# Motif Mapper VBA-WsH Version 3.1.2

VBA-WsH Author: Kenneth W Berendzen
Perl Author: Kurt Stüber (select scripts)
Developers: Dierk Wanke, Kenneth W Berendzen, and Kurt Stüber

Site: http://www.motifmapper.de

## 0. Short Description

Motif Mapper is ideal for the Frequency and Distribution Analysis of elements in:

- Genomic pseudomolecules (i.e. chromosomes)
- Specific sequence sets (e.g. promoters) including frequency-distribution curves

The Motif Mapper VBA-WsH package is an association of independent scripts to assist in the analysis of the distribution and frequency of DNA elements. The scripts are are command line driven and written in the Visual Basic and Windows Scripting language recognized by the Microsoft Word's Macro Compiler (Editor). All output and input files are ASCII-text files whose data can be manipulated and viewed using array based spreadsheet applications. Alternative data-types extensions are to ease output identification. Any request for Perl scripts should be sent to Kurt Stüber. This package is therefore limited to Window operating systems (OS) since other OSs do not use the same objects (for example OS9 for Macintosh), but there is no overt support of any company or product. If there is significant in re-writing the package into another format please contact Kenneth Berendzen.

# 1. Table of Contents

## 2. License

The Motif Mapper Package is Open Source and covered by two licenses.
The first license is the GNU General Public License, Version 2 and the second is the Motif Mapper "Creative License". Both licenses are attainable from the compressed Package file. Updates and new algorithms will be posted from time to time on the internet at http://www.motifmapper.de .

Disclaimer:
All documentation and code has been provided as accurate and functional as possible but no guarantee or warrantee in any form is granted. All information and documentation is provided without any user support.
The user agrees to use any information obtained herein at their own risk.

When publishing, please reference:

Motif Mapper (http://www.motifmapper.de) K.Berendzen, K.Stüber, and D.Wanke, 2004.

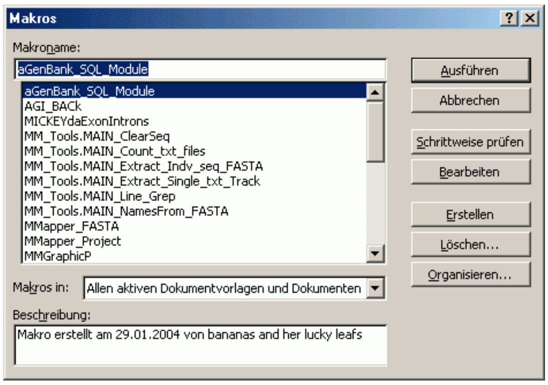
## 3. Core Algorithm and Principle

The Motif Mapper algorithm scans a string (one side of a DNA strand) identifying the non-overlapping number and position of a query (motif). Non-overlapping scoring reduces frequency increases of auto-correlating elements. Longer DNA strands are typically broken down into smaller widths required by the user, thereby creating a non-overlapping sliding window effect. This method facilitates the creation of a distribution map as well as simplifies the computational load due to the nature of the programming language. The observed frequency and expected frequency are returned for each element, allowing frequency based analyses for accessing over- and under-representation. However, in order to reduce autocorrelation - only the 5-prime motif is advanced (non-overlapping) allowing partial redundancy for over-lapping scoring of downstream motifs of single dyads. IUPAC handling has an additional benefit of preventing overlap scoring of degenerate motifs, which is an excellent way to score the number of physical binding sites. If there is really a strong interest in an auto-correlating version please contact motifmapper.

The estimated Observed and Expected probabilities are given for each motif entered which are mathematically the reciprocal of the average distribution length between events. Users are required to enter the CG content, and is set to a default of 0.36 for *Arabidopsis thaliana.*

The scripts have been organized as independent "Modules". The user can observe module names when the program is preceded by "MAIN". The modules are available from our webpage in compressed form and it is important to clarify the package sub-divisions. Each module carries all functions for the programs that it contains and therefore each does not require any other module, thus giving modular programs. However, the form frmFile.frm is mandatory for all modules and must be installed separately if the user does not use the provided Template file.

The programs are shown below in several images of Figure.1 since all of the twenty-eight programs are not visible on the Macro activation screen at the same time. (Please also notice that my version is in German, but the same format is of course in the English, or any other language version. All command line prompts are in English.)

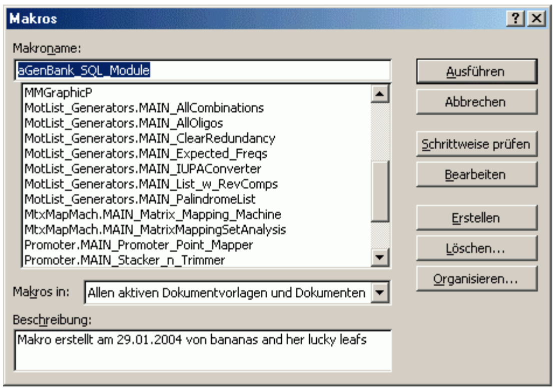


Makroname: aGenBank_SQL_Module

aGenBank_SQL_Module
AGI_BACk
MICKEYdaExonIntrons
MM_Tools.MAIN_ClearSeq
MM_Tools.MAIN_Count_txt_files
MM_Tools.MAIN_Extract_Indv_seq_FASTA
MM_Tools.MAIN_Extract_Single_txt_Track
MM_Tools.MAIN_Line_Grep
MM_Tools.MAIN_NamesFrom_FASTA
MMapper_FASTA
MMapper_Project
MMGraphicP

Makros in: Allen aktiven Dokumentvorlagen und Dokumenten

Beschreibung: Makro erstellt am 29.01.2004 von bananas and her lucky leafs

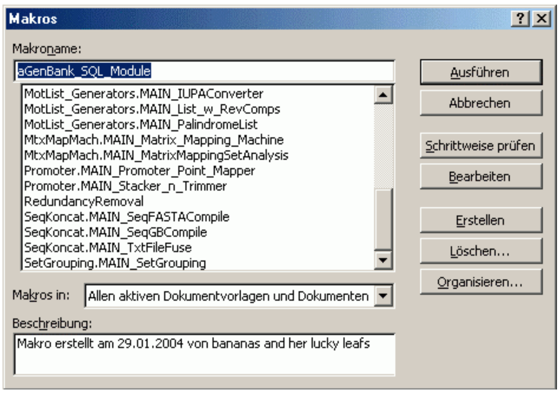


Makroname: aGenBank_SQL_Module

MMGraphicP
MotList_Generators.MAIN_AllCombinations
MotList_Generators.MAIN_AllOligos
MotList_Generators.MAIN_ClearRedundancy
MotList_Generators.MAIN_Expected_Freqs
MotList_Generators.MAIN_IUPAConverter
MotList_Generators.MAIN_List_w_RevComps
MotList_Generators.MAIN_PalindromeList
MtxMapMach.MAIN_Matrix_Mapping_Machine
MtxMapMach.MAIN_MatrixMappingSetAnalysis
Promoter.MAIN_Promoter_Point_Mapper
Promoter.MAIN_Stacker_n_Trimmer

Makros in: Allen aktiven Dokumentvorlagen und Dokumenten

Beschreibung: Makro erstellt am 29.01.2004 von bananas and her lucky leafs



Makroname: aGenBank_SQL_Module

MotList_Generators.MAIN_IUPAConverter
MotList_Generators.MAIN_List_w_RevComps
MotList_Generators.MAIN_PalindromeList
MtxMapMach.MAIN_Matrix_Mapping_Machine
MtxMapMach.MAIN_MatrixMappingSetAnalysis
Promoter.MAIN_Promoter_Point_Mapper
Promoter.MAIN_Stacker_n_Trimmer
RedundancyRemoval
SeqKoncat.MAIN_SeqFASTACompile
SeqKoncat.MAIN_SeqGBCompile
SeqKoncat.MAIN_TxtFileFuse
SetGrouping.MAIN_SetGrouping

Makros in: Allen aktiven Dokumentvorlagen und Dokumenten

Beschreibung: Makro erstellt am 29.01.2004 von bananas and her lucky leafs

Figure 1. The programs as displayed from the Macros “Run” option under “Extras” in MS Word.

