Useful Microsoft® Word™ Macros for Molecular Biologists and Protein Chemists

BioTechniques 28:1198-1201 (June 2000)

Gerry Shaw

University of Florida, Gainesville, FL, USA

ABSTRACT

Biologists today make extensive use of word processing programs for the production of research reports, literature reviews and grant proposals. Frequently, such programs become the default platform for viewing and the later publication of protein and nucleic acid sequence data. Thus, researchers often switch between their word processor and more specialized programs designed to analyze protein and nucleic acid sequences. It would be more convenient to perform these simple sequence analyses using the word processor without switching to another program. The focus here is on the use of the Visual Basic® programming language, which is built into all recent versions of Microsoft® Word® to generate surprisingly complex and useful macros that can conveniently analyze several important features of protein and nucleic acid sequences. The standard Word interface can also be easily modified to display and run these macros from a pull-down menu. Several examples of this approach are provided.

INTRODUCTION

Most biological researchers constantly switch between word processors and the more specialized programs that are designed for various types of protein and nucleic sequence analysis. It would be more convenient and efficient to perform simple types of protein and nucleic acid sequence analysis without having to leave the word processing program. As demonstrated here, it is relatively easy to customize standard versions of Microsoft® Word™ versions 6.0, 97 or 2000 to perform a variety of useful manipulations that would normally require a separate suite of programs. These Word versions include Visual Basic®, a powerful programming language that allows the production of surprisingly complex and useful macros.

Several examples of such macros, of particular use to protein chemists and molecular biologists, are given here. These macros work equally well on Macintosh computers, IBM PCs and compatibles. They can also be used in Microsoft Office® programs such as Excel® and PowerPoint®, which also support the Visual Basic programming language.

MATERIALS AND METHODS

Microsoft Word version 6.0 was used initially, and work was continued using Visual Basic programming with Word 97 and Word 2000 when they became available. The Visual Basic macros described here run identically on all three versions; the compiler is built into these versions, and the macros can be constructed with no additional purchases. Releases of Word before version 6.0 use Word Basic, a now obsolete macro language. Although the language is clearly related to Visual Basic, it can be regarded as a distinct and incompatible programming language, and the macros described here must be modified to run on such older versions. However, versions of some of the programs listed here have been written in Word Basic and are available from the author.

RESULTS

A simple example of a useful Visual Basic macro is shown here. Scientists frequently need to know the molecular weight of a sequence or part of a sequence. It would be convenient if the user could select a segment of nucleic acid or protein sequence with the mouse and press a button, tab or hotkey to determine the exact calculated molecular weight of the selection. Table 1 shows a simple Visual Basic macro that performs this function for DNA sequences and is called NucWeight.

Even those with limited programming experience can probably follow this program. Visual Basic programs start with **Sub**, followed by the name of the macro (**Nucweight()** in this example), and end with **End Sub**. Lines beginning with a single apostrophe (') are remarks, with text included for those who wish to understand or modify the program. In the example, the lines beginning with an apostrophe give information

Table 1. Macro

```
Sub Nucweight()
'
'Nucweight Macro
'Macro recorded 07/02/99 by Gerry Shaw
'This version for DNA sequences: for RNA MW for A =
  329.2, U = 306.1, G = 345.2, C= 305.2
'

  X = Len(Selection.Text)
  For Y = 1 To X
    Select Case Mid$(Selection.Text, Y, 1)
      Case "A"
        MW = MW + 313.2
      Case "T"
        MW = MW + 304.2
      Case "G"
        MW = MW + 329.2
      Case "C"
        MW = MW + 289.2
      Case Else
        Z = Z + 1
    End Select
  Next
  MW = MW + 18
  If (MW > 18) Then MsgBox ("Selection includes " &
X - Z & " bases, Molecular Weight= " & MW &
Daltons")
  If (MW = 18) Then MsgBox ("No sequence selected")
End Sub
```

about when the program was constructed and by whom. They also give information indicating how to modify the program to read RNA sequences. Because such text is completely ignored by the Visual Basic compiler, it can have any desired content.

