Supported in part by research grants from the United States Public Health Service R01-DK-34281 (M.A.L.), a Clinical Associate Physician Award (RR-00722-2251) from the National Institutes of Health (W.F.S.), NIH, NCCR, OPD-GCRC Grant No. RR00052 and a grant from the Interthry Research Foundation (MS). This research was performed while Dr. Ringel was a Pfizer Postdoctoral Fellow. Address correspondence to Dr. Michael A. Levine, The Johns Hopkins University School of Medicine, 720 Rutland Street, 863 Ross Building, Baltimore, MD 21205, USA. Internet: mlevine@welchlink.welch.jhu.edu

Received 10 March 1997; accepted 29 July 1997.

M.D. Ringel, W.F. Schwindinger, M. Saji, M.A. Zeiger and M.A. Levine

*Johns Hopkins University
School of Medicine
Baltimore, MD, USA*

Adaptation of Standard Spreadsheet Software for the Analysis of DNA Sequences

BioTechniques 23:131-138 (January 1998)

ABSTRACT

This paper presents use of spreadsheet software to derive various statistics on nucleotide and codon distribution and frequency from gene sequence data, including χ^2 tests and their derivatives, and proportion of nucleotides in the third position. The basic principles can be easily extended to other more complex functions. In addition, it can be used to translate a nucleic acid sequence into a protein sequence or to produce a codon usage table. This adaptation permits sequence analysis without expensive dedicated software and easy modification of the analysis according to the user's requirements. It is also ideal for introducing biological science students to the programming potential of spreadsheets.

INTRODUCTION

The ever increasing number of DNA sequences being deposited in various databases necessitates the easy availability of simple computer programs for sequence analysis. Frequently, a problem arises in finding a program able to perform the exact type of analysis required by the experimenter. To most molecular biologists, untrained in computer programming languages, the prospect of having to write a new program or even alter an existing one is a daunting task. A solution to this problem may lie in use of the common spreadsheet program such as Microsoft® Excel®, as suggested by Shaw (9), whose original paper adapted Excel for the analysis of protein sequences. Here, we present an Excel adaptation that can be used for the analysis of DNA sequences, generating a range of statistics on nucleotide and codon usage, in addition to performing translation. Further modification of the program to perform more complex functions requires only a familiarity with Excel.

MATERIALS AND METHODS

Computer Programs Used

The programs used in this work are those found already installed in a wide range of IBM®-compatible personal computers—Netscape to allow access to the Internet, Microsoft Word™ Version 6.0 (Microsoft, Redmond, WA, USA) as a word-processing program and Microsoft Excel Version 5.0 as a spreadsheet program. However, other Internet viewers, spreadsheet packages and word-processing documents can be used instead.

Source of DNA Sequences

DNA sequences for input can be accessed and downloaded directly from databases available on the Internet or can be copied directly from sequence information generated by an automated sequencer. DNA sequence information for testing the program was accessed from the GenBank® database (<http://www2.ncbi.nlm.nih.gov/cgi-bin/genbank>). Sequences from GenBank

Short Technical Reports

were accessed using Netscape and copied into a word-processing document (Windows® Version 6.0 of Microsoft Word) by “Copy”ing and “Paste”ing.

Formatting the Sequence for Analysis into a Format for the Excel Spreadsheet

Once in the Word document, all numbers can be removed by selecting the *Replace* function within the *Edit* menu on the toolbar. This was done by selecting the *Any Digit* option from the *Special* option in the *Replace* function, while leaving the *Replace With* option blank, followed by selecting the *Replace All* button. All spaces within the text were removed in a similar manner. For GenBank sequences downloaded in their entirety as Word files, all the descriptive information must also be removed. In addition, all *Paragraph Mark* symbols at the end of lines should be removed, by using the *Special* button within the *Replace* dialogue box. All untranslated 5' and 3' sequences must also be excised. Most GenBank entries give details as to the precise locations of the translational start and stop codons and also splice junctions where relevant.

