WinGramm Version 1.0, Xalapa, May 2001
"Grammatical Complexity Analysis"

Program Description

by Rainer Feistel and Miguel Angel Jimenez-Montano

1. General

The program WinGramm ist freely distributable for non-commercial use only.
It is intended for scientific and engineering applications by scientists, engineers, or students.
WinGramm runs on Windows platforms, like Win95, Win98, WinNT 4.0, Win 2000.
It does not need any special hardware configuration except a graphics display, but sufficient memory and processor speed may prove very recommendable for use with longer sequences.

The authors cannot guarantee that the program is error-free. They do not take any warranty for possibly occuring problems caused by the use of the program or of its results. The use of the program is entirely at the risk of the user. On the other hand, hints on bugs or improvements of the program are always appreciated.

To install WinGramm on your computer, copy the components of the installation package,

- WinGra1.cab
- WinGra2.cab
- Setup.lst
- Setup.exe

into a suitable folder and start Setup.exe.

WinGramm works with time series files and with sequence files. Time series should be files with a single figure - the current value - per line, either integer or floating point, without any headlines, separators etc. Text behind the figure (or additional figures) is ignored. Sequence files consist of a number of lines with a number of characters per line. This might be one character per line, or all characters in just one line, or a mixture like in ordinary ASCI text files. Each character found is treated as symbol of the sequence, with the following exceptions. The characters linefeed LF = <10> and carriage return CR = <13> are expected to separate the lines, they are ignored when reading. Note that there are no spaces inserted automatically in case you study usual text files. You should completely refrain from using the other control characters <0> to <31>, and <255>, as symbols of your sequence. However, "space" = <32> is a permitted symbol. Briefly, the list of the 223 usable characters <32> to <254> is

!"#$%&'()*+,-
./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuv
wxyz{|}~□€□,ƒ„…†‡^‰Š‹Œ□Ž□□''""•--˜™š›œ□žŸ
¡¢£¤¥¦§¨©ª«¬®¯ °±²³´µ¶·¸¹º»¼½¾¿ÀÁÂÃÄÅÆÇÈÉÊËÌÍÎÏÐÑÒÓÔÕÖ×ØÙÚÛÜÝÞßàáâãäåæçè
éêëìíîïðñòóôõö÷øùúûüýþ

The way of appearance of several of these characters on your screen will depend on the regional country code settings on your computer.

WinGramm is used to generate and to analyze symbol sequences with the methods of subword entropies, grammatical complexities, and surrogate statistics. These are ways to study the information content, the complexity or the redundancy of any temporal data series in form of symbolic dynamics, of biomolecular sequences [1-4] of digital information carriers or of human writings[3]. For the details of these tools the user is referred to the literature listed at the end.

2. Menu "File"

2.1 File / Display Sequence File:
The file you select will be read and shown in a text window for inspection

2.2 File / Save Image as:
Offers you to save the contents of the main window as bitmap graphics (.bmp)

2.3 File / Print:
Prints the main window (including frame and caption) to paper on the default printer

2.4 File /Exit:
Ends WinGramm

3. Menu "Edit"

3.1 Edit / Copy :
Copies the content of the main window to the clipboard to be inserted into other Windows applications by their "paste" command.

3.2 Edit / Fractionate Sequence
A selected sequence can be split into one or more of its parts, and these fractions are linked to each other and saved as a new sequence file. The fractions are allowed to overlap, and they do not need to appear in the same order as in the sequence. Example: 101-300,1-200 will create a new sequence with 400 symbols, starting with the 101st of the original, and with the first of the original in position 201. This option is intended to easily cut pieces of interest out of long biomolecular chains.

3.3 Edit / Change Alphabet
This option is to exchange the letters of the sequence by another set of letters, like switching from (A,C,G,T) to (A,C,G,U) if you want to compare DNA with RNA, or if you want to have only the Purine/Pyrimidine (R,Y) characters represented in the original RNA (A,C,G,U) sequence. For the latter, as example, the old set ACGU has to be replaced by the new one, RYRY.

3.4 Edit / Generate a Sequence:
Allows you to create a certain reference time series or a corresponding symbol sequence.
The settings window which opens has 4 boxes, "Output", "Alphabet", "Partition", and "Source".

Within the "Output" box you can decide if you want the file be generated as a time series, i.e. a series of floating point values, one per line, or as a sequence of symbols, either one letter per line or all letters in a single long line. In the latter two cases, if selected, the "Partition" box becomes visible.