The first significant line of computer code in the program is **X = Len(Selection.Text)**, which simply finds the length of the selected text and assigns this number to the variable X. (In all cases, the user will select text using the computer mouse.) The code **For Y = 1 To X** is a loop that starts at 1 and ends at X, the number of characters selected by the mouse. The line **Select Case Mid\$(Selection.Text, Y, 1)** simply takes one character from the selected text each time the loop is executed, starting with the first character and going sequentially to the end of the selection.

The following lines use the **Case** command to compare each selected character with A, T, G or C. **Case A** means that if the character extracted from the selected text is A, then the following lines of code up to the next **Case** are executed. Here, 313.2 Da, the molecular weight of adenine deoxyribose monophosphate minus the weight of one water molecule, is added to **MW**, a variable that stores the molecular weight of the selected sequence. The following lines add the appropriate molecular weight amount in exactly the same way if the selection contains T, G or C bases.

The final **Case** entry, **Case Else**, deals with characters in the selected text that are not A, T, G or C. If a character in the selected text is not A, T, G or C, the code **Z = Z + 1** increases the variable **Z** by one. Since **X** is the total length of the selected text and **Z** is the number of non A, T, G and C characters, **X-Z** corresponds to the number of bases in the selected text, a calculation actually performed later on.

The lines **End Select** end the part of the macro related to the **Select Case** function, and **Next** indicates the end of the **For** loop. The molecular weights used in this part of the program are bases within DNA sequences bound to deoxyribose and phosphate and lack one water molecule when compared to base deoxyribose monophosphates free in solution. However a nucleic acid sequence has a free N- and C-terminus that, together, contain one water molecule. Therefore, the weight of one water molecule must be added to any nucleic acid sequence to get the correct molecular weight. For this reason, the line **MW = MW + 18** is added.

Finally, the results are printed out using the message box function, **If (MW > 18) then MsgBox ("Selection includes " & X-Z & " bases, Molecular Weight= " & MW & "Daltons")**. This line prints out the number of bases and the calculated molecular weight of the selected sequence, if at least one A, T, G or C has been detected. The format for the **MsgBox()** function is simply that text to be printed out is in quotation marks, variables are entered directly and the two types of information are separated by the **&** character. The next line, **If (MW = 18) Then MsgBox ("No sequence selected")**, prints out the message that no sequence has been selected if either no text was selected or if the selected text contains no A, T, G or C characters. In either case, **MW** remains at 18.

The final line, **End Sub**, indicates the end of the source code for this macro. This simple program completely ignores any numbers, spaces, table or other formatting characters in the selected text, and in fact, ignores all characters except uppercase A, T, G and C. Database entries or raw sequence data can therefore be examined without reformatting to remove numbers and spaces. To make the program read lowercase characters, simply change the case of the letters searched for by the **Case** function. Alternately, Word has a command in the Format menu that can change the case of selected text from lower to uppercase. To make the macro determine the molecular weight of RNA sequences, it is only necessary to add 16 to the relevant molecular weights associated with A, G and C to account for the extra oxygen atom, to substitute **Case "U"** for **Case "T"** and finally, to enter 306.1 for the molecular weight of each U base.

To incorporate this program into your personal Word program, go to the Tools menu, select the Macro option and then Macros from the next menu. When another menu appears, type Nucweight (or whatever name you choose) in the Macro Name box. Press the Create button, which takes you to the built-in Visual Basic editor/compiler. Now, either type in the program or paste it in if you obtained it electronically from the Internet so that the lines not starting with the apostrophe are the same as those shown in Figure 1.