A comma should be inserted before each nucleotide by using the *Replace* function within the *Edit* menu on the toolbar to replace 'A' with 'A,' etc. This should result in the type of appearance shown in Figure 1. The sequence is then put into a table format with three letters per row. This is done by highlighting the text using the mouse and selecting the *Convert Text to Table* option from the *Table* option on the menu bar. From this menu, set the number of letters per row to three by typing “3” in the *Number of Columns* box. At this point, the *Separate Text* indicator should be on *Commas*, but if necessary,

this should be selected. Then select *OK*. The resulting 3-column table can be highlighted, copied and pasted directly into Excel. Figure 2 gives an example. The 3-column table is the preferred format for those calculations, which are position-specific, e.g., the %A at the first position, etc. as described below. For this type of calculation, the material is ready to be pasted into the spreadsheet program.

However, for calculation of codon usage and derivatives of these, the three columns must be converted to a single-column table. This may be done by highlighting the table and selecting the *Convert Table to Text* function from the *Table* menu. The *Other* option should be selected, and in the box to the right of this, enter a space. Selecting *OK* will produce text where there are three letters per row, and they are separated by spaces. The spaces should be removed by the *Replace* function as already described. This will result in the sequence taking the form as shown in Figure 3.

EXAMPLES OF POTENTIAL APPLICATIONS

Generally, we have found it best to insert descriptive information into the first row (cells A1, B1, C1 etc.) and enter all sequence information below commencing in the second row (cells A2, B2, C2 etc.). In addition, all calculations are based on a sequence of 1000 codons (finishing in row 1001), but altering the range of the number of codons counted can be done by changing the equation to count either a greater or lesser number of cells. This principle can also be applied to all of the applications listed below.

Translating DNA Sequences

This uses a translation table that is

entered in the spreadsheet as two adjacent columns, e.g., columns R and S. Column R contains the 64 different trinucleotide sequences (e.g., TTT, TTC etc.). Column S contains the appropriate amino acid encoding this sequence (e.g., Phe, Leu etc. or K, L etc.). The sequence, in single-column format (see Figure 3), is entered on the left of the spreadsheet, e.g., in column A. The other two alternate reading frames can be calculated at this time. This is done by removing the first nucleotide of the original DNA sequence for frame two or the first two nucleotides for frame three. The truncated sequences should be treated in the manner described above and can be inserted into columns B and C, respectively. The three reading frames can be translated in columns E, F and G, respectively. If the first triplet of the first reading frame is in cell A2, then the command to translate it in cell E2 is:

```
=VLOOKUP(A2,$R$2:$S$65,2, FALSE)
```

where the translation table covers rows 2–65 in columns R and S. This command should then be copied in all cells in column E:

```
=VLOOKUP(A3,$R$2:$S$65,2,FALSE)  
=VLOOKUP(A4,$R$2:$S$65,2,FALSE)  
=VLOOKUP(A5,$R$2:$S$65,2,FALSE)
```

and column E then copied in its entirety into columns F and G:

```
=VLOOKUP(B2,$R$2:$S$65,2,FALSE)  
=VLOOKUP(B3,$R$2:$S$65,2,FALSE)  
=VLOOKUP(B4,$R$2:$S$65,2,FALSE)  
:  
:  
:  
=VLOOKUP(C2,$R$2:$S$65,2,FALSE)  
=VLOOKUP(C3,$R$2:$S$65,2,FALSE)  
=VLOOKUP(C4,$R$2:$S$65,2,FALSE)
```

This results in the contents of the cell of interest from column A being compared with the 64 cells in column R.

at,ggg,act,ggagg,acgagc,aaaag,atgctt,accga,atc,ggag,atc,ctgagg,aggagg,gaagagg,gaag,
g,agga,attag,ggal,ccc,ctaac,acagtg,agag,agc,aatgc,gag,cagt,ggag,aaa,tg,tgtaa,agg,ccc,
ggg,agc,ggc,tagag,ctct,gatg,agc,gig,attc,ctct,cgat,cac,atac,aga,agg,agatt,gc,acgg,agg,
g,ctct,tgact,ctt,gcat,gc,gagg,gacc,attg,cgt,ggccc,cac,aa,ctct,tta,aca,act,tg,aaa,taa.

