Exploring Structure and Evolution of the Genetic Code with the Software Tool GCAT

E. Fimmel[1*1], M. Gumbel[1], and L. Strüngmann[1]

[1]Institute for Mathematical Biology, Mannheim University of Applied Sciences
{e.fimmel,m.gumbel,l.struengmann}@hs-mannheim.de
http://www.mbi.hs-mannheim.de

Abstract. The genetic code can be seen as the major key to biological self-organisation. In fact, all living organisms regardless of whether they are plants, bacteria or mammals have the same molecular bases: Adenine, cytosine, guanine, and thymine. Unidimensional sequences of these bases contain the genetic information for the synthesis of proteins in all forms of life. Thus, one of the most fascinating questions is to explain why evolution has produced the current genetic code and why it exists in its present form.

Motivated by these fundamental questions, a new software tool – *Genetic Code Analysis Toolkit (GCAT)* – was developed which can be used to investigate properties of the genetic code in order to develop hypotheses about its origin and evolution. The main focus of the tool has been put on the graphical visualisation of the data.

In the present paper we will describe in short the tool GCAT and give a couple of applications presenting new results on circular codes and the structure of some ancient codes.

Keywords: Circular Codes, Genetic Information, Binary Dichotomic Algorithms, GCAT, Genetic Code Analysis Toolkit.

1 Introduction

The genetic code is a dictionary that translates triples of molecular bases into amino acids which are then combined to form proteins – the main ingredient for all kind of livings. Protein synthesis is therefore the essential mechanism that nature uses in forming life. A long evolutionary process has produced the current genetic code table passing through several ancient codes and the central and fascinating question is why the code is at it is. Is it optimised in any sense or is its present form just a *frozen accident*? After Watson and Crick [3] had discovered the DNA helix structure and then deciphered the encoding of 20 amino acids (plus a stop signal) by 64 codons, the scientific community was tempted to believe that

[1] Corresponding author.

Z. Hu et al. (eds.), *Advances in Artificial Systems for Medicine and Education*,
Advances in Intelligent Systems and Computing 658, DOI 10.1007/978-3-319-67349-3_2

the most central riddle of nature was solved and it was hoped that (genetic) deceases could be cured in the near future. However, even after the 1000-Genome-Project [20] we are still far from this paradisiac situation. Nowadays, it is clear that there are more secrets in protein synthesis than we know and it is believed that only interdisciplinary efforts by biologists, physicians, chemists, bioinformaticians and mathematician will help to fully understand this complex process.

About 20 years ago, a class of so-called *circular* genetic codes[2] were found statistically in genetic coding sequences [2]. These codes have remarkable prop-erties that seem to play an essential role in the process of translation. They could be the reason for one of the fundamental features of the genetic code, the *noise immunity*: The ribosome apparently has mechanisms to detect and correct errors that arise during the translational process, for instance, those caused by unintentional insertion or deletion of a base in a sequence. The translation process in the ribosome is naturally error-prone and it is highly unlikely that a complex system like this develops without any error correcting or detecting mechanism. A series of publications [4, 5, 7, 8, 9] shows fascinating similarities between properties of circular codes and processes within protein synthesis. Moreover, a recent theory states that nature evolved from simpler ancient codes that had high error correcting features to more complex codes by paying the price of error correction and instead using a nested second code - a circular code. Thus, also from an evolutionary point of view circular code theory is of great importance and at the same time extremely difficult from a mathematical point of view.

Inherently one often needs to examine many particular cases in the above research in order to understand a pattern behind them for suggesting a theoretical explanation of a process. This is very time-consuming, if done manually, and in particular in the case of the genetic code one has to face a complexity that is comparable to those of a chess game on a 8×8 chess board. Moreover, manual handling of such examples is error-prone and a manual investigation of *real* data, for instance sets of data from GenBank or from FASTA files, is completely impossible due to their sizes (compare also [1, 12, 15]). Thus, the only possible approach to achieve useful results on this field is to have a computer aid. However, there was no re-usable and expandable tool accessible for this kind of genetic data that had a user-friendly editor and a main focus on its graphical output. We hope to fill this slot by means of our software *Genetic Code Analysis Toolkit* (GCAT). GCAT is capable to handle sets of codons as well as genetic sequences from GenBank or FASTA files. A special feature of the tool is a comprehensive editor which, amongst other things, visualises graphically the processes implemented. This is one of the characteristics which distinguishes GCAT from other tools like Bioconductor [14] that require knowledge in scripting.