## 4. Installation

The entire package, including auxiliary programs and forms, has been saved in a ".dot" format ("Template") which can be opened by directly clicking on the template file or linked by going to "Tools/Templates and Add-Ins/" (German: "Extras/Vorlage und Add-Ins/") and then place it in the Global documents by clicking on "Insert" (German: "Hinzüfugen) and locate the template file. The template file must be present in order to be accessed. It is not stored somewhere else as a copy. Then check the Template file entered to activate it (this is done automatically the first time you install the template this way). If the later option is taken, the VBScript Regular Expressions 5.5 must be activated. In English, open the Extras/Macros/Visual-Basic Editor. Once in the Editor, go to "Extras/References" (German: "Extras/Verweise"). Look for the Library and activate it. When Word is closed, the package will also be closed. Each time the user wants to run a program from the package, one must go back to "Tools/Templates and Add-Ins/" and reactivate the package. Another alternative is to open the Template file then go to "Tools/Templates und Add-Ins/Organize" (German: "Extras/Vorlagen und Add-Ins/Organisieren"). The from "Macro Project Elements available in" (German: "Makroprojectelemente verfügbar in") select MM3.1.2 and each module can be copied from MM3.1.2 to the Normal, the global word Template. After this is done, then all of the programs are then permanently installed and are accessible from "Tools/Macro/Macros" (German: "Extras/Makro/Makros/"). Please remember, that in order to use any of the programs, one must have Word open and the "Template" activated, therefore it is just easier to just click on the template file. This also means that one has to have Word open in order to run the programs since Word then serves as the complier. Since this is only one installation option, we also provided all modular programs separately in a compressed file which can by manually loaded into Word's "Normal" which is always available for macro use. Go to the Editor, then "File/Import File" (German: "Datei/Datei importieren")and choose the file. Be careful not to save the modules a word document! The module or form will be loaded into the Normal (global template) and will be permanently available.  For a short discussion about security risks and potential macro viruses, see **security**.

# 5. Definitions

## 5.1 General Definitions

motif

a small sequence of letters (a word) which can be contained within a larger word.

GC content

refers to the combined frequency of Guanine (G) and Cytosine (C).
GC content = 1 - (Feq.Adenine + Feq.Thymine).

word

like in English, a series of letters orientated from left to right without spaces between the letters constituting one single entity.

element

typically, an element is a DNA motif that has been characterized at some molecular biological level.

string

a long continuous stretch of characters, including white spaces, a computational term.

DNA strand

the DNA strand refers to the molecular structure also called the "double helix", possessing two complementary poly-nucleotide molecules that are paired together by hydrogen bonds between individual complement nucleotide subunits.

distribution map

any type of schematic representation of location and number of motif occurrences along a string.

bytes

each letter (nucleotide) is typically saved in one byte of information in a computer. One byte is composed of 8 bits of information. Since the DNA strand is represented with one side (as the other side is "encoded" in the given strand), each nucleotide takes one byte of memory space, but contains BOTH sides of the DNA strand, thus each byte contains information for one base pair.

IUPAC

The standardized IUAPC Ambigous nucleotide codes recognized by Motif Mapper.

| | |
|---|---|
| A; T; G; C | nucleotide |
| R = A or G | puRines |
| Y = C or T | prYimidines |
| W = A or T | Weak hydrogen bonds |
| S = G or C | Strong hydrogen bonds |
| M = A or C | aMino group at common position |
| K = G or T | Keto group at common position |
| H = A, C, or T | not G |
| B = G, C, or T | not A |
| V = G, A, or C | not T |
| D = G, A, or T | not C |
| N = G, A, T, or C | aNy |

bps

base pairs. See bytes for discussion.

module

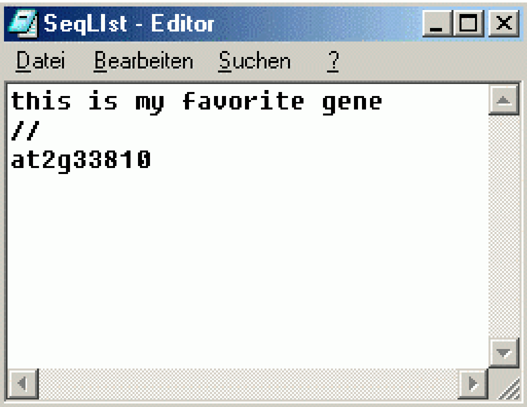
here, the organization of subroutines (programs) resident in a Visual Basic Project. (Please refer to Microsoft's homepage for more information).

MM-list

- a list of motifs preceeded by a line with only ".."
- a list of NAMES preceeded by a line with only "//"

Anything above these lines is ignored, allowing the user to place comments.

Each motif or NAME must be on a separate line. Example :



```
this is my favorite gene
//
at2g33810
```

FASTA

a format where a sequence is identified by a name on a preceding line following a ">" character. The end of a line can be identified as a LF (ASCI-II 10) or CR (ASCI-II 13). CR stands for "Carriage Return" and most text editors will display this character as such (the following text is then on a new line).
Example:

```
>aGeneName
atcgatgctagctgatcgtagctgactgatcgtgatcgatc
>anotherGene
gatcgactacgatcttctctatcattctatctacgatagga
```

dyad

a dyad is a pair of DNA elements that are bound by a transcription factor with spacing between the two motifs due to minimal to no contact of the protein to the DNA between the motifs. Dyads can be represented as xxx{n,m}xxx , where "x" represents any nucleotide, "n" a minimal number of nucleotides before the 3p motif that is allowed to be "m" nucleotides away. MotifMapper employs non-autocorrelative scoring for 5p motifs, that is if a dyad is found, then the next search begins at the nucleotide after the 5p motif. However, there are two options for 3p motifs: one, any 3p motif must be independent of the 5p motif and distances are calculated beginning after the 5p motif; two, 3p motifs are allowed to overlap with the 5p motif (except for the initial nucleotide, otherwise that would constitute a recursive call for direct repeats) and distances are calculated as occurring within in the 5p motif region plus any imposed spacing from the user.

Observed frequency (Obs.p)

is an estimate of the average probability to observe an event that has been scored for present/not-present over a series of trials based on an actual number of events counted. However, when talking about DNA motifs observed in a DNA strand, the observed frequency describes the length between each motif assuming that the motif is evenly distributed. Therefore, the observed frequency describes the average length one would expect to go before observing the given motif once. Frequencies are usually employed to compare different motifs in order to determine if one motif is differentially present.

Expected frequency (Exp.p)

is an estimate of the average probability to observe an event that has been scored for present/not-present over a series of trials based on sub-word frequencies. Quite often, the expected frequency is calculated based on the individual nucleotide frequencies. This however is drastically shifted in di-nucleotide frequencies and longer words are best approximated by using the direct sub-word content (reference goes here). That is, one would use the observed frequencies of Wn-1 to calculate the Wn expected frequency. Motif Mapper uses the known the nucleotide frequency (GC content) as a better approximation of frequencies than assuming equal weighting of the 4 DNA nucleotides in which each has a 25% chance of being in each position for any word, which does not occur in nature too often.

## 5.2 Explanation of Frequency Output

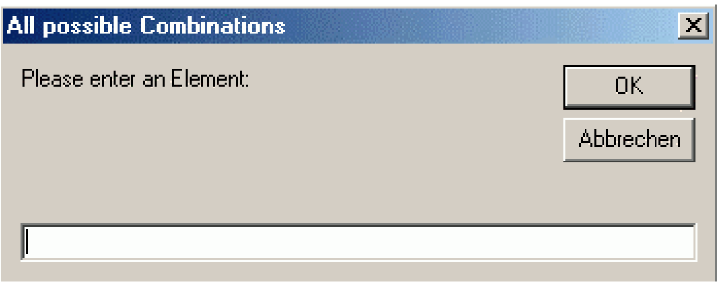
The estimated Observed and Expected probabilities are given for each motif entered which are mathematically the reciprocal of the average distribution length between events. This is also the averaged distribution frequency commonly employed in publications. An informal mathematical argument is given by K.Berendzen as Calculation of the observed probability. When dealing with plus/minus probability events, the binomial distribution provides a means to calculate the proportion of times an event happens, in our case the presence of a motif. It is well known that with a large number of trials (n) and a low probability of observing the event the binomial distribution approaches a limiting distribution called the Poisson distribution [Statistics of Experiments, Box Hunter and Hunter, John Wiley and Sons, 1978]. The larger (n) becomes the more the Poisson distribution can be well approximated by a normal distribution. The binomial and Poisson distributions must be calculated for the probability of observing none, one, two or more events and provide a well defined means to determine significance [van Helden et al., *J. Mol. Bio.* 1998. 281: 827-42, van Helden, *Bioinformatics* 2004. 20(3): 399-406]. However, the question can be asked what is the waiting time (number of trials) before one observes an event with a probability (p)? In other words, when searching for words in DNA/RNA strings, how many bases must one sample before one finds the sought motif ? The geometric distribution returns the number of unsuccessful trials before the first success, or in our case, the length of the DNA strand before the first motif is found. The reciprocal of the frequency distribution can be shown to be a good approximation of the geometric distribution and is employed in Motif Mapper. The negative-binomial distribution is a repetition the geometric distribution until the r-th success by simple multiplication of the number of desired successes. Only the number of failures is returned by the geometric and negative-binomial distribution, which again, is the length of the DNA strand before the first motif (or r-th motif) is found. Due to the fact that both the binomial, Poisson, geometric and negative-binomial distributions are normalizing with large (n) also give frequency distribution shapes, which are normal, since they describe the average number of bases (trials) between motifs (events) .

## 6. Getting Started with Motif Mapper

### 6.1 Example One

After you have installed the package you may want to see if you can run the programs. It is a good idea to test one of the programs; the example given here is MAIN_AllCombinations under the MotList_Generators module.

Open Word and go to "Tools/Macro/Macros" (German: "Extras/Makro/Makros") and then find "MotList_Generators.MAIN_AllCombinations" . Choose run. You will then see the Command Line prompt, or Standard-Input for Visual Basic.