Figure 1. A typical example of the first 275 nucleotides of a DNA sequence after it has been edited to remove all numbers, and a comma has been inserted in front of each letter.

When an identical match is made between two cells, then the corresponding cell in column E is assigned the value of the cell in column S. This allows a sequence to be translated. In the case of translations being performed for all three reading frames, this takes the form shown in Figure 4.

Counting Stop Codons

In an appropriate cell, for instance I2, enter the command:
=COUNTIF(\$E\$2:\$E\$1001,"stop")
This can be "pasted" into cells J2 and K2 to display the number of stop codons per reading frame. It will check each of the first 1000 codons to look for STOP codons (TAG, TGA or TAA).

The first reading frame should have a single stop codon, but several may appear in the two others. This step is non-essential, but it may give an indication of any errors that have occurred in the process of copying and pasting the data into Excel.

Creating a Codon Usage Table

The codon usage table is an essential element in the analysis of codon bias (8). This method uses the single-column format (Figure 3). The frequency of each codon in the open reading frame (ORF) (and the others if necessary) can be calculated using the COUNTIF function, and the codon usage table displayed in columns M and N as shown below:

a	t	g
g	g	a
c	t	g
g	a	g
g	a	c
g	a	g
c	a	a
a	a	g
a	t	g
c	t	t
a	c	c
g	a	a
t	c	c
g	g	a
g	a	t
c	c	t
g	a	g
g	a	g
g	a	a
g	a	g
g	a	a
g	a	g
g	a	g
g	a	a

Figure 2. The first 25 codons of the sequence shown in Figure 1, after the *Convert Text to Table* function has generated a 3-column table. This may be copied directly into Excel and is the preferred format for position-specific calculations, e.g., % C in the 2nd position, etc.

aga
gga
cga
gaa
gac
gag
caa
cat
caa
caa

Figure 3. The first 10 codons of the same sequence after creating single columns of three letters. This is the preferred format for entry to Excel for calculations of codon usage and derivatives thereof.

Short Technical Reports

Cell	Entry
M2	AAA
N2	=COUNTIF(\$A\$2:\$A\$1001,M2)
M3	AAC
N3	=COUNTIF(\$A\$2:\$A\$1001,M3)

Completing the codon set in column M and copying the above commands in column N generates a codon usage table for the first reading frame, for the first 1000 codons. Triplet usage tables for other reading frames can be generated by copying column N in its entirety into columns O and P and modifying it accordingly. For example, in cell P2, the command used would read:

=COUNTIF(\$C\$2:\$C\$1001,M2) etc.

At first glance, this process may seem rather laborious, as it appears to require entering data in 128 different cells. However, it can be performed more rapidly in the case of column N by entering the value in cell N2 as described and then highlighting both it and the remaining cells to be filled. From the *Edit* option on the toolbar, select *Fill*, and from the submenu, select *Down*. This will complete the cells more rapidly.

Calculating the Frequency of Each Nucleotide in Each Position

There are two methods that can be used to calculate the frequency of a nucleotide in a given position. The strategy to be used depends on which type of further calculations will be performed.

Met	Trp	Gly
Gly	Asp	Thr
Leu	Trp	Gly
Glu	Arg	Gly
Asp	Thr	Arg
Glu	Ser	Ala
Gln	Lys	Lys
Lys	Arg	Asp
Met	Cys	Ala
Leu	Leu	Tyr
Thr	Pro	Arg
Glu	Asn	Ile
Ser	Pro	Arg
Gly	Glu	Arg
Asp	Ile	Ser

Figure 4. Translation of the first 15 codons of the above sequence in the 3 reading frames.