After checking the text, go back to Word, open a nucleic acid sequence and select part of it with your mouse. Now go back to the Visual Basic editor and run the macro by pressing

the right arrowhead symbol. If the program has been entered correctly, a window displaying the number of bases and the molecular weight of the selected sequence should appear, similar to the one shown for a different macro in Figure 1. If there is a mistake in the program code, the Visual Basic compiler will highlight the line containing the error, which can then be corrected. After exiting the Visual Basic compiler, the program can be run at any time by going to the Tools menu, selecting Macro and from there, selecting Macros. Next, select Nucweight from the list and press the run button.

Though somewhat inconvenient because it requires several keystrokes to activate the macro, this procedure is workable. However, it is much more convenient to modify the menus of Word to include a new pull-down menu for this and other macros. To do this on an IBM PC running Word, select Tools, then Customize and then the Commands tab. Finally, select New Menu, and the text New Menu will appear to the right of the selection box. This object can be selected with the mouse and dragged to the desired place on the Word interface where it will appear under the name New Menu. You can change the name by pressing the Modify Selection button and selecting the New Menu name in the properties window with the right mouse button.

As Figure 1 shows, I made a new menu item called MolBio and placed it next to the Help menu, which also includes tabs for the other macros described later. On a Macintosh computer, the procedure is essentially the same except that the New Menu item is added to the various icons below the pull-down menus, rather than among the pull-down menus as shown in Figure 1. To add individual macros to the MolBio menu, go to Tools, select Customize and then Select Macros from the Commands menu. A list of macros will appear and those selected can be dragged and dropped to the MolBio menu.

The macros may have longer names than are convenient for your use—the Nucweight macro was listed as Normal.Autoexec.Nucweight on my PC. However, when this macro is dragged to the MolBio menu, you can select the Modify selection tab and use the right mouse button to change the macro name to Nucweight.

The source codes for this and several other macros can be downloaded from the software library at www.biotechniques.com/ or from www.ufbi.ufl.edu/~shaw/molbio.htm. These can be loaded as a text file into Word, copied and simply pasted into the Visual Basic editor to simultaneously generate all of these macros. The macros can then be incorporated into the menu system as described above.

The first of these other macros, Aacomp, counts the number of each amino acid in a selected sequence and prints out the molar percentage of each. The Decode macro translates the first 24 bases of a selected nucleic acid sequence into the corresponding 7–8 amino acids of protein sequence in all

Figure 1. Example of a Visual Basic macro at work in a customized version of Word. The top shows part of the Microsoft Word interface with a new menu item, MolBio, included as described in the text. This menu contains all of the described macros arranged in alphabetical order. In this case, a peptide sequence has been selected with the mouse (outlined in black), and the Proweight macro has also been selected. The lower part of the figure shows the result of this selection, indicating the number of amino acids selected and the exact molecular weight of this sequence.

three forward reading frames. It can be easily modified to translate longer regions of nucleic acid sequence, simply by increasing the size of the **outputarray(24)** function (details are included as comments in the appropriate file).

A macro called IEP determines the isoelectric point of a peptide or entire protein selected using the mouse. This program works by first counting how many potentially ionizable (proton-accepting) amino acid residues are present in the selected text and then calculating the total charge of the sequence at pH 2.0. If the charge is positive, as will be the case for almost all peptides, the program then calculates the total charge at pH 2.1, then pH 2.2 and so on, until the charge becomes either zero or negative. The pH at which this occurs is printed on the screen and is a good estimate of the isoelectric point of the sequence. This information predicts where a particular protein will appear on 2-D gels and when it will elute from ion exchange columns.

The Protweight macro is similar to the Nucweight macro described in detail earlier, but it calculates the molecular weight of selected protein sequence, and is shown in operation in Figure 1. The program is useful for rapidly determining the expected SDS-PAGE mobility of recombinant proteins and protein fragments and is particularly useful when studying results from mass spectroscopy experiments. The Reformat macro will remove all numbers, spaces, periods, hyphens and tab- and line-feed characters from a selected sequence. It is useful for preparing database sequences for manipulation in Word.