Within the "Alphabet" box you can specify the length of the series to be generated, either the number of floating point numbers, or the number of symbols to be written to the file. If you have chosen "Symbols" in the "Output" box, you can now set the number of symbols (or partitions) to be used to convert the series of numbers into a series of symbols. The value you enter should be at least 2 and not exceed 223. Corresponding to the number, WinGramm will choose a suitable set of characters:

no of letters	characters used
4:	"A","C","G","T"
2 to 10 :	"0" to "9"
11 to 16:	"0" to "9", "A" to "F"
20:	"G","A","D","P","S","C","N","E","K","V","Q","T","M","L","I","R","H","F","Y","W"
17 to 26:	"A" to "Z"
27 to 36:	"A" to "Z", "0" to "9"
37 to 52:	"A" to "Z", "a" to "z"
53 to 62:	"A" to "Z", "a" to "z", "0" to "9"
> 62:	<32> to <254>

Within the "Partition" box, which is only available if you selected "Symbols" in the "Output" box, you can choose between different ways to map the input stream of numbers into an output stream of symbols by defining partition intervals.
"Equal Frequency" will place the partition intervals such that all symbols of sequence will appear with equally often. In case of just two symbols, the threshold will be the median of the distribution. In general, this method will transfer the most information from the number series to the symbol series. WinGramm will show the probability distribution P(x), that a number less than or equal to x is found, and the partitions used in a separate window.
"Equal Intervals" will split the range between the smallest and the biggest number in equal fractions, so that rare values will be reflected by rare symbols.
"Multiples of R.M.S." will take the standard deviation of the distribution as width of the partition intervals, centered about the average value. This choice separates "average" from "exceptional" values by converting them into different symbols. Numbers outside the specified partitions will be added to the nearest valid partition.

Within the "Source" box you can choose between several standard number generation methods, more or less random or "chaotic".
"White Noise" produces equally distributed random numbers between 0 and 1.
"Exponential Noise" produces random numbers between 0 and infinity with mean = 1 and r.m.s. = 1.
"Gaussian Noise" produces random numbers between minus and plus infinity with mean = 0 and r.m.s. = 1.
„Bernoulli“ outputs binomial random numbers (n,p) between 0 and n. Both n and p can be set.
"Logistic Map" is the recurrence relation x(i+1) = R * x(i) * (1 - x(i)) with x(0) random between

0 and 1. The parameter R can be set to any value between 0 and 4, causing periodic or chaotic behaviour.
"Henon", either the x(i) or the y(i) chaotic series, is computed by

$$x(i+1) = 1 - a * x(i) * x(i) + b * y(i)$$
$$y(i+1) = x(i)$$

with the parameters
$a = 1.4, b = 0.3$

and with initial conditions
$-1 < x(0) < 1$ randomly
$-1 < y(0) < 1$ randomly

"Lorenz", either x(i) or y(i) or z(i) chaotic series, is computed by

$$x(i+1) = x(i) + dt * s * (y(i) - x(i))$$
$$y(i+1) = y(i) + dt * (x(i) * (r - z(i)) - y(i))$$
$$z(i+1) = z(i) + dt * (x(i) * y(i) - b * z(i))$$

with the parameters
$dt = 0.01, s = 10, r = 28, b = 8/3$

and with initial conditions
$-30 < x(0) < 30$ randomly
$-40 < y(0) < 40$ randomly
$0 < z(0) < 50$ randomly

"SineWave" is computed as a smooth periodic function

$$x(i) = \sin(2 * PI * i / T)$$

where T can be set to any value.

3.5 Edit / Discretize a Time Series
This option transforms a given time series into a chain of symbols. The options are exactly as described under point 3.4 except that the given data file replaces the internal number generator as data source.

3.6 Edit / Make Surrogate Files
Given a sequence file of your choice, this is to construct a set of surrogate sequence files [8] corresponding to it. A window pops up that will ask you the number of surrogates you want to produce, and if these random files should either contain the same frequency of the symbols as the original one, or the same frequency of symbol pairs, or of symbol triplets. Note that this option is intended to have surrogates for an external purpose; if you just want the statistical properties of a surrogate ensemble and not the single surrogates themselves, you may want to turn to the menu 5.8, "Calc/Surrogate Statistics"

4. Menu "View"

4.1 View / Close All Text Windows
After handling several sequences, the corresponding text windows with sequences and analysis results may have accumulated on your screen. To get rid of them without closing separately one by one, click here.

4.2 View / Close All Plot Windows
After handling several sequences with their distribution functions, the corresponding graph windows may have accumulated on your screen. To get rid of them without closing separately one by one, click here.

5. Menu "Calc"

5.1 Calc / Repeating Subwords
A sequence you select is analyzed for repeating subwords of any length. All these words, sorted by their length and frequency of appearance, together with the positions they take in the sequence, and their average repeating distances with standard deviations, are listed in a result file which is shown in a text window.