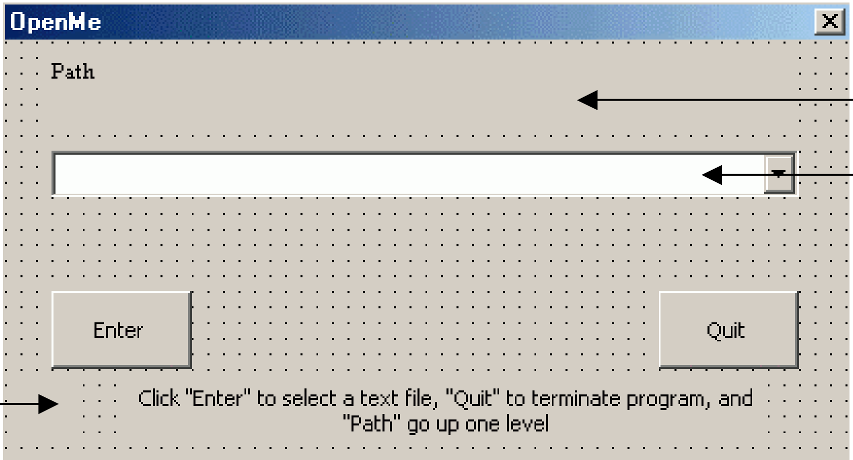
Enter any element (here: ELEMENT_GIVEN) and the total possible combinations will be returned under C:\MotifLists\ bearing a file name "ELEMENT_GIVEN-perms.txt". All redundancy is be removed. For example, if we enter "123" for ELEMENT_GIVEN, the output file is "123-perms.txt" found under the C:\MotifLists\ directory.

Output from input "123":

```
Permutation Motif List Generated by MotifMapperVBA-WsH Package Vers.1.2
Total Permutations of Element: 123
Maximum number of permutations possible: 6
..
123
132
213
231
312
321
```

## 6.2 Opening Files, the form OpenMe

One could enter file names from the Command Line, but this can be quite tedious. Instead the form OpenMe provides a simple graphical user interface.



Click "Enter" to select a text file, "Quit" to terminate program, and "Path" go up one level

click here to go up in the directory tree

click here to open directories and select files

click here to change the file type (default is txt). Password = "forsure"

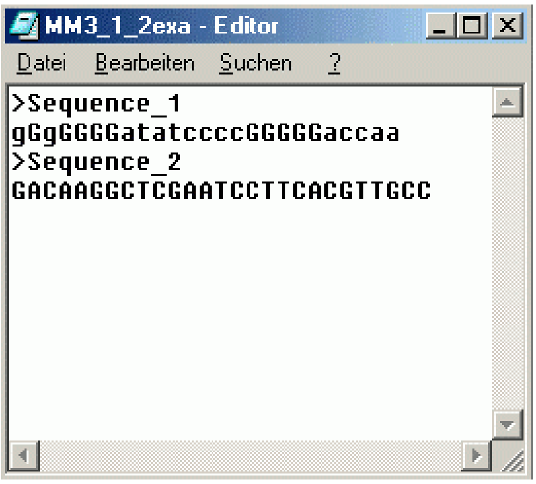
The GUI (Graphical User Interface) form "OpenMe" which allows the user to select files All files initially must be text-files only in order to avoid any hidden characters that are normally tucked away in word-processing documents which can skew the analysis. Therefore by DEFAULT, only ".txt" appendices are accepted to prevent any input errors. In the bottom left hand corner, is a toggle function allowing the user to change the filename extension when necessary, for example when opening ".mset" output files which are text files, but with a different filename extension. YES!, the toggle function is hidden, just click there and find out! You can see it here since this is an image of the form before compiling.

This small program helps break the tediousness of command line driven programs.

You might notice some other sequence related programs have not completely integrated automated processing for all of their applications (programs). This means, that in a sense, they are still command line driven forcing the user to interact at points where it doesn't make sense, like forced saving of a file before continuing work with it.

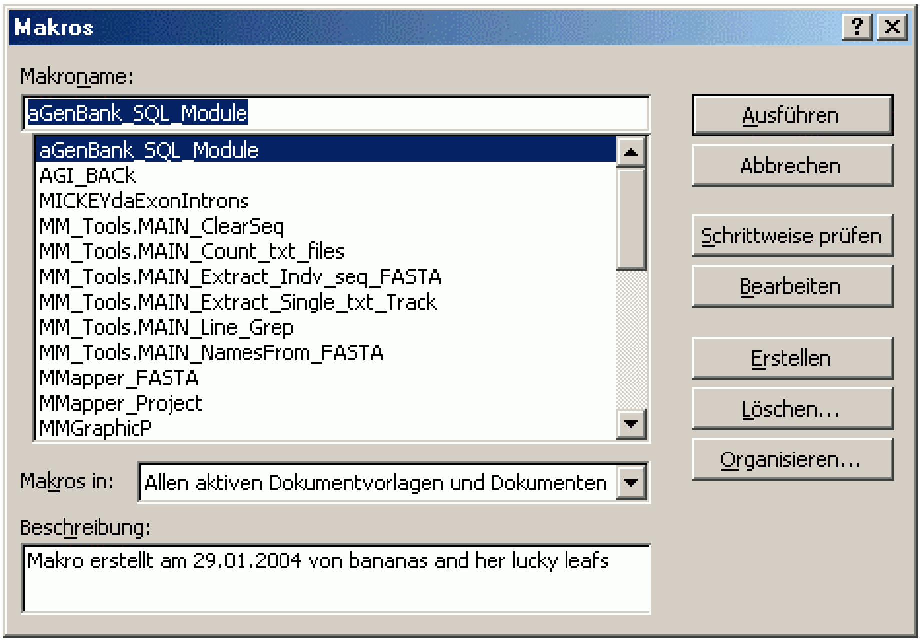
## 6.3 Example Two, running MMapper_FASTA

**( 1 )** You need to have a text file with at least one FASTA formatted sequence. For this introduction, two small random sequences were typed in.
If you want, you can copy them into a text file and save them and run the program from your machine.
>Sequence_1
gGgGGGGatatccccGGGGGaccaa
>Sequence_2
GACAAGGCTCGAATCCTTCACGTTGCC
The two sequences given here were saved in a file called "MM3_1_2exa.txt", like so :



```
>Sequence_1
gGgGGGGatatccccGGGGGaccaa
>Sequence_2
GACAAGGCTCGAATCCTTCACGTTGCC
```

**( 2 )** Make sure that you have tried out Example One on your machine already. Open Word again, and go to "Tools/Macro/Macros" (German: "Extras/Makro/Makros") and then find MMapper_FASTA. The Macros should look something similar to the screenshot shown below. Click on MMapper_FASTA to run the script.

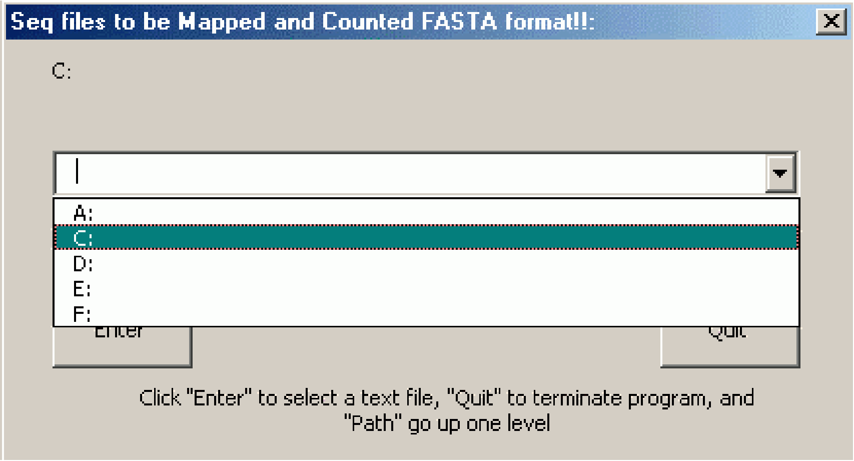


Makroname: aGenBank_SQL_Module

Makros in: Allen aktiven Dokumentvorlagen und Dokumenten

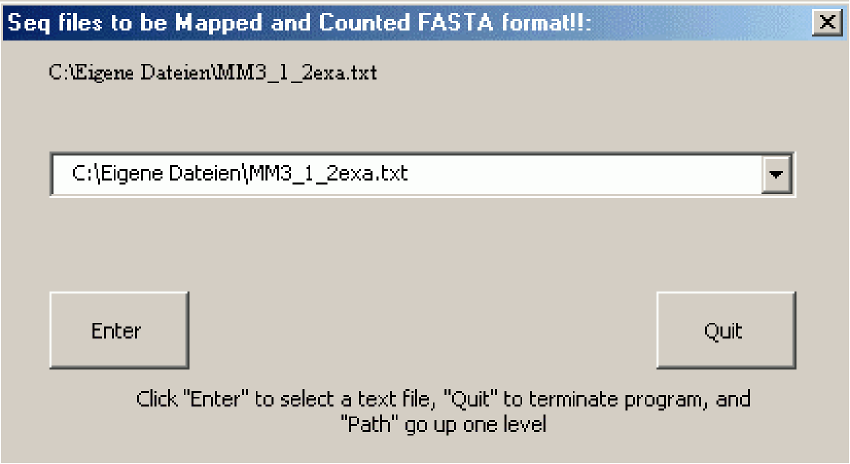
Beschreibung: Makro erstellt am 29.01.2004 von bananas and her lucky leafs

**( 3 )** The first interface that opens is the OpenMe form. You are required to enter at least one file, but are allowed to enter multiple FASTA formatted files for this script.