Calculating the frequency using 3-column calculations. The simplest method of performing this calculation is using 3-column sequence input (see Figure 2), and it can be entered in columns U, V and W. This is generally the format used where the result is an end in itself, and no additional calculations will be performed on the data.

Using this method, to calculate the frequency of adenine residues in the first position, enter the command:

=COUNTIF(U2:U1001,"a")/COUNTA(U2:U1001)

in a chosen cell. The COUNTIF numerator in this function counts the number of adenine residues in column U, and the COUNTA denominator counts all residues. To count the proportion of cytosine, the command is

=COUNTIF(U2:U1001,"c")/COUNTA(U2:U1001)

likewise repeated for guanine (g) and thymine (t). For the second position, the commands are the same except that the array U2:U1001 becomes V2:V1001 and likewise becomes W2:W1001 for the third position. This generates four values of nucleotide frequency for each position, and these can be combined to generate other statistics. For instance, the frequency of adenine and the frequency of guanine in the third position can be combined to generate the total purine frequency in the third position etc. The total frequency of nucleotides in the reading frame can be calculated by expanding the counted array. For instance, the command:

=COUNTIF(U2:W1001,"a")/COUNTA(U2:W1001)

gives the total frequency of adenine residues in all three positions.

GC3s is the total proportion of G and C in the third position (7) and can be calculated using the command:

=(COUNTIF(W2:W1001,"c")+COUNTIF(W2:W1001,"g"))/COUNTA(W2:W1001)

Calculating the frequency using single-column calculations. The method shown in the section above is the most simple means of calculating the frequency of nucleotides in a given position, but it is not easy to apply this method to some of the calculations described below. In the case of calculations used to work out further values, such as the expected frequency of a given codon (see *Calculating Expected Values of Codons*), it is better to use the method described in Figure 5. Briefly, this method sums the number of times a given codon occurs where the nucleotide of interest is contained in the position of interest. For example, the abundance of the A in position 1 would be the sum of all the codons that have A in their first position (i.e., AAA, AAC, AAG, AAT, ACA.....etc). To calculate the abundance of each nucleotide across the gene, this is just a simple matter of summing its occurrence in positions 1, 2 and 3. From here, to calculate the frequency of a nucleotide, it is simply a matter of dividing the total by the number of times that the nucleotide could possibly occur (e.g., for positional calculations, it would be a case of dividing the total number of A nucleotides in position 1, AXX, by the total number of nucleotides present in all of the first positions, the sum of AXX, CXX, GXX and TXX).

Cell	Entry	Comments
M67	=SUM(N2:N17)	Adds the number of "A" nucleotides in position 1
M68	=SUM(N18:N33)	Adds the number of "C" nucleotides in position 1
M75	N2+N6+N10+N14+N18+N22+N26+N30+N34+N38+N42+N46+N50+N54+N58+N62	Adds the number of "A" nucleotides in position 3

Figure 5. Some examples of the entries into cells as a means of calculating the abundance of each nucleotide at a given position.

Short Technical Reports

Calculating Expected Values of Codons

Derivation of codon bias indices requires that observed codon frequencies be compared with expected frequencies. The expected frequencies can be entered next to the observed, for instance, in column Q. There are three methods of deriving expected frequencies, as follows:

Elementary calculations based on assuming 25% frequency of each nucleotide. The simplest, if most naive, method of calculating expected values for each codon is to assume that each nucleotide occurs with a frequency of 0.25 [i.e., it assumes that the reading frame has DNA arranged in a random manner (5)]. In this case, there is an equal likelihood of each codon, and the expected value is therefore calculated by entering the command:

=SUM(N2:N65)/64

in all cells of column Q. Although this is the easiest method, it takes no account of the amino acid composition of the gene and that there can only be one stop codon per ORF.