The Reverse macro will print the antisense sequence of the 3' 30 bases of selected nucleic acid sequence on the screen. It is printed in the 5' to 3' orientation and is useful for the construction of 3' oligonucleotide primers for PCR and other purposes. The Signature macro simply prints out the author's name, affiliation, address, telephone, fax and e-mail information, saving much time when writing letters.

Finally the Tm macro calculates the length, GC content and melting temperature of selected nucleotide sequences. The program makes use of two different methods for calculating the melting temperature (6). The first is appropriate for longer nucleic acid sequences and is from the work of McCarthy et al. (4); the second, from Itakura and co-workers (7), is appropriate for sequences up to 18 bases. Selecting which method to use is determined by the number of bases in the selected nucleic acid sequence. This macro allows the design of compatible oligonucleotides for sequencing and PCR applications.

DISCUSSION

The methods presented here are simple examples of what can be accomplished using Visual Basic combined with Microsoft Word. Obviously, with an understanding of the Visual Basic programming language, these macros can be modified to perform other useful functions, and macros for other purposes can also be designed. This approach has several obvious advantages over other methods. First, it is free for the majority of us who already have Microsoft Word or a similar program. Second, with a little knowledge of Visual Basic, the user can see exactly what the program is doing and easily modify it—

not usually the case with commercial packages. Third, the macros run in Microsoft Word exactly the same on IBM, IBM-compatible PCs and Macintosh computers, so that they should be useful for collaborations or teaching across these two platforms. The macros described here can be downloaded from the software library at www.biotechniques.com/ or from www.ufbi.ufl.edu/~shaw/molbio.htm.

Because the entire suite of the most recent Microsoft Office programs uses the Visual Basic programming language, it is also possible to use similar macros for Excel, PowerPoint and other Microsoft Office programs. With more sophisticated programming, it should be possible to automatically generate graphs and figures from sequence data in Word documents. In fact, previous papers show how it is possible to use Microsoft Excel to perform quite complex types of protein and nucleic acid sequence analysis and plot out the results (1,2,4,5). Hopefully, this article will stimulate others to create and make available programs that use Microsoft Office's versatile suite of programs.

ACKNOWLEDGMENT

This work was supported in part by a grant from the American Heart Association.

REFERENCES

1. **Anbazzhagan, R. and E. Gabrielson.** 1999. Spreadsheet-based program for alignment of overlapping DNA sequences. *BioTechniques* 26:1180-1185.
2. **Han, B. and A.H. Tashjian, Jr.** 1998. User-friendly and versatile software for analysis of protein hydrophobicity. *BioTechniques* 25:256-263.
3. **McConaughy, B.L., C.D. Laird and B.J. McCarthy.** 1969. Nucleic acid reassociation in formamide. *Biochemistry* 8:3289-3295.
4. **McEwan, N.R. and D. Gatherer.** 1998. Adaptation of standard spreadsheet software for the analysis of DNA sequences. *BioTechniques* 24:131-138.
5. **Shaw, G.** 1996. Protein sequence interpretation using a spreadsheet program. *BioTechniques* 19:978-983.
6. **Wahl, G.M., S.L. Berger and A.R. Kimmel.** 1987. Molecular hybridization of immobilized nucleic acids: theoretical concepts and practical considerations. *Methods Enzymol.* 152:399-407.
7. **Wallace, R. B., J. Shaffer, R. F. Murphy, J. Bonner, T. Hirose and K. Itakura.** 1979. Hybridization of synthetic oligodeoxyribonucleotides to phi chi 174 DNA: the effect of single base pair mismatch. *Nucleic Acids Res.* 6:3543-3557.

Received 7 September 1999; accepted 7 February 2000.

Address correspondence to:

Dr. Gerry Shaw
University of Florida College of Medicine
Department of Neuroscience
The University of Florida Brain Institute
JHMC Box J-100244, Gainesville,
FL 32610, USA
Internet: shaw@ufbi.ufl.edu