Click "Enter" to select a text file, "Quit" to terminate program, and "Path" go up one level

Go to the appropriate folder and choose the file. The file-path will appear above in the Path-Display Object. If you click on the wrong folder, you can go back up in the directory tree by clicking on the Path-Display Object itself. Experiment!



Seq files to be Mapped and Counted FASTA format!!:

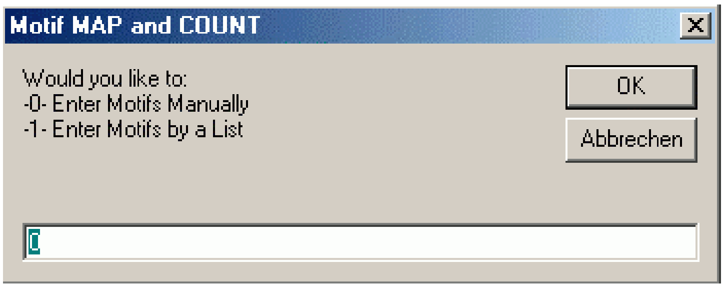
C:\Eigene Dateien\MM3_1_2exa.txt

C:\Eigene Dateien\MM3_1_2exa.txt

Click "Enter" to select a text file, "Quit" to terminate program, and "Path" go up one level

Click "Enter" for the files you are satisfied with.

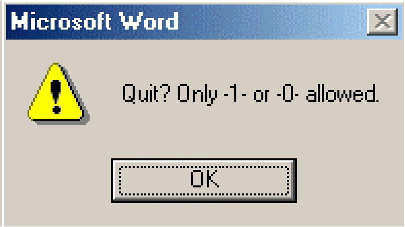
**( 4 )** After the files are entered, the user is required to enter the motifs.



Motif MAP and COUNT

Would you like to:
-0- Enter Motifs Manually
-1- Enter Motifs by a List

Two options are given; either enter motifs manually, that is one-by-one until the user is finished OR the user can select a text-file motif list. If the user selects any other character besides "0" or "1" then this error message below is displayed and the program terminated.



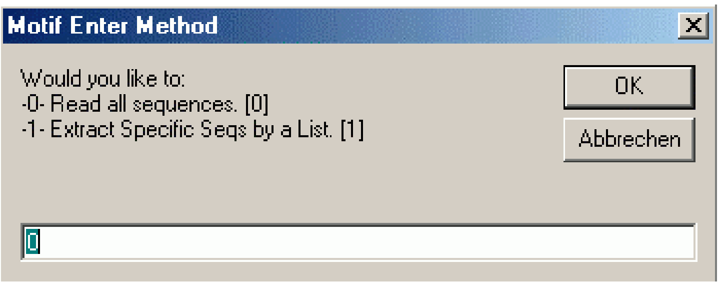
Microsoft Word

Quit? Only -1- or -0- allowed.

For this example the following motifs were entered in the following order :

```
gg
Gg
GC
TCT
GG{2}G
gg{0,2}G
GG{2,8}cc
```

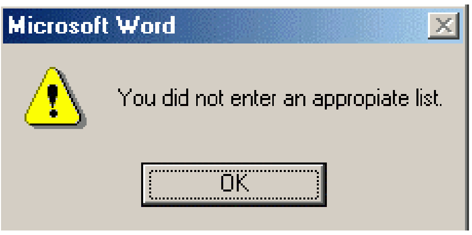
**( 5 )** Once the motifs are entered the user the gets a choice to either analyze ALL sequences in the given FASTA formatted file or to only scan a select number of sequences. Only a select number of sequences matching a NAME list are scored.


Motif Enter Method

Would you like to:
-0- Read all sequences. [0]
-1- Extract Specific Seqs by a List. [1]

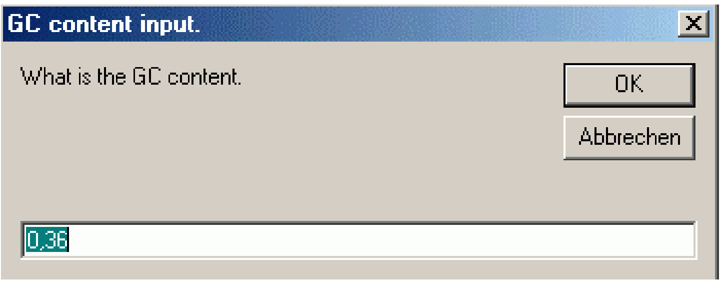
0

If neither a "0" or a "1" are entered, the program terminates, and if an inappropriate list is entered, a error message is displayed and the program terminated.


Microsoft Word

You did not enter an appropiate list.

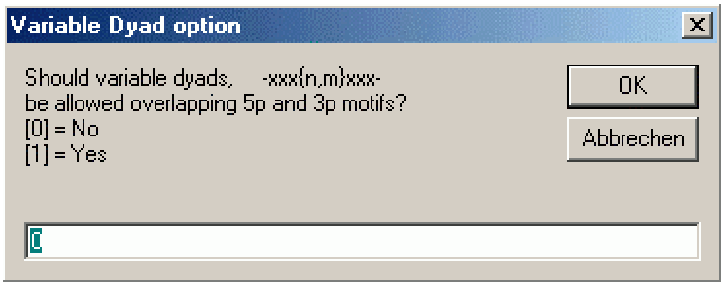
**( 6 )** The user can then set the GC content necessary to calculate the expected frequency. Keep in mind that if you are working with many different organisms and would like to use the expected frequency data, then it is best to enter each FASTA sequence list individually instead of entering all of your files at once. However, if this is not the case and one would like to generate expected frequencies, the program MotList_Generators.MAIN_ExpectedFreqs is available. GC content is checked to make sure that the value entered is numeric and between 0 and 1.


GC content input.

What is the GC content.

0.36

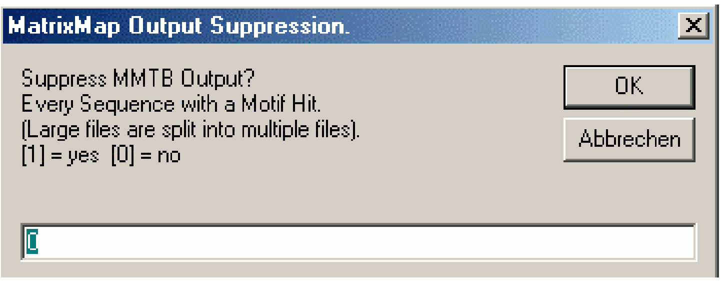
**( 7 )** Next, the option for dyad scoring is given. The Motif Mapper word-counting algorithm is based on non-overlapping words read from left to right of the input sequence. The default, and recommend scoring method is not to allow the 3p dyad-motif to overlap with the 5p dyad-motif. However, it may be interesting to know if a dyad does overlap. This is especially useful for the dyad-pair motifs that have micro-homology or are very auto-correlative.



Should variable dyads, -xxx{n,m}xxx-
be allowed overlapping 5p and 3p motifs?
[0] = No
[1] = Yes

0

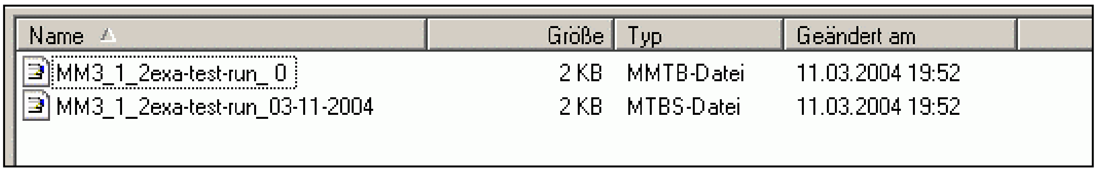
**( 8 )** The user is now given a choice to suppress the MMTB output.



Suppress MMTB Output?
Every Sequence with a Motif Hit.
[Large files are split into multiple files].
[1] = yes [0] = no

0

Two output files are generated by MMapper_FASTA.



| Name | Größe | Typ | Geändert am |
|---|---|---|---|
| MM3_1_2exa-test-run_0 | 2 KB | MMTB-Datei | 11.03.2004 19:52 |
| MM3_1_2exa-test-run_03-11-2004 | 2 KB | MTBS-Datei | 11.03.2004 19:52 |

The first file "MMTB" is appended with "_0" and contains for each sequence each motif hit. If there are multiple hits for a motif, then each motif hit is separated with a semi-colon ";". Manual extraction of this data can be used to create distribution maps and used for statistical analysis. A small sample of the MMTB output is shown below for the motifs "gg" and "Gg". Scoring is case-insensitive, therefore the number of motifs found are identical. The output file is a TAB delineated file, this means the white-space between the columns "Seq" and "Seq.len" is only the TAB character (ASCII-code 9). To easily see the column information, it is recommend to open the file from a spreadsheet application like Excel or PAST (http://folk.uio.no/ohammer/past).