5.2 Calc / Exceptional Subwords
A sequence you select is analyzed for repeating subwords of any length. For all repeating words longer than 2, the probability p for their occurence is estimated by the frequency by which its letters and its pairs occur in the sequence, and compared to the actual frequency f of the word. If this frequency it appears with is at least 3 times higher than it is probable if considered as a first order Markov chain, f >= 3*p, then the word is notified as exceptional [11], and is listed in the result file together with its values for f and p and the positions where it occurs in the sequence.

5.3 Calc / Subword Entropies
A sequence you select is analyzed for repeating subwords of any length. The corresponding subword entropies [5] up to the length of the longest repeating word are computed and printed to a result file which is shown in a separate text window.
Entropies H(l) are defined as the sum of P(i) * log(P(i)) over all words i of length l in the sequence, log is the logarithm with alphabet size as base, and P(i) = f(i) / N(l) are relative frequencies. f(i) is the absolute frequency (i.e. the count) the word i appears with, and N(l) is the number of subwords of length l that fit overlapping into the given sequence. For a sequence of lenght L, N(l) = L - l + 1.
Relative entropies h(l) = H(l+1) - H(l) describe the average chance that a certain letter follows a given word of length l. They are always equal to 1 for random, infinite sequences.

5.4 Calc / Conditional Entropies
A sequence you select is analyzed for repeating subwords of any length. For those words W which repeat as least as often as there are letters in the alphabet, the Conditional Entropies [13] are computed as h(W) = - sum P(X) * log(P(X)), where the sum extends over all letters X of the alphabet, and P(X) is the relative frequency of the letter X in positions just behind W.

5.5 Calc / Grammatical Complexity
A window opens with two selection boxes, "Analyze" and "Method". The first one offers the

choice between the treatment of a single sequence file, say "sequence.txt", or a set of files, in this case all files with extension ".txt" within the same folder as "sequence.txt". In the first case all grammatical rules will be presented in the result file, in the second case you only get all the particular complexities together with a simple statistics. In the "Method" box, four different ways are offered how the grammatical rules are to be constructed. Although none of those can guarantee to find the genuine minimum complexity value, all should result on average in comparable values, however. "NVOGRAM" [6] is a fast and effective mixed strategy: First all pairs are searched which repeat more than twice, and replaced by new symbols ("non-terminals"), and their replacement rules are noted. After they have exhausted, all repeating words are taken, beginning with the longest, until no repeating word longer than 2 is found. "Shortest First" always substitutes the shortest repeating words first, "Longest First" does the same beginning with the longest repeating words. "Maximum Gain", finally, compares the gains of all possible substitutions of any length and takes next the one which pays best. This is a "steepest descent" path to the minimum.

5.6 Calc / Algorithmic Distance

You are prompted to enter two sequence files, the distance between them is to be computed. Note that the method is in no way restricted to sequences of equal length. It can even be used if the sequences are built of different alphabets, but this is not at all the intended way of application. For both sequences, x , y, their grammatical complexities K(x) and K(y) are computed by the NVOGRAM method, compare 5.5. Next, both are concatinated to form a single sequence, and again the complexity K(xy) is determined. Then, the algorithmic distance [7] between both is given by D(x, y) = K(xy) + K(yx) - K(x) - K(y).
A similar measure [10] is the corresponding relative distance d(x, y) = 2*D(x, y) / (K(xy) + K(yx).
The Conditional Complexities („Algorithmic Entropies“) K(x|y) = K(yx) - K(y) and K(y|x) are given as well.

Although the Symmetry of Information Theorem states K(xy) = K(yx), this is not necessarily true for finite sequences and the approximate grammatical complexity. Therefore, we use here the symmetrical expression (K(xy) + K(yx))/2 instead of simply K(xy).

5.7 Calc / Distance Matrix Tree

After you have selected a sequence file, all files with same extension in the same folder will be used to construct a „phylogenetic tree“. First, the relative algorithmic distance is computed for each pair of sequences, see 5.6, and printed to the result file. Note that for speeding up the computation, symmetry K(xy) = K(yx) is assumed and only one of both is actually computed.

Then, as a simple and fast example, a binary cluster tree is constructed using minimum distances. First, the pair with the smallest distance is taken to form a binary cluster. For this cluster, the distance to all remaining sequences is determined as the particular minimum of both the distances of the cluster sequences. Then, the cluster replaces the two sequences in the now reduced distance matrix, and the process is repeated until a single cluster, the root, remains.
Behind the distance table, the edge list and the tree are shown in the report file, „Similar.log“.

Note that there are many sophisticated tree construction algorithms available elsewhere [10, 12] which start from a given distance matrix.