2 Software

GCAT is a Java-based application with a comprehensive user interface for a rich user

[2] More details can be found in Section 'Circular Codes' below.

experience. Although developed as a stand alone software, its architecture enables the integration of bioinformatic toolkits like Bioconductor or R [18] in general. Technically, GCAT consists of two major components:

1. GCAT comes with a domain model for nucleic bases and amino acids that is linked and integrated with BioJava [17]. BioJava is a feature rich frame-work for bioinformatics, and GCAT primarely uses it to import and export sequences from a file or web services.
2. Operations are the key functions in GCAT which are applied to a sequence or a set of codons. They either modify or analyze a collection of tuples and the result of an operation may be the input for another. Operations can be used in an interactive mode similar to a shell or a workbench like R Studio or Matlab or in batch mode. Each operation may also have a graphical user interface. All operations are designed as plug-ins that are loosely coupled with GCAT's core components. New plug-ins can easily be added even without re-compiling the software.

GCAT is freely available at http://www.gcat.bio.

3 Applications

In this section we discuss three applications of our tool GCAT that led to new results on the genetic code and its evolution. The first two show interesting properties of circular codes and the results in Theorem 1 and Theorem 2 lack theoretical proofs so far but have been purely obtained by investigating the data using GCAT. The third result sheds some light on a theory by Jiménez-Montano about the origin of the genetic code.

3.1. Circular Codes

A fundamental feature of the genetic code is its noise immunity: It is resistant to mutational effects caused by e.g. insertion or deletion of a nucleic base or simply an incorrect grouping of codons as a consequence of a false reading frame. Such mutations can be fatal during the translation process, may result in nonfunctional proteins and can thus lead to genetic diseases like, first of all, some kinds of cancer, Tay-Sachs disease, or HIV.

In 1996, Arqués and Michel [2] selected, by means of statistical analysis, the 20 most frequent codons in the 3 different reading frames of genetic coding sequences. The analysis was done in large gene populations of prokaryotes and eukaryotes (later on extended to bacteria and viruses). Surprisingly, the set of codons they found formed a *circular code*. Such codes are a weaker version of *comma-free codes* which had been proposed by Crick in 1957 [3] as a solution for the frame-shift problem. Every out-of-frame codon can be recognised immediately by comma-free codes as it cannot belong to the code by definition. However, an experimental evidence that comma-free codes are used in nature has been missing. Circular codes have likewise the property of

detecting frame-shift mutations during the translation as every sequence of bases has at most one decomposition into codons from a circular code if read on a circle. However, the frame-shift error can only be detected eventually and after at most 13 bases. The property of circularity as well as that of comma-freeness are preserved under a systematic exchange of nucleic bases and a reversing permutation of the codon bases. This led to a systematic study and classification of circular codes in [4] using group theory.

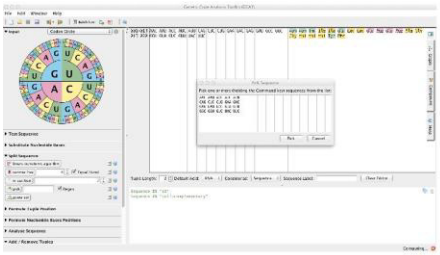
Even though circular codes have been found in nature as subcodes of the genetic code and although they compile very well with the codon bias of coding sequences, it is not clear how nature uses their error detecting and error correcting features. This asks for a comprehensive and systematic statistical investigation of data sets from data bases like GenBank in order to better understand the properties of circular codes and their appearance in coding sequences as well as the possible role they play in forming the three dimensional structure of proteins. In GCAT many features have been implemented which are designed to carry out this analysis for either sets or sequences of codons. Below we list some exam-ples of such functionalities of GCAT which have turned out to be useful when working with circular codes:

- Test Sequence: The tool tests the given set of codons for properties like comma-freeness, circularity, or self-complementarity.
- Substitute Nucleotide Bases: Applies one of the 24 possible permutations of nucleotide bases to all codons in a given code.
- Split Sequence: In this functionality of the tool different possibilities to split a given code according to some rules are collected. It includes the important BDA subunit (compare the corresponding subsection below) and splittings into comma-free subcodes.
- Permute Nucleotide Bases Positions: In each codon in a given code the order of bases is changed according to the same rule.
- Analyse Sequence: This functionality provides different statistical features for a given code like number of amino acids coded or codon usage in a sequence from a given file.

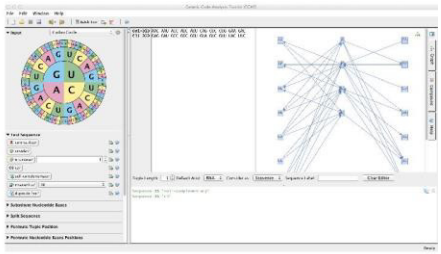
We close this subsection with a new theorem that was obtained by using the *Split Sequence* feature of our tool. However, it lacks a theoretical proof so far.

Theorem 1. *Any maximal circular code can be partitioned into* 4 *comma-free codes of size* 5.

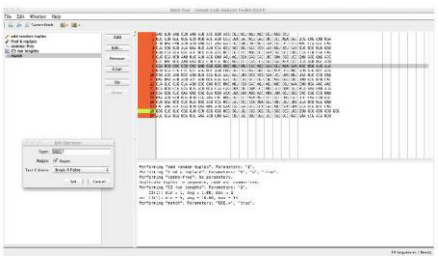
In Figure 1 (a) an example is given where a maximal circular code is divided into four comma-free subcodes each of size five.



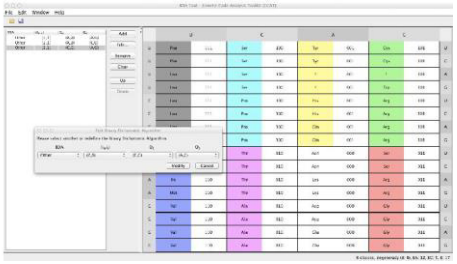
(a) Splitting into comma-free subcodes



(b) A maximal self-complementary circularcode



(c) Batch tool



(d) Ancient genetic code with the BDA editor

Fig. 1. Screenshots of GCAT.

3.2. Graphical Visualisation

In [5] a new graph theoretical approach to investigate properties of subcodes of the genetic code was suggested. Given a subcode X of the genetic code, i.e. a set of codons, one associates a directed graph to X by assigning to each codon $N_1N_2N_3$ two edges: one connecting the first base with the second and third $N_1 \rightarrow N2N3$ and a second edge connecting the first two bases with the last one $N_1N_2 \rightarrow N_3$. Figure 1 (b) shows an example of this graphical presentation. The directed graph associated to a code visualises the code's properties like circularity and comma-freeness very well. For instance, a code is circular if and only if its associated graph is acyclic, i.e. does not contain any circle, while comma-freeness is equivalent to a maximal path length not exceeding 2 (see [5] for more details). Using the same approach for dinucleotides instead of trinucleotides (codons) showed a beautiful similarity between the theory of dinucleotide circular codes and tournaments on four vertices (see [5]) while restricting to graphs with maximal path length equal to 1 even led to the discovery of a new class of strong comma-free codes (see [6]).

