctgcttgaaagatcactgtgtcttccctttgatgaccccaa
SEQUENCE_TARGET=201,24
PRIMER_MAX_END_STABILITY=250
PRIMER_PRODUCT_SIZE_RANGE=100-280
=
```

Fig. 21: an example for exporting primer3 input file

An example for exporting to excel file was shown in Figure 19.

	A	B	C	D	E	F	G	H
1	motif	repeats	length	seq	start	end	left	right
2	GAAA	18	72	GAAAGAA	3100010	3100081	caagaccaa	aaagaccaa
3	TTCT	18	72	TTCTTTCT	5078406	5078477	ctatattgtac	ctctctctctt
4	AAGA	18	72	AAGAAAG	6784172	6784243	TTATACCC	aaggaagga
5	CTTT	18	72	CTTTCTTT	7609160	7609231	AGCCACC	ttctttcttcct
6	TCTT	18	72	TCTTTCTT	7792541	7792612	ttttctgcattt	tctctttttttcg
7	CTTC	18	72	CTTCCTTC	8571670	8571741	TTCAGTTA	ctctctctctc
8	AGAA	18	72	AGAAAGA	11051580	11051651	AAGGAAA	agacagaaC
9	GGAA	18	72	GGAAGGA	11338128	11338199	attgctataca	gCTTAAAA
10	CTTT	18	72	CTTTCTTT	14621435	14621506	CATCCTG	cCATATTT
11	TTCT	18	72	TTCTTTCT	15224497	15224568	Ctatcctagt	ttttctttctttc
12	TTCC	18	72	TTCCTTCC	16496590	16496661	taagtgctga	ttgtttctttctt

Fig. 22: an example for exporting to excel file

6.4 how to use SQL to manipulate SQLite database

SWR also allows the creation of SQL statements to select microsatellites or count the number of microsatellites. Before using “execute SQL”, you should know the structure of database,(please see 5.6). And you can go to the website (www.w3schools.com/sql/default.asp) to learn how to use SQL language to select data within a database. The generated SQLite database also can be opened by SQLiteMan (http://sourceforge.net/projects/sqliteman/), SQLite Database Browser (http://sqlitebrowser.sourceforge.net/) or SQLiteSpy (http://www.yunqa.de/delphi/doku.php/products/sqlitespy/index).

How to input a SQL statement and execute SQL was shown in Figure 23.

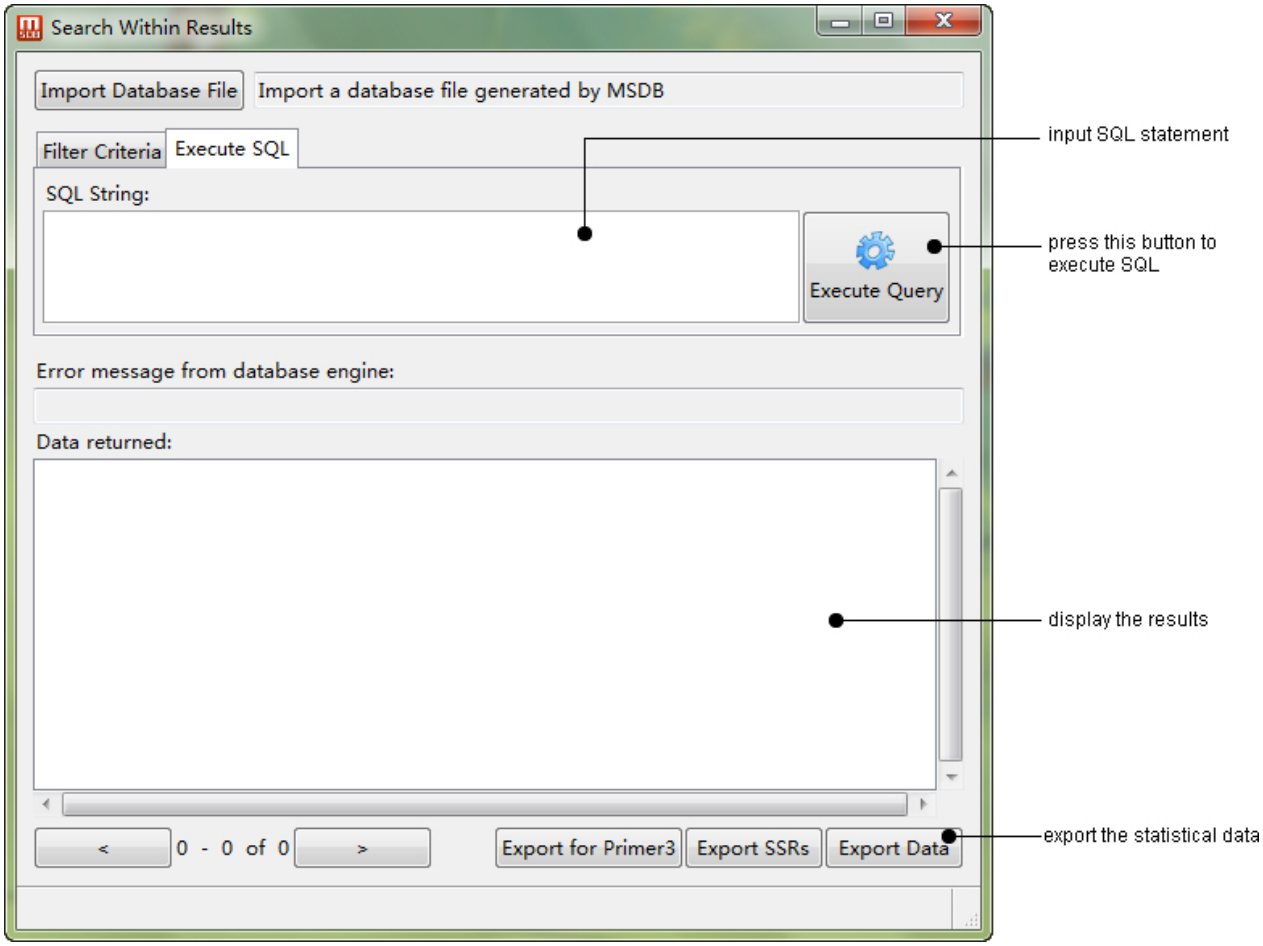

Fig. 23: the window of executing SQL statement

6.4.1 Select SSRs using SQL statement

First, you should import the SQLite database generated by MSDB's main program. We use the microsatellite database of the X chromosome of the mouse generated above as an example. If you want to select all the columns from the "ssr" table, you can use the following SELECT statement:

```
SELECT * FROM ssr
```

When execute this SQL statement, all microsatellites and their whole information in the database will be listed. The result will look like this:

uid	motif	type	complexi	repeats	length	seq	start	end	left	right	source