Example "MMTB" output:

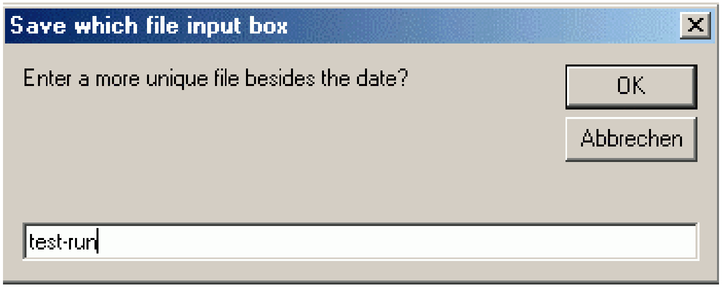
| Seq | Seq.len | Seq.Ori | MOTIF | Pos. | Total Counts | Exp.p | Exp.Occr | Obs.p | Obs.p/Exp.p |
|---|---|---|---|---|---|---|---|---|---|
| Sequence_1 | 25 | U | gg | 1;3;5;16;18; | 5 | 0,0324 | 0,81 | 0,2 | 6,17283950617284 |
| Sequence_1 | 25 | U | Gg | 1;3;5;16;18; | 5 | 0,0324 | 0,81 | 0,2 | 6,17283950617284 |

The file "MTBS" contains a summary of all motifs and scores their frequency against the entire data set length. This is particularly useful for defined sequence sets, like promoters, introns or exons. For the example given, the "Absolute Sequence Length" is 52 used to calculate the total number of expected occurrences "Abs.Exp.Occ".

| MOTIF | Seqs Hit | Total Hits | Obs.p | Exp.p | Abs.Exp.Occ. | Obs.p/Exp.p | |
|---|---|---|---|---|---|---|---|
| gg | 2 | 6 | 0,115384615384615 | | 0,0324 | 1,6848 | 3,56125356125356 |
| Gg | 2 | 6 | 0,115384615384615 | | 0,0324 | 1,6848 | 3,56125356125356 |

See the MMapper_FASTA description for further information.

**( 9 )** Finally, the user can enter a unique name for personal reference. For this example the name "test-run" was entered. The final names are shown above in Section 8.



Save which file input box

Enter a more unique file besides the date?

test-run

The output is delivered under the directory C:\MapnCtr\ NAME OF INPUT FILE \ . That means, for this example the two output files shown in Section 8 are found under C:\MapnCtr\MM3_1_2exa\ .

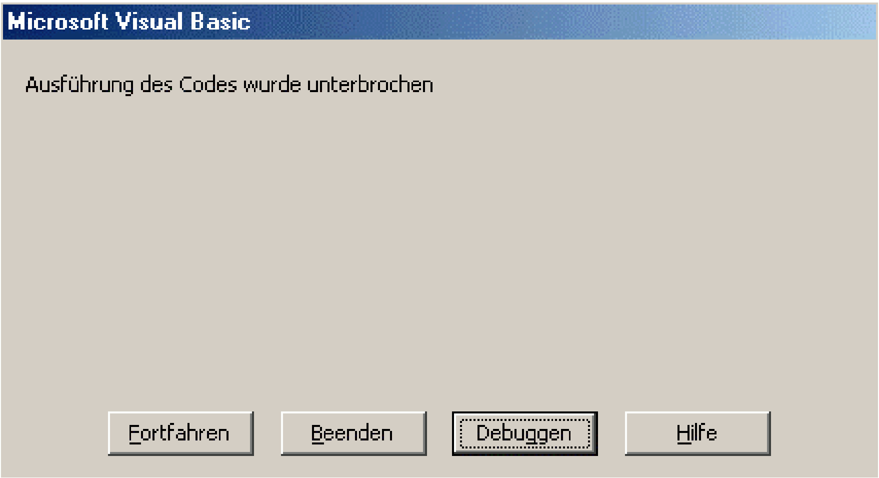
**( 10 )** Once the final command-line is entered the program runs until it is finished. Keep in mind that Word is now running the program and can not be used to perform any other functions. Once the program is finished you are allowed to work with Word again. So that means, as soon as Word is finished you can proceed and run the program again if you want! It is not recommended to try to work with any other programs while running the scripts, although it is possible on occasion there is a memory conflict since the script has priority. In other words, you have to wait anyway until your program is finished running.

**( 11 )** The output from Example Two can be found as:
"mmtb.txt" AND "mtbs.txt" at http://www.motifmapper.de
under Documentations/Getting Started/(11).

**( 12 )** If you have successfully executed Exercise one and two then you are ready to use MotifMapper VB-WsH. The package is open-source and licensed under the GNU General Public License, so feel free to get into the code if you wish. Otherwise, this package should help you in the analysis of large DNA-sequence data sets on the privacy of your own computer.

## 6.4 A few words on Error-Handling

What happens when unacceptable input is entered?
The majority of command-line requests have error-handling code. For example, the form OpenMe has an internal check when the user enters a file by requesting a conformation. Error handling has been implemented as either input restriction by calling a continuous loop until the appropriate input is entered, or accepting the input and if inappropriate, terminating the program. Program termination is implemented in case the user has decided that he or she would like to start over. On rare occasion, some of the command lines recognize the command "quit" as an exit command in order to terminate the program. While the best attempt has been made at including error-handling at most input requests, neither do all command-lines all contain error trapping code nor do all error checks guarantee that the user will not enter something that is nonsense and cause the program to crash. When this happens the Visual Basic debugger is called and this window will probably appear.



Microsoft Visual Basic

Ausführung des Codes wurde unterbrochen

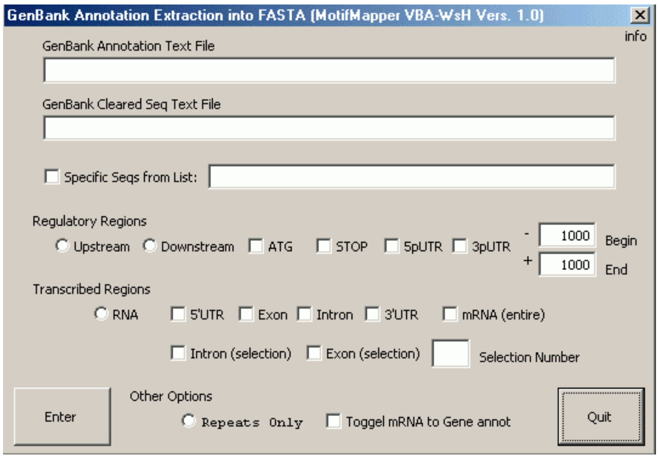
The four options are (1)"Continue" (2)"End" (3)"Debug" (4)"Help".
The majority of crashes will probably come from user incompatible input. For example if the user types "gg{g}g" instead of "gg{2}g" the debugger is called since there is no handling to guarantee the the value in-between the two curly-brackets is numeric. Therefore, it is up to the user to guarantee that he or she enters sensible data.

# 7. Programs

## 7.1 Analysis Programs

### 7.1.1 aGenBank_SQL_Module

This program is a moderately powerful sequence extraction tool for GenBank annotated sequence files that are found at http://www.ncbi.nlm.nih.gov. We are in no way affiliated with GenBank, but the sequence repository is open and available to the public and offer a large variety of sequence data. The majority of what is annotated on a GenBank annotation file can be extracted with this program and the sequences are written to an outfile in FASTA format. In addition, upstream and downstream sequences can be extracted from any relative start position. Two separate text files are needed; a GenBank file and the sequence itself in a cleared format. The GenBank file must be in text format (please visit the homepage for tips on retrieving and working with text files) and the sequence can be cleaned (extracted) using **SeqGBCompile** as long as only one file is cleared at a time (please see the program documentation for more detail). Only if these two files belong to each other then will any extracting be allowed (thus nonsense is prevented). An additional object for the module is needed, an object form called frm_aGenBank, which is the GUI presented below and is automatically loaded with the template package. I hope Figure 4 is self-explanatory. Extraction rules are logical and you will notice them as you experiment with the program, for example; in order to get all individual exons or introns you just have to check exon(selection) or intron(selection) and leave the selection number blank. All sequences will then be retrieved individually in FASTA format. If you check the intron box above these functions, it is assumed that the user wants the transcription product from ATG to STOP including introns.



GenBank Annotation Extraction into FASTA (MotifMapper VBA-WsH Vers. 1.0)

- 1000 Begin
+ 1000 End

info
GenBank Annotation Text File
GenBank Cleared Seq Text File
□ Specific Seqs from List:
Regulatory Regions
○ Upstream ○ Downstream □ ATG □ STOP □ 5pUTR □ 3pUTR
Transcribed Regions
○ RNA □ 5'UTR □ Exon □ Intron □ 3'UTR □ mRNA (entire)
□ Intron (selection) □ Exon (selection) Selection Number
Other Options
○ Repeats Only □ Toggel mRNA to Gene annot
Enter
Quit

Figure 4. The Graphical User Interface controlling the GenBank SQL sequence extraction program GenBank_SQL_Module.