The tool GCAT can be used to display and investigate the graphs associated to subcodes of the genetic code. Various layouts (organic, stack, circle) of the graph can be displayed in the editor. Thus properties like maximal path length or acyclicity of the graph can be easily seen in GCAT. A new theorem that has been obtained using our tool is the following. It will be discussed in [7]:

Theorem 2. *A maximal circular code is self-complementary if and only if it satisfies*

the following two conditions:

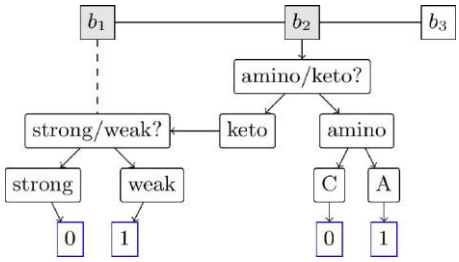
(i) the set of vertices is self-complementary;
(ii) for any vertex v the number of outgoing edges of v equals the number of ingoing edges of its complemented vertex.

The necessity of the two conditions in Theorem 2 is easily established but the sufficiency again lacks a theoretical proof so far. In Figure 1 (b) an example is given where the graph of a maximal self-complementary circular code is displayed.

3.3. Binary Dichotomic Algorithms

When it comes to mutations, the essential question is how a codon can be read correctly by the ribosome. The investigations in [13, 10] showed that it is possible to determine a codon uniquely by applying a sequence of so called Binary Dichotomic Algorithms (BDA). Those algorithms ask chemical questions to the bases involved in the codon that might be performed by the ribosome.

GCAT contains such a component (1d) that analyses and links the structure of the genetic code to BDAs. A BDA partitions codons into two classes (0 or 1) and it is a generalisation of known partitions like the Rumer- [19], Parity-, Complementary- or Hidden-classification [11]. Fig. 2 shows an example for the Rumer-BDA.



(a) Visual representation

BDA	(*i1*, *i2*)	*Q1*	*Q2*
Rumer	(2, 1)	(*C*, *A*)	{*C*, *G*}

(b) Table representation

Fig. 2. The Rumer classification is an important example for a binary dichotomic algorithm. Here a codon is classified into the classes 0 or 1 depending on the chemical properties of the bases at positions 1 and 2. (a) shows a visual representation of the algorithm. The first decision is related to base 2. If it is of type keto (i.e. U or G) the first base is also considered: If the first base is of type strong (C or G) a 0 is assigned and otherwise a 1. For instance, the codon CUG obtains 0. (b) lists the same information in a table format. (i1, i2) indicates the involved bases for the first and second decision (or question), Q1 and Q2 are the bases used for the decision (Q like question).

When such a classification of a codon is repeated with different BDAs we can concatenate the classes to a binary word consisting of zeros and ones. This leads to a larger number of different words (or classes) and enables us to create more than two classes for a single codon. This method was used to address different questions: We have shown that it is possible with a set of six BDAs to create 64 classes [13], i.e. every codon can be distinguished uniquely. In particular, the ribosome is a possible biological entity that implements this mechanism. Weak base-pairs (i.e. A-U or U-A) are indistinguishable one from another in the minor groove of an RNA strand. Strong base-pairs (C-G or G-C) also cannot be distinguished in the minor groove, but display a different profile of hydrogen bonds compared to weak base pairs. These observations narrow down possible BDAs. However, it is still possible to create 64 classes with reading-head-compatible BDAs though [13]. This suggests a hypothesis for a possible error-detection mechanism in the translation process.

Another application is to match ancient codes by means of BDA classifications. The evolution theory according to Jiménez-Montano [16] claims that at the beginning there were only eight amino acids. Such a genetic code table could be perfectly fitted (see table 1). Here again, like in the Rumer-BDA above, the first two bases are required to classify the ancient amino acids. Fig. 1 (d) shows the BDA editor that produces this ancient code. We, again, close this section with the following theorem:

Theorem 3. *The ancient code by Jiménez-Montano can be produced by BDAs.*

Table 1. Ancient genetic code according to [16]. (a) This precursor code has eight amino acids represented by eight classes which can be fit with three BDAs shown in right table (b).