Calculations based on the frequency of each nucleotide across the gene. If the frequency of each nucleotide is known, a value can be given for expected trinucleotide frequency based upon this. Figure 5 and the Section *Calculating the Frequency Using Single-Column Calculations* demonstrate the best method to perform the calculations to determine the frequency of each nucleotide.

Thus, the expected abundance of the trinucleotide ATG would be the frequency of A, multiplied by the frequency of T, multiplied by the frequency of G, multiplied by the total number of codons.

Calculations based on the amino acid frequency. The above method does not take account of the amino acid sequence. In the case of 2-fold synonymous groups such as lysine (AAA and AAG), the expected values would be calculated as follows (based on the functions described in the Section *Creating a Codon Usage Table*):

=M2/(M2+M4) and =M4/(M2+M4)

This assumes that the expected value of each of the two codons used is 50% of

the total value of number of times that the amino acid is present. These equations could be filled into the cells AA2 and AA4 for AAA and AAG, respectively, as described in the section above. In a similar way, the equation for the expected values for asparagine encoding codons would be:

=M3/(M3+M5) and =M5/(M3+M5)

and these would be inserted into cells AA3 and AA5 for AAC and AAT, respectively.

This is the method used for calculating expected codon frequency when deriving Scaled chi-square (10).

Calculations for a coding region on the basis of frequency of the third nucleotide. A more accurate method of calculating expected codon usage is to bias expected values to reflect the composition of the third nucleotide in all of the codons. For example, the expected values of AAA and AAG would now become:

Cell AA2 =(M2/(M2+M4))×(M75/(M75+M77))

Cell AA4 =(M4/(M2+M4))×(M77/(M75+M77))

where M2+M4 is the total number of lysine residues, and M75/(M75+M77) is the proportion of A nucleotides used in position 3, relative to the total codons using either A or G in position 3 (see Figure 5).

Other frequencies can be calculated in a similar manner (e.g., AAC):

Cell AA3 =(M2/(M2+M4))×(M76/(M76+M78))

Calculation of the Difference Between Observed and Expected Values and Chi Squares

Differences between expected and observed values can be calculated by inserting the following equations in column AB:

Cell AB2 =(M2-AA2)

Cell AB3 =(M3-AA3)

Cell AB4 =(M4-AA4)

Chi-squared values for each codon can be calculated by filling in the following equations to column AC:

Cell AC2 =((M2-AA2)^2)/AA2

Cell AC3 =((M3-AA3)^2)/AA3

Cell AC4 =((M4-AA4)^2)/AA4

The chi-squared total can be calculated by using the following equation:

Cell AC66 =SUM(AC2:AC65)

This is then divided by twice the total number of codons to generate Scaled chi-square (10), using for instance the command:

Cell AC67 =AC66/((COUNTA(A2:A1001))×2)

Calculating Frequency of Other Sequences

Recently, the importance of also studying sequences that span codons was highlighted by the observation of J.M. Smith and N.H. Smith (6) that the sequence AG-G (where “-” denotes the division between two consecutive codons) was selected against. Other selective constraints on sequences spanning two codons are at present unidentified. The sequence AG-G appears as a triplet in the second reading frame, so its frequency can be calculated by applying the techniques described above for calculating trinucleotide sequences listed in column A of the single-column calculations to column B trinucleotide sequences.

Microsoft Excel can be used to calculate the relative frequency of any given sequence within the gene, in any reading frame. The same method is used as for triplet sequences; however, the column number is varied, and there are more reading frames. For instance, when examining the frequency of occurrence of hexanucleotides, there are six reading frames, and a 6-column table must initially be generated. This can be compressed to a single column as above.