### 7.1.2 MMapper_Project

Command lines are controlled by a series of dialog boxes called “InputBoxes” or “MessageBoxes” requesting information or returning information respectively :

1. Which files to be read.
2. Motifs are taken (manual or list). See **motifs** below-.
3. An appended name to the outfile is requested, otherwise a non-descriptive default name is given.
4. A choice between distribution mapping, counting-frequency statistics, or both is given.
5. GC content is asked with default GC=0.36 given for *Arabidopsis*.
6. Variable dyad option. [0] = no overlapping of 3p motif with 5p motif; [1] = overlapping allowed.
7. A counting project option is given, which will deduct a user defined number of bytes from the final sequence length (appropriate for only single query).
8. Output is delivered in "C:\MapnCtr\" in a NEW FOLDER bearing the query sequence file name. All map files (that is each motif) are written to individual files and count-data is returned in one single text file per input sequence file. Both maps and counting infos can be analyzed in spreadsheet analysis programs, for example MS Excel or PaST (http://folk.uio.no/ohammer/past). Data can be either copy-pasted or opened directly.
9. The window size is requested (suggested max 25 000).

**Motifs** are inputted manually or by **list**. Multiple files can be entered to be searched. Please note that ANY file can be mapped, because it is even possible to open an image file as a text file; the only result will be rubbish of course. Therefore, it is critical that the user knows and understands what file they are using what their search DNA string is. The option to subtract a set number of bytes (bps) is present just in case one is using a string of discontinuous sequences or for example, a sequence with too many Ns which will heavily skew the frequency calculations. We assume that the user understands, at least in principle, frequency analysis and will be able to recalculate the **frequency** if he or she discovers that they do have a large number of Ns in their sequence or something else similar. It does not make any sense to subtract a set number of bytes when analysing multiple files if only one file needs the correction. In this case it is better to make a separate run, or recalculate the frequency after the run. The primary data, the number of times a motif is observed, can not be changed and is always accurate as is the subsequent distribution map, it is only frequency conversion which needs occasional attention, for example if one needs motifs contrasted to their **expected frequency** with the proper **GC content** in a multi-file run. This is a slight hindrance for multiple files, but extra command prompts will only make the program tedious and user-feisty. All IUPAC characters are recognized except "X".

The maximum window length that I can get on my computer is 25 000 **bytes** = 25 000 **bps**, but this could be longer with a faster processor and more memory. Even though the core algorithm is the same in all versions, not all program versions use the same input or return the same output, each naturally having their advantages and disadvantages. For example, one would use one version for a continuous string input and another version for FASTA formatted data.

With increasing motif length (>15bp) it is very important to use larger windows if scanning long DNA strands like a chromosome. This is critically important for very long motifs or very rare motifs, since every window destroys n-1 chances for identifying your element (n = length of motif). For smaller motifs, if you have a large data set (again a chromosome), you should still get a fair distribution because increasing the window size typically exposes more of the more frequent and a bit more of the very infrequent, that is the frequency distribution of observed hexamers in searched sequences does not change between window sizes. Of course, take at least 1kb (1000bps) when scanning along chromosomes since very small windows will eventually disrupt frequencies (because you interfere with chances of observing them by cutting the DNA into pieces). In other words, there is a type of periodicity of error that comes from sampling error when breaking down the available trials which are relatively proportional to the strand length.

### 7.1.3 MMapper_FASTA

This is a very powerful complement to the MM_Project. Output is not the same as MM_Project. Sequences must be in FASTA format and each line (sequence identifier and the sequence itself) must be terminated by a CR (ASCI II-code 13) or a LF character (ASCI II-code 10) not just a tab. Multiple FASTA files can be entered. Motif-lists are also accepted as are sequence specific lists, to allow the user to specific which sequences from a FASTA file she would like to analyze. The distribution information is ideal for smaller sequences since the exact 5-prime position is returned for every non-overlapping motif found. This output is appended at “.MMTB”. This is in great contrast to MM_Project where only counts per window, not per position, are given because MM_Project is optimized for analyses of long DNA stretches. The 5-prime positions of each motif is given separated from the next identified motif by a semi-colon ";". Dyads are returned as "n-m;" for each entry. As a side note, if the file is opened from Excel, you will automatically be given the choice of which characters can be used as column break identifiers. Extremely large files are broken at 60,000 rows, near the limit for MS Excel, and therefore a appending “_0” is given to the name of the initial file. A summary file, grouping all motifs counts against the total scanned sequence length is given as a summary appended by “.MTBS”. All IUPAC characters are recognized except "X".

Command Lines for MMapper_FASTA:

1. Which files to be read, FASTA format only!
2. Which motifs (manual or list). See **motifs** under MMapper_Project.
3. Option to search all sequences with the file or a subset. If a subset is chosen then a list denoted by “//” on a single line must be entered. Please consider that the file names are case insensitive.
4. GC content.
5. Variable dyad option. [0] = no overlapping of 3p motif with 5p motif; [1] = overlapping allowed.
6. Option to suppress the MMTB file is given. This file contains all hits per motif per sequence and can be extremely long for long motif lists or large sequence sets. The summary file is always returned.
7. Outfile append Option.

### 7.1.4 MtxMapMach

This module is ideal for the analysis of the presence of motif elements that can be defined by a matrix (PPSM). Output is returned under input specific file name sub-directories under folder "C:\MaxtixMaps\". Input must be in **FASTA** format and sequence lists ("//") are also handled. Matrices are entered as positional arrays where the elements must be of equal length and the total number composing the matrix group must be known. The number of sequences with a particular nucleotide in each position is adaptively scored 5-prime to 3-prime. Total scores are defined by (and would be equal to one for a perfect hit):

$$( \Sigma \text{ Score of Nucleotides at each position } ) / L / N$$

where L is the length of the motif and N the number of entries use to create the matrix values. The matrix is run along the length of each sequence and the score returned at each 3-prime position. Where there is a non-permissible letter (for example a "G" in position 3 is 0 indicating that a "G" is never present at that position) then the element is thrown out by setting the entire value to zero. A companion file is written containing a list of each sequence mapped who contain at least one matrix hit that is above an arbitrary threshold value set by the user. This specific matrix application designed by Björn Hamberger, MPIZ Cologne 2002 (bjern@canade.ca).

Matrix format::
You can store a matrix as a text file anywhere. The general formula is N:=#-#-#-#-#-#-#-# where the last number (#) must NOT be followed by anything else (although an accidental “-“ will be removed from the end), otherwise an error will occur (case is irrelevant). Therefore the scores of a matrix are formulated for each position for each nucleotide in 5prime to 3prime orientation. Also note the sum of each position must be equal to the number of sequences used to create the matrix. That is, each score for each position is per sequence.

An example

| | |
|---|---|
| seq1 | ATGCTA |
| seq2 | ATCCTA |
| seq3 | ATGCTT |

and corresponding matrix :

```
a:=3-0-0-0-0-2
t:=0-3-0-0-3-1
g:=0-0-2-0-0-0
c:=0-0-1-3-0-0
```

#### 7.1.4.1 Command Lines for Matrix_Mapping_Machine:

1. number of sequences used to create the matrix
2. matrix is entered (see notes below)
3. user defined threshold is set
4. which files to be read (FASTA format only!)
5. Option for general file append to all outfiles
6. Option to suppress individual sequence matrix outputs (if suppressed only the summary file is returned giving the list of sequences that have one or more hits above the given threshold
7. Option to search all sequences or a list "//"

#### 7.1.4.2 Command Lines for MatrixMappingSetAnalysis:

1. Number of sequences used to create the matrix
2. matrix is entered
3. threshold
4. Input files are taken (Stack_N_Trim format only!)
5. Option to append outfile.

### 7.1.5 Promoter.MAIN_Promoter_Point_Mapper

After aligning sequences with **Stacker_n_Trimmer** (although one sequence can also be mapped) and for each motif (entered manually or by MM-list) a complete pass is made for each data set once for all motifs (exactly like MotifMapper). Output is delivered to a sub-directory bearing the filename in folder "C:\PointePath\". We recommend that promoters are taken from a known position like the ATG encoding the first methionine of the respective protein, therefore "rooting" their data set to a fixed known biologically functional region. The entire sequence is virtually converted to a "hit = 1" or a "non-hit = 0" for each element, for the length of the element and added to the next sequence. Output files should be opened with a spreadsheet analysis program (not provided). The algorithm is originally written by Dr. Kurt Stüber (MPIZ 2002). His program, written in Perl, returns motifs as a single count for each at the 5-prime position, whereby the data is transformed (to account for overlapping elements) and outputted as PNG images with the distribution map of each element; raw data is also available.

Command Lines:

1. list of motifs (manual or list "..")
2. files to be read (MUST BE STACKED!!!)