(a) Ancient code table

	U	C	A	G	
U	Leu 111	Pro 100	! 001	Arg 101	U
U	Leu 111	Pro 100	! 001	Arg 101	C
U	Leu 111	Pro 100	! 001	Arg 101	A
U	Leu 111	Pro 100	! 001	Arg 101	G
C	Leu 111	Pro 100	! 001	Arg 101	U
C	Leu 111	Pro 100	! 001	Arg 101	C
C	Leu 111	Pro 100	! 001	Arg 101	A
C	Leu 111	Pro 100	! 001	Arg 101	G
A	Val 110	Ala 010	Asp 000	Gly 011	U
A	Val 110	Ala 010	Asp 000	Gly 011	C
A	Val 110	Ala 010	Asp 000	Gly 011	A
A	Val 110	Ala 010	Asp 000	Gly 011	G
G	Val 110	Ala 010	Asp 000	Gly 011	U
G	Val 110	Ala 010	Asp 000	Gly 011	C
G	Val 110	Ala 010	Asp 000	Gly 011	A
G	Val 110	Ala 010	Asp 000	Gly 011	G

(b) BDAs

BDA	(*i1*, *i2*)	*Q1*	*Q2*
A1	(2, 1)	(*A*, *U*)	{*A*, *G*}
A2	(2, 1)	(*A*, *U*)	{*U*, *C*}
A3	(2, 1)	(*C*, *G*)	{*A*, *G*}

4 Discussion and Conclusion

The Genetic Code Analysis Toolkit (GCAT) is a novel user-friendly and extensible software tool which is intended to be used by (theoretical) biologists, mathematicians and physicists to explore structural properties of the genetic code. The software is open source, so it can be developed further in order to establish hypotheses concerning structural properties of the genetic code and execute tests to prove these. In the present paper we have described in short some already implemented functionalities of the tool and have given some exam-ples of its applications. A special feature of the tool is its graphical visualisation of the genetic code and a possibility to consider not only genetic codes consisting of trinucleotides but also n-nucleotides for every given natural n. It allows, for instance, to examine dinucleotide and tetranucleotide genetic codes that can be useful in order to study the evolution of the modern genetic code.

Acknowledgments We would like to thank Karin Adler for her contribution to ancient genetic codes. The authors are deeply grateful to Kristian Kraljic for his excellent work in the implementation of the GCAT and to the Karl Völker Foundation for their financial support during the work on the GCAT.

References

1. Mohammed Abo-Zahhad,Sabah M. Ahmed,Shimaa A. Abd-Elrahman,"Genomic Analysis and Classification of Exon and Intron Sequences Using DNA Numerical Mapping Techniques", International Journal of Information Technology and Computer Science(IJITCS), vol.4, no.8, pp.22-36, 2012. DOI: 10.5815/ijitcs.2012.08.03
2. Arqu`es D. G. and Michel C. J.: *A complementary circular code in the protein coding genes*, J.Theor. Biol., 182, 45-58, 1996.
3. Crick F., Griffith J. S., and Orgel L. E.: *Codes without commas, Proceedings* of the National Academy of Sciences of the United States of America, 43, 1957, 416-421.
4. Fimmel E., Giannerini S., Gonzalez D., Strüngmann L.: *Circular codes, symmetries and transformations* J. of Mathematical Biology, (July 2014), 70(7):1623-44.
5. Fimmel E., Michel C.J., and Strüngmann L.: *n-nucleotide circular codes in graph theory*, Phil Trans.A, volume 374: 20150058, 2016.
6. Fimmel E., Michel C.J., and Strüngmann L.: *Strong comma-free codes*, submitted for publication.
7. Fimmel E., Michel C.J., and Strüngmann L.: *Self-complementarity in genetic codes*, in preparation.
8. Fimmel, E.; Strüngmann, L.: *On the hierarchy of trinucleotide n-circular codes and their corresponding amino acids*, J. Theor Biol., 2015, 364, 113-120.
9. Fimmel E., Strüngmann L.: *Codon Distribution in Error-Detecting Cir-cular Codes*, Life 2016, 6(1), 14; doi:10.3390/life6010014.
10. Fimmel, E., Danielli, A., Strüngmann, L.: *On dichotomic classes and bi-jections of the genetic code*, J. Theor. Biol. 336 (0), 2013, 221?230.