5.8 Calc / Surrogate Statistics
For a given sequence of your choice, a related surrogate ensemble [8] and its statistical properties can be computed . First, you have to specify the number of surrogates, i.e. the size of the ensemble to be investigated. Next, you can preserve either symbols, pairs, or triples when creating surrogates. Preserving triples will cause surrogates with properties much closer to the original one than preserving just symbols will do. The statistics done by the program uses grammatical complexities computed with the NVOGRAM method, see 5.5.
If "Add Entropies" is checked, for all surrogates subword entropies are extracted additionally up to length 3 if preserving only symbols, and to length 5 if preserving even triples.
If „Add Exceptional Words“ is checked, for all repeating words of the original sequence their average frequency in the surrogate ensemble is counted. If the original frequency is three times higher than average, the words, their frequencies and positions are provided in the final report.
For all complexities and, if wanted, entropies, ensemble averages and standard deviations are calculated and printed into a result file which is shown in a separate window. Additionally, the so-called S measure (deviation from the mean in units of standard deviations) and the surrogate redundancy [9] are evaluated from complexities. The latter one has gained attention because it is an almost length invariant measure of sequence complexity.

6. Menu "Help"

6.1 Help / About
Shows some information about the authors of WinGramm.

6.2 Help / References
Shows some background references for the algorithms applied in the program.

6.3 Help / Using WinGramm:
Shows this program description, if the WinGramm.rtf file is placed in a folder with the program.

7. Acknowledgement

The algorithmic conception of WinGramm is by:
Miguel Angel Jimenez-Montano, Xalapa, Mexico
Email address: mjimenez@mia.uv.mx

The concepts of Conditional Entropies and Exceptional Subwords were suggested by
Werner Ebeling, Berlin, Germany
Email address: werner@summa.physik.hu-berlin.de

Program implementation of WinGramm was done by:
Rainer Feistel, Warnemuende, Germany
Email address: rainer.feistel@io-warnemuende.de

The program is available for free download from the homepage of one of us,
http://www.io-warnemuende.de:20080/homepages/rfeistel/index.html

The development of WinGramm was supported by:
CONACYT Project #32201-E

"Structure and Evolution of the Genetic Code"

8. Literature

[1] L.L. Gatlin : Information Theory and the Living System. Columbia Univ. Press, New York 1972

[2] W. Ebeling, R. Feistel, M.A. Jimenez-Montano: On the theory of stochastic replication and evolution of molecular sequences. Rostocker Phys. Manuskripte **2**(1977) 105

[3] W.Ebeling, M.A. Jimenez-Montano: On grammars, complexity and information measures of biological macromolecules. Math. Biosc. **52**(1980)53

[4] W.Ebeling, R. Feistel: Physics of Self-Organization and Evolution (in German). Akademie-Verlag Berlin 1982.

[5] A.M.Yaglom, I.M.Yaglom: Probability and Information (in German). Verlag der Wissenschaften, Berlin 1965

[6] M.A.Jimenez-Montano, T.Poeschel, P.E.Rapp, in: Biological Complexity, Symposium, Montevideo, Uruguay 1997 p.133-142

[7] W.H.Zurek: Thermodynamic cost of computation, algorithmic complexity and the information metric. Nature, Vol. **341**, 14 Sep 1989, p.119-124

[8] S.F.Altschul, B.W.Erickson: Significance of Nucleotide Sequence Alignments: A Method for Random Sequence Permutation that Preserves Dinukleotide and Codon Usage. Mol. Biol. Evol. **2**(6):526-538. 1985

[9] P.E.Rapp, C.J.Celluci, K.E.Korslund, T.A.A.Watanabe, M.A.Jimenez-Montano: An Effective Normalization of Complexity Measurements for Epoch Length and Sampling Frequency. Phys.Rev.E, (in press)

[10] M.Li, J.H.Badger, X.Chen, S. Kwong, P.Kearney, H.Zhang: An information-based sequence distance and its application to whole mitochondrial genome phylogeny.
Bioinformatics **17**,2 (2001) 149-154

[11] C.Nicolis, W. Ebeling, C. Baraldi: Markov processes, dynamic entropies and the statistical prediction of mesoscale weather regimes,
Tellus 49A (1997) 108-118

[12] Kuhner, M.K. and J. Felsenstein. 1994. A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. Molecular Biology and Evolution 11:459-468.

PHYLIP program: Joe Felsenstein, Department of Genetics, University of Washington
Email address: joe@genetics.washington.edu
URL: http://evolution.genetics.washington.edu/phylip/software.html

[13] W. Ebeling, Prediction and entropy of nonlinear dynamical systems and symbolic sequences. Physica D **109** (1997) 42-52