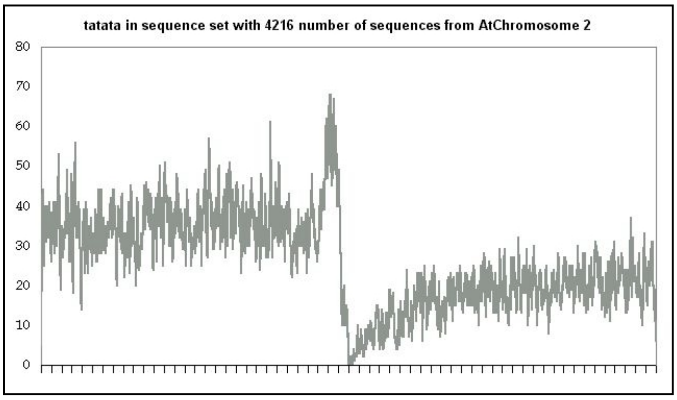


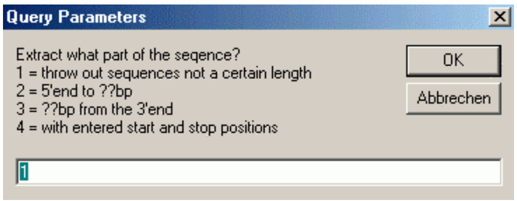
Figure 2. Promoter_Point_Mapper output viewed with a graph option from MS Excel. 1500bp upstream and 1499bp downstream of the "A" of annotated ATGs were extracted with aGenBankSQL, Stacked with Stack_N_Trim and computed with Promoter_Point_Mapper for the query "TATATA". Sequences are 5p to 3p with respect to the start codons.

### 7.1.6 Promoter.MAIN_Stacker_n_Trimmer

This program is designed to prepare sequence data for the PromoterPointMapper and the MaxtrixMachineSetAnalysis by trimming and aligning all sequences. Sequences can be aligned by a default 5-prime or 3-prime end, an inputted number of nucleotides away from the selected end, or both ends can be justified by entering the 5-prime and 3-prime position. Any sequences which do not qualify these three choices because they do not fulfil the minimal length are filled in on the side of justification with blank spaces. Finally, all sequences can be written to a single text file with a LF and CR separating each sequence from the next (in principle - simply removing the FASTA sequence identifiers. In other words, it puts individual sequences on a single line one after the other.

Command Lines:

1. Output option – StackNtrim file or a new FASTA file, input FASTA!
2. All sequences or from a list "//".
3. Option name append to outfile.
4. Type of extraction (shown below in Figure 3):



Extract what part of the seqence?
1 = throw out sequences not a certain length
2 = 5'end to ??bp
3 = ??bp from the 3'end
4 = with entered start and stop positions

1

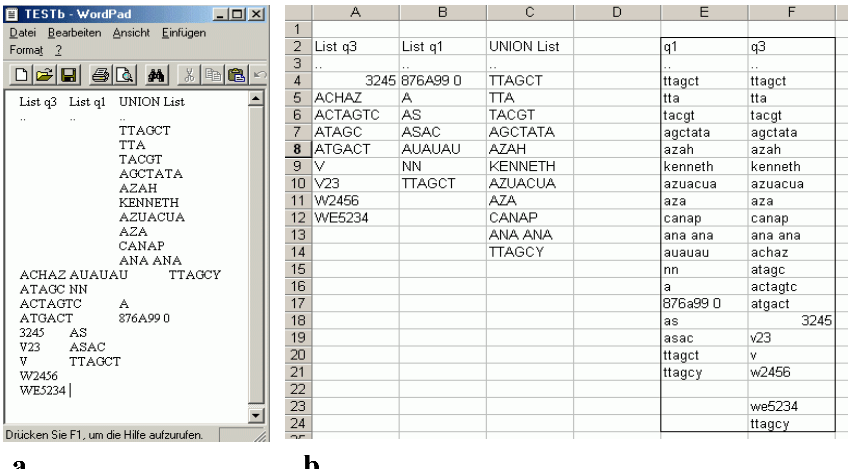
Figure 3. The Command Line prompt requesting the type of extraction that the user would like to perform.

### 7.1.7 SetGrouping

While the output from this program looks rather clumsy, it is important to point out what it is good for. This program compares two lists ("..") and returns either the UNION list (the list of all elements that are present in both lists) or ALL lists (that is, each list which is NOT present in the other list and the UNION list). It does not really matter what type of list it is. Blank spaces are removed from the outside of a **word** however. The ALL list function is quite useful <u>except</u> that redundant elements are not identified when the second list (actually the searched list) contains only one member and the first list has the element twice. While it is not explicitly clear from Figure 4 without this explanation, the list "q1" was entered first and the list "q3" was entered second. The word "ttgact" (the program can be set to case dependent) is clearly identified in the UNION list, but not deleted from the "q1" list. Actually, the word was deleted from both lists, but the second entry in list "q1" was tested against list "q3" after it had "lost" its only "TTGACT". Nevertheless, for non-redundant lists (or lists where it is prevalent) this small program offers a powerful way to rapidly compare large data sets which need to be crossed against each other.

Command Lines:

1. Output file name.
2. Case independence.
3. First List.
4. Second List.
5. Option [ALL] sets or [UNION] set only.



```
List q3    List q1    UNION List
..         ..         ..
                      TTAGCT
                      TTA
                      TACGT
                      AGCTATA
                      AZAH
                      KENNETH
                      AZUACUA
                      AZA
                      CANAP
                      ANA ANA
ACHAZ AUAUAU          TTAGCY
ATAGC NN
ACTAGTC               A
ATGACT                876A99 0
3245       AS
V23        ASAC
V          TTAGCT
W2456
WE5234
```

|  | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 |  |  |  |  |  |  |
| 2 | List q3 | List q1 | UNION List |  | q1 | q3 |
| 3 | .. | .. | .. |  | .. | .. |
| 4 | 3245 | 876A99 0 | TTAGCT |  | ttagct | ttagct |
| 5 | ACHAZ | A | TTA |  | tta | tta |
| 6 | ACTAGTC | AS | TACGT |  | tacgt | tacgt |
| 7 | ATAGC | ASAC | AGCTATA |  | agctata | agctata |
| 8 | ATGACT | AUAUAU | AZAH |  | azah | azah |
| 9 | V | NN | KENNETH |  | kenneth | kenneth |
| 10 | V23 | TTAGCT | AZUACUA |  | azuacua | azuacua |
| 11 | W2456 |  | AZA |  | aza | aza |
| 12 | WE5234 |  | CANAP |  | canap | canap |
| 13 |  |  | ANA ANA |  | ana ana | ana ana |
| 14 |  |  | TTAGCY |  | auauau | achaz |
| 15 |  |  |  |  | nn | atagc |
| 16 |  |  |  |  | a | actagtc |
| 17 |  |  |  |  | 876a99 0 | atgact |
| 18 |  |  |  |  | as | 3245 |
| 19 |  |  |  |  | asac | v23 |
| 20 |  |  |  |  | ttagct | v |
| 21 |  |  |  |  | ttagcy | w2456 |
| 22 |  |  |  |  |  |  |
| 23 |  |  |  |  |  | we5234 |
| 24 |  |  |  |  |  | ttagcy |

Figure 5. Output from SetGrouping showing two difficult lists of members to compare. Please note that any type of list can be sorted. a) is the raw output from SetGrouping using the ALL option. b) the output from (a) "cleaned up" for the two list unique members by sorting the output with MS Excels automatic sorting function. The two unsorted lists are contained within the box. However, each list must be an individual textfile in order to run SetGrouping (shown here only for information).

### 7.1.8 SeqKoncat

This module is intended for large probability mapping projects in which various files that are in one folder (say a list of BACs that are in FASTA format) that would be useful if they were together as one sequence for analysis in one text file. This program requires two parameters, a common string in the file names (or one then it is unique!) and the file name extension. If no filename is given the all files bearing the file-type extension will attempted to be concatenated. The sequences must be in FASTA format for **SeqFASTACompile**. The program inserts a single blank space between each sequence and writes a companion outfile indicating how many sequences were read and how many bytes were added to the sequence outfile critical for accurate calculation of **Obs.p**. A list of sequence names is not given however. Outfiles are written into "C:\Conkat\". The other version allows the fusion GenBank formats, clearing all non-sequence information. In addition, a small program called **TxtFileFuse** fuses textfiles together without modification and writes a companion file listing the files that were fused. TextFileFuse will fuse any group of text files, no handling occurs, the user gets simply one file with all files fused.

#### 7.1.8.1 Command Lines for SeqFASTACompile

1. Output file name is requested.
2. A common string name in files to be concatenated. If left blank then all FASTA sequences from each file found in the given folder will be put together in one file as described above.
3. A common file type extension is requested.
4. The folder is given by typing the path in to the folder of choice.

#### 7.1.8.2 Command Lines for SeqGBCompile

(remember that the GenBank files must be in text format)

1. The folder is given by typing the path in to the folder of choice.
2. Output file name is requested.
3. Common string in files to be concatenated is requested concatenated (if left blank then all GenBank sequences in the given folder will be put together in one file). If only one file is entered (i.e. the name is completely specific, then the extracted sequence will be properly extracted and useful for the aGenBankSQL.
4. A common file type extension is requested.
5. A text-line identifier that indicates where the sequence <u>begins</u> on the next line. The identifier is <u>on the line before</u> the sequence. By default "ORIGIN " is given. ALL non-IUPAC accepted characters are removed (thus making a cleared sequence!).
6. A text-line identifier indicates where the sequence <u>ends</u>! Typically the "//" is used as a termination signal.