11. Gonzalez D.L., Giannerini S., Rosa R.: *Strong short-range correlations and dichotomic codon classes in coding DNA sequences* Phys. Rev. E: Stat. Nonlin. Soft Matter Phys. Nov 2008. 78 (5 Pt 1), 051918.
12. Syed Mahamud Hossein, S.Roy,"A Compression & Encryption Algorithm on DNA Sequences Using Dynamic Look up Table and Modified Huffman Techniques", International Journal of Information Technology and Computer Science(IJITCS), vol.5, no.10, pp.39-61, 2013. DOI: 10.5815/ijitcs.2013.10.05
13. Gumbel M., Fimmel E., Danielli A., Strüngmann L.: *On Models of the Genetic Code generated by Binary Dichotomic Algorithms*. Biosystems 128 (2015) 9-18. http://dx.doi.org/10.1016/j.biosystems.2014.12.001.
14. Huber W., Carey V. J., Gentleman R., Anders S., Carlson M., Carvalho B. S., Bravo H. C. , Davis S., Gatto L., Girke T., Gottardo R., Hahne F., Hansen K. D., Irizarry R. A., Lawrence M., Love M. I., MacDonald J., ObenchainV. , Ole? A. K., Pag`es H., Reyes A., Shannon P., SmythG. K., Tenenbaum D., Waldron L., Morgan M.: *Orchestrating high-throughput genomic analysis with bioconductor*, Nat Methods 12 (2) (2015) 115? 121. doi:10.1038/nmeth.3252. URL http://dx.doi.org/10.1038/nmeth.325210 E. Fimmel et al.
15. J.K. Meher, M.R. Panigrahi, G.N. Dash, P.K. Meher,"Wavelet Based Lossless DNA Sequence Compression for Faster Detection of Eukaryotic Protein Coding Regions", International Journal of Image, Graphics and Signal Processing(IJIGSP), vol.4, no.7, pp.47-53, 2012. 10.5815/ijigsp.2012.07.05
16. Jim´enez-Monta˜no, M. A.: *Protein evolution drives the evolution of the genetic code and vice versa. Biosystems*, (1999), 4(1-2):47?64.
17. Prlic A., Yates A., Bliven S. E., Rose P. W., Jacobsen J., Troshin P. V., Chapman M., Gao J., Koh C. H., Foisy S., Holland R., Rimsa G., Heuer M. L., Brandstätter-Müller H., Bourne P. E., Willis S.: *Biojava: an open-source framework for bioinformatics in 2012*, Bioinformatics 28 (20) (2012) 2693?2695. doi:10.1093/bioinformatics/bts494. URL http: //dx.doi.org/10.1093/bioinformatics/bts494
18. R Development Core Team, R: *A Language and Environment for Statisti-cal Computing, R Foundation for Statistical Computing,* Vienna, Austria, ISBN 3-900051-07-0 (2008). URL http://www.R-project.org
19. Rumer Yu.B.: *Translation of ?Systematization of Codons in the Genetic Code [I-III]? by Yu. B. Rumer (1966-1968)*. Phil. Trans. R. Soc. A, 2016, 374: 20150446
20. The 1000 Genomes Project Consortium: *A global reference for human genetic variation*. Nature, 2015, 526:7571, 68?-74, http://dx.doi.org/ 10.1038/nature15393

http://www.springer.com/978-3-319-67348-6

Advances in Artificial Systems for Medicine and Education

Hu, Z.; Petoukhov, S.; He, M. (Eds.)
2018, XIII, 344 p. 128 illus., Softcover
ISBN: 978-3-319-67348-6