1	T	p	1	13	13	(T)13	3006569	3006581	gcttcttgta	caatgttttc	mouse_ch
2	C	p	1	13	13	(C)13	3011412	3011424	atctttgcat	gcattcttg	mouse_ch
3	T	p	1	14	14	(T)14	3018188	3018201	AAACAAT	gttttttttttt	mouse_ch
4	T	p	1	14	14	(T)14	3018203	3018216	GGGAGG	attccagat	mouse_ch
5	TTGT	p	1	5	20	(TTGT)5	3019659	3019678	tcataagg	ttgggaga	mouse_ch
6	TGTT	p	1	6	24	(TGTT)6	3025674	3025697	AGACAAT	ATTGGTA	mouse_ch
7	AG	p	1	8	16	(AG)8	3028600	3028615	cgggtgtg	acagagag	mouse_ch
8	CA	p	1	11	22	(CA)11	3031392	3031413	CAGGAAG	tacacacag	mouse_ch
9	AC	p	1	8	16	(AC)8	3031449	3031464	TGTCTCA	agagaga	mouse_ch
10	AG	p	1	27	54	(AG)27	3031465	3031518	GAGCCCTTCAAACA		mouse_ch
11	CTTT	p	1	6	24	(CTTT)6	3039572	3039595	tcttgcttttt	cttaatttctt	mouse_ch
12	T	p	1	14	14	(T)14	3039624	3039637	tgtctttgtta	gctgtggg	mouse_ch
13	A	p	1	16	16	(A)16	3041544	3041559	ttgtctccttt	TGAAAGA	mouse_ch
14	CAAA	p	1	6	24	(CAAA)6	3042496	3042519	taaggtaa	AACCCAA	mouse_ch

Now you want to select the content of columns named “motif”, “repeats”, “start” and “end” from the “ssr” table. you can use the following SELECT statement:

SELECT motif, repeats, start, end FROM ssr

The result will look like this:

motif	repeats	start	end
T	13	3006569	3006581
C	13	3011412	3011424
T	14	3018188	3018201
T	14	3018203	3018216
TTGT	5	3019659	3019678
TGTT	6	3025674	3025697
AG	8	3028600	3028615
CA	11	3031392	3031413
AC	8	3031449	3031464
AG	27	3031465	3031518

If you only want to select dinucleotide microsatellites from the “ssr” table, you can use the following SELECT statement:

SELECT * FROM ssr WHERE length(motif)=2

The result will look like this:

uid	motif	type	complex	repeats	length	seq	start	end	left	right	source
7	AG	p	1	8	16	(AG)8	3028600	3028615	cgggtgtg	acagaga	mouse_
8	CA	p	1	11	22	(CA)11	3031392	3031413	CAGGAA	tacacaca	mouse_
9	AC	p	1	8	16	(AC)8	3031449	3031464	TGTCTCA	agagaga	mouse_
10	AG	p	1	27	54	(AG)27	3031465	3031518	GAGCCC	TCAAACA	mouse_
15	AC	p	1	7	14	(AC)7	3046289	3046302	gggcattg	atacacac	mouse_
22	AT	p	1	32	64	(AT)32	3079582	3079645	TTCCTAG	acacacac	mouse_
23	AC	p	1	7	14	(AC)7	3079646	3079659	ATTTTGG	atacatttg	mouse_
25	AT	p	1	19	38	(AT)19	3091300	3091337	gaaaagta	gagcttac	mouse_
26	AT	p	1	9	18	(AT)9	3098797	3098814	aaatatca	ttcatatata	mouse_
27	AC	p	1	10	20	(AC)10	3098863	3098882	tggaagca	atatatata	mouse_

You are also allowed select columns from the “file” table. Use the following SQL statement to view the information in “file” table:

SELECT * FROM file

The result will look like this:

filename	size	count	length
mouse_chrX	166650300	83913	2288133

Note: only when you select all the columns from the “ssr” table can you use the “Export SSRs” and “Export for Primer3” button to export SSRs. Otherwise, you can use the “Export Data” button to export the returned data.

6.4.2 Count the number of SSRs

If you want to count the number of microsatellites in the database, you can use the following SELECT statement:

SELECT count(*) from ssr

The result will look like this:

count(*)
83913

If you want to count the number of microsatellites according to the motif, you can use the following SELECT statement:

SELECT motif, count(motif) FROM ssr GROUP BY motif

The result will look like this:

motif	count(*)
A	9570
AAAC	269
AAACA	143
AAACAA	12
AAACC	12
AAACCA	4
AAACG	1
AAACT	1
AAAG	170
AAAGA	43

If you want to count the number of dinucleotide microsatellites according to the motif, you can use the following SELECT statement:

SELECT motif, count(motif) FROM ssr WHERE length(motif)=2 GROUP BY motif

The result will look like this:

motif	count(*)
AC	5695
AG	2437
AT	3218
CA	2634
CG	41
CT	1551
GA	2019
GC	40
GT	3363
TA	2341
TC	2917
TG	4999

If you want to count the number of microsatellites according to the length of motif, you can use the following SELECT statement:

SELECT length(motif),count(motif) FROM ssr GROUP BY length(motif)

The result will look like this:

length(motif)	count(motif)
1	20886
2	31255
3	7065
4	18337
5	4611
6	1759

6.4.3 Count the length of SSRs

If you want to count the length of all microsatellites in database, you can use the following SELECT statement:

SELECT sum(length) from ssr

The result will look like this:

sum(length)
2288133

If you want to count the length of microsatellites according to the length of motif, you can use the following SELECT statement:

SELECT length(motif), sum(length) FROM ssr GROUP BY length(motif)

The result will look like this:

length(motif)	sum(length)
1	342414
2	974436
3	194721
4	530908
5	163340
6	82314

6.4.4 Export returned data

The returned data above can be exported by clicking the “Export Data” button. The data can be exported to an Excel file or CSV file. The CSV file which is a comma-separated values file stores tabular data in plain-text form can be used by R.