#### 7.1.8.3 Command Lines for TxtFileFuse

1. Folder which contains the files to be fused (note only one folder is allowed).
2. Output file name.
3. Common string in file names (if left blank then all files in the given folder will be put together in one file).
4. Common file extension.

### 7.1.9 MMGraphicP

This program is not designed for screening, but as a rapid way to get your cis-elements in a presentation. This program accepts FASTA formatted files (more than one can be opened). Any type of motif can be entered excluding dyads, but IUPAC characters are accepted. The reverse complement is automatically generated and mapped as well. The final image (presented in Figure 6) can be copied anywhere as the image is part of a new MS Word document. I don't recommend doing more than 30 sequences at a time since you will end up making 30 new Word documents (but this is only my own personal opinion). In Figure 5 a promoter from 5prime to 3prime is presented with boxes indicated random motifs I entered. A maximum of 16 motifs can be entered simultaneously. If you need more, please email me or download the MMGraphicP module and adjust the colour limitations.

Command Lines:

1. Motifs (manual or list)
2. FASTA files to be inputted.

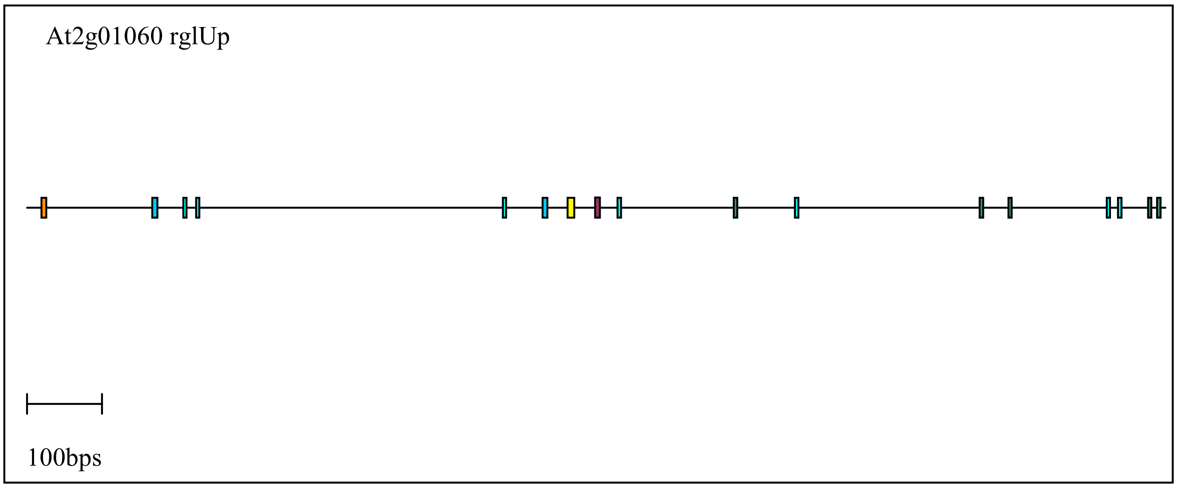


Figure 6. Example image of MMGraphicP output excluding the motif boxes which would be below the graphic and color coded as expected (actual image is larger).

## 7.2 Accessory Programs

MotifList_Generators**.** All output files are written to folder C:\MotifLists\.

### 7.2.1 AllCombinations

Input a word (any set of characters including internal blank spaces) and all possible combinations excluding any reverse complement are returned without redundancy as a MM-list. Maximum length is set to 15 but beware! The total number possible can be astounding. This N-ary recursion algorithm is adapted from http://www.case.monash.edu.au/~lloyd/tildeAlgDS/Recn/Perm.html .

### 7.2.2 AllOligos

Asks for a length and all possible combinations are returned.

### 7.2.3 ClearRedundancy

An MM-list can be cleared of all redundant entries.

### 7.2.4 ExpectedFrequencies

The expected frequencies of motifs (manual or list) are returned with a given **GC content**. This is particularly useful when a run is made with different species and the expected frequencies need to be recalculated (IUPAC probabilities are also a handled).

### 7.2.5 IUPAC_converter

Input a motif and all possible elements are written without redundancy, including the reverse-complement as a MM-list.

**The standardized IUAPC Ambigous nucleotide codes recognized by Motif Mapper.**

| | |
|---|---|
| A; T; G; C | nucleotide |
| R = A or G | puRines |
| Y = C or T | prYimidines |
| W = A or T | Weak hydrogen bonds |
| S = G or C | Strong hydrogen bonds |
| M = A or C | aMino group at common position |
| K = G or T | Keto group at common position |
| H = A, C, or T | not G |
| B = G, C, or T | not A |
| V = G, A, or C | not T |
| D = G, A, or T | not C |
| N = G, A, T, or C | aNy |

### 7.2.6 List_w_RevComps

Returns a list and its reverse compliments without redundancy.

### 7.2.7 PalindromeList

Multiple lengths can be entered and a single file is written having all palindromes for each length in MM-list format.

MM_Tools**.** All output files are written to folder C:\SeqFiles\.

### 7.2.8 ClearSeq

This program clears an entire text file of every character that does not conform to IUPAC characters (see below), returning a cleared sequence under the “C:/SeqFiles” directory folder.

### 7.2.9 **Count_txt_files**

A small application to count the number of A, T, C, G, N and blank spaces in a textfile. Useful for checking the work of experimental measures.

### 7.2.10 **Extract_Indv_seq_FASTA**

This program splits up FASTA formatted sequence files into a unique file for each sequence identified. Output files are delivered into the folder "C:\SeqFiles\" without further sub-directories. If only a subset of sequence files are desired and the exact name is known (case independent), then a list of sequences to be returned can be defined by a list preceded by "//" (each sequence name must be on a separate line).

### 7.2.11 Extract_Single_txt_Track

Just in case you get a whim and want to extract a region from file (for example a portion of sequence) then the position where to begin and end has only to be given and a new text file is written containing all byte information within this region.

### 7.2.12 Line_Grep

This small program allows the user to see each line of a text file line by line starting from the beginning of the file.

### 7.2.13 NamesFrom_FASTA

This extracts the names from a FASTA file and returns them as a list in a new file. Very useful when you get stuck with output files that are only in FASTA without a summary and you just need a name list.

### 7.2.14 BONUS PROGRAM : MickeydaExonIntron

This program is great for a quick drawing of exon-intron structure to scale on one MS Word document. If one wants to compare many genes, one must check for the scale bar length and proportionally correct all images. The input must maintain the format explained below. That is, there must be a beginning open parenthesis and at the end a closing parenthesis, exons are represented as two numbers separated by two dots and the next intron by one comma. The number of where something starts and ends is relative.

Command Lines:

1. Gene length, i.e. "(6390..11055)"
2. CDS, i.e. "(6995..7118,7263..7350,7794..7965,8244..8380,8450..8555,8765..8854,9034..9174,9337..9480,9770..9940,10325..10390,10476..10550,10741..10860,10975..11055)"
3. Option, name for gene.

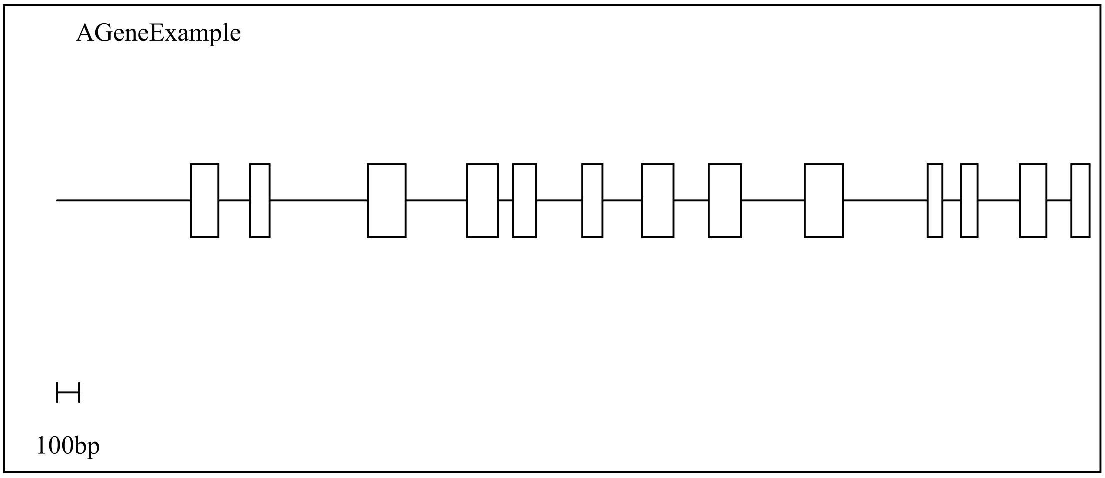


Figure 8. Example output of MickydaExonIntron using the coordinates given in the example presented in the text (actual image is larger).