

Identification and Analysis of Patterns in DNA sequences, the Genetic Code and Transcriptional Gene Regulation

Dissertation

zur Erlangung des akademischen Grades
doctor rerum naturalium (Dr. rer. nat.)



vorgelegt dem Rat der Biologisch-Pharmazeutischen Fakultät
der Friedrich-Schiller-Universität Jena

von Magister der Mathematik Swetlana Nikolajewa
geboren am 19. November 1975 **in** Rostow-am-Don, Russland

Die vorliegende Arbeit wurde in der Arbeitsgruppe Theoretische Systembiologie (TSB) des Leibniz-Instituts für Altersforschung Fritz-Lipmann-Institut e. V. (vormals IMB) unter der Leitung von Dr. Thomas Wilhelm angefertigt.



Gutachter:

1. Prof. Dr. Stefan Schuster

2.

3.

Tag des Rigorosums:

Tag der öffentlichen Verteidigung:

Abstract

The present cumulative work consists of six articles linked by the topic "Identification and Analysis of Patterns in DNA sequences, the Genetic Code and Transcriptional Gene Regulation". We have applied a binary coding, to efficiently find patterns within nucleotide sequences. In the first and second part of my work one single bit to encode all four nucleotides is used. The three possibilities of a one - bit coding are: keto (**G,U**) - amino (**A,C**) bases, strong (**G,C**) - weak (**A,U**) bases, and purines (**G,A**) - pyrimidines (**C,U**). We found out that the best pattern could be observed using the purine - pyrimidine coding. Applying this coding we have succeeded in finding a new representation of the genetic code which has been published under the title "A New Classification Scheme of the Genetic Code" in "Journal of Molecular Biology" and "A Purine-Pyrimidine Classification Scheme of the Genetic Code" in "BIOForum Europe". This new representation enables to reduce the common table of the genetic code from 64 to 32 fields maintaining the same information content. It turned out that all known and even new patterns of the genetic code can easily be recognized in this new scheme. Furthermore, our new representation allows us for speculations about the origin and evolution of the translation machinery and