Educational Use of Spreadsheet Programs

The use of the personal computer (PC) as an educational tool in biological science courses is almost as old as the PC itself (2). Although customization of spreadsheet programs is not computer programming in the strict sense, a case can be made that it is a transferable skill in that science graduates will encounter spreadsheet programs and more general use of PCs in many of their subsequent careers. Pro-

Short Technical Reports

grams of the type presented here, in combination with access to sequence databases through Netscape, can be used to introduce undergraduates to basic keyboard and spreadsheet skills, to assist in Internet familiarization and to introduce various concepts in bioinformatics, such as the nucleotide and codon usage statistics described above. Once students are familiar with the mechanism by which the program produces third-nucleotide usage in codons (GC3s) (7) etc., they may be given the opportunity to derive other sequence-based statistics, such as effective codon number (11) or P2 (4). Students could quickly begin to adapt the programs to perform sequence analyses of their own choice, leading to what Belbin and Belbin (1) have described as "discovery learning", which has the advantage of greater long-term retention than more traditional teacher-centered delivery.

CONCLUSIONS

The methods described here can be used to allow relatively easy analysis of DNA sequences without the need for expensive commercial software. They are relatively simple to design and can be customized to the requirements of a single experimenter, without the need for any computer programming per se. In addition to the simple nucleotide and chi-squared calculations described here, we have used this type of spreadsheet for analyzing DNA for a number of more complex statistics, e.g., effective codon number (11) and intrinsic codon deviation index (3).

REFERENCES

1. **Belbin, E. and M. Belbin.** 1969. The Discovery Method in Training. HMSO, London.
2. **Bryce, C.F.A.** 1981. Computer-based learning in biochemistry, p. 13. In C.F.A. Bryce (Ed.), Biochemical Education. Croom Helm, London.
3. **Freire-Picos, M.A., M.I. Gonzalez-Siso, E. Rodriguez-Belmonte, A.M. Rodriguez-Torres, E. Ramil and M.E. Cerdan.** 1994. Codon usage in *Kluyveromyces lactis* and in yeast cytochrome *c*-encoding genes. *Gene* 139:43-49.
4. **Gouy, M. and C. Gautier.** 1982. Codon usage in bacteria—correlation with gene expressivity. *Nucleic Acids Res.* 10:7055-7074.
5. **Konopka, A.K.** 1994. Sequences and codes: fundamentals of biomolecular cryptology, p. 119-174. In D.W. Smith (Ed.), Biocomputing: Informatics and Genome Projects. Academic Press, San Diego.
6. **Maynard Smith, J. and N.H. Smith.** 1996. Site-specific codon bias in bacteria. *Genetics* 142:1037-1043.
7. **Sharp, P.M. and K.M. Devine.** 1989. Codon usage and gene expression level in *Dictyostelium discoideum*: highly expressed genes do 'prefer' optimal codons. *Nucleic Acids Res.* 17:5029-5039.
8. **Sharp, P.M., M. Stenico, J.F. Peden and A.T. Lloyd.** 1993. Codon usage: mutational bias, translational selection, or both? *Biochem. Soc. Trans.* 21:835-841.
9. **Shaw, G.** 1995. Protein sequence interpretation using a spreadsheet program. *BioTechniques* 19:978-983.
10. **Shields, D.C. and P.M. Sharp.** 1987. Synonymous codon usage in *Bacillus subtilis* reflects both translational selection and mutational biases. *Nucleic Acids Res.* 15:8023-8040.
11. **Wright, F.** 1990. The 'effective number of codons' used in a gene. *Gene* 87:23-29.

Address correspondence to Neil R. McEwan, Soil Biology Unit, Land Resources Department, Scottish Agricultural College, Craibstone Estate, Bucksburn, Aberdeen, AB21 9TQ, Scotland. Internet: n.mcewan@ab.sac.ac.uk. Derek Gatherer can be contacted at: School of Biomolecular Sciences, John Moores University, Byrom Street, Liverpool, L3 3AF, England, UK. Internet: d.gatherer@livjm.ac.uk

Received 16 June 1997; accepted 19 September 1997.

Neil R. McEwan and Derek Gatherer¹

*Scottish Agricultural College
Aberdeen, Scotland*

*¹John Moores University
Liverpool, England, UK*