7 Sliding window plot

Sliding window analysis is a commonly used method for studying the properties of molecular sequences: data are plotted as moving averages of a particular criterion for a window of a certain length slid along a sequence. MSDB also contains a graphical application, SWP (sliding window plot), developed to reveal the distribution of microsatellites on a chromosome.

7.1 window of SWP

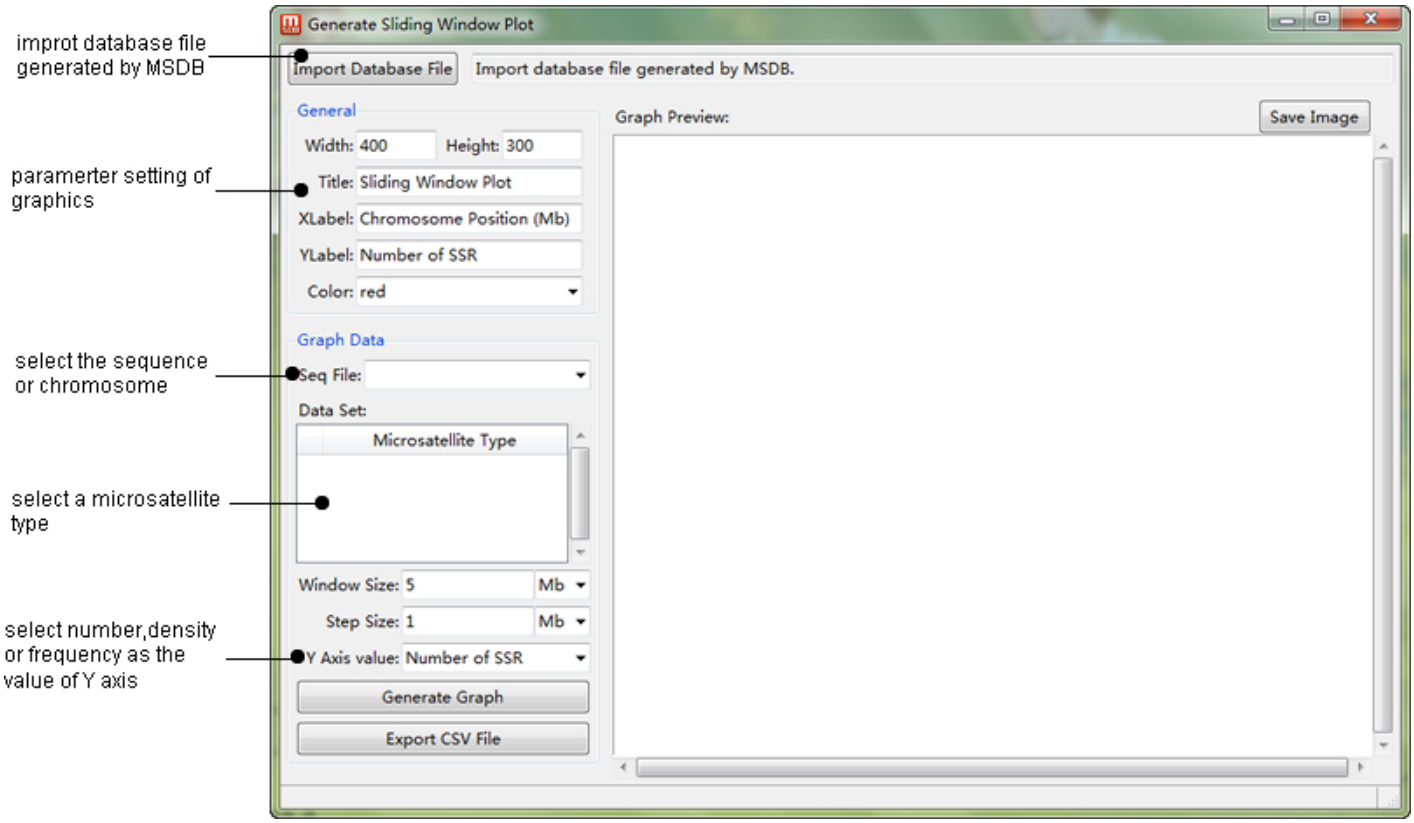

Fig. 20: the window of SWP

7.2 Generate sliding window plot

SWP can generate a sliding window plot for each sequence file or chromosome. SWP also allows generating sliding window plots for different types of microsatellites in the same graph (select multiple types of microsatellites using the Ctrl or Shift keys in data set). It is very important to select appropriate window size and step size for generating a sliding window plot. If the window is too small, it is difficult to detect the trend of the criterion measured; If too large, you could miss meaningful data.

7.3 Examples for SWP

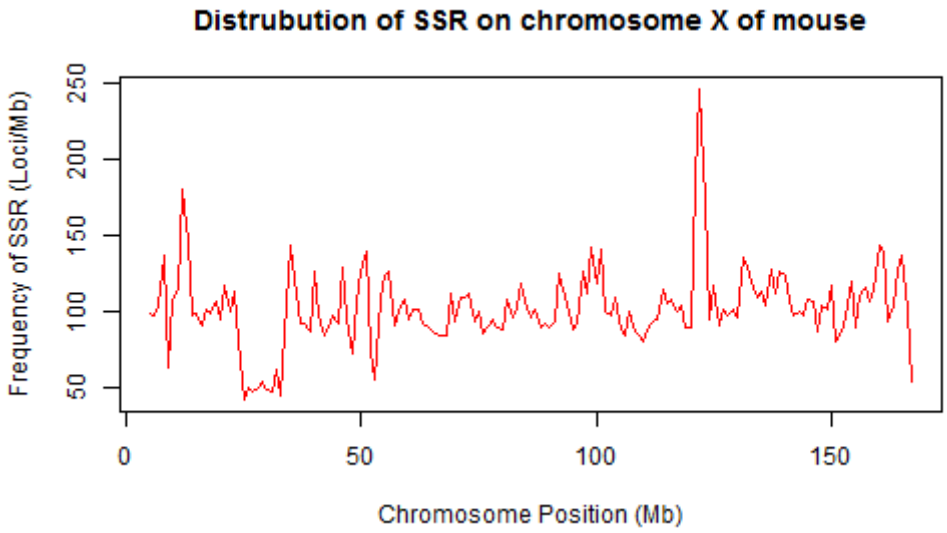

Fig. 21: Sliding window analysis of microsatellite frequency on the X chromosome of the mouse with a window size of 5 Mbp and a step size of 1 Mbp.

7.4 Export CSV file

The program SWP also allows you to export a CSV file, which can be used by R to generate graphics. The first two lines of the CSV file are the definitions of the columns.

```
#position - chromosome position (Mb)
#frequency - frequency of SSRs (Loci/Mb)
"position","frequency"
5,99.6
6,98
7,104.6
8,136.8
9,63.4
10,108.6
11,114.4
12,181.2
13,146
14,98.2
15,98.6
16,91.6
17,102
18,99.6
```

There is an example showing how to use the CSV file in R to draw a sliding window plot. We assume the generated CSV file was placed in the directory C:/Users/Leme/Desktop in Windows. The code was shown in the following:

```
x11()
setwd("C:/Users/Leme/Desktop")
table <- read.csv("table.csv", header=T, sep=",", dec=".", comment.char="#")
x <- table$position
y <- table$frequency
plot(x, y, type="l", main="sliding window plot", xlab="chromosome position (Mb)", ylab="frequency of SSRs (Loci/Mb)", col="red")
```

The generated sliding window plot will look like the Figure 21.

8 Feedback and Bug Reports

Though we are using this software successfully, there's a small chance of bugs turning up somewhere. Should you find any, please contact us and we will remove them. We welcome comments and suggestions for new features or other improvements.

9 Contact

Author: Lianming Du

Email: adu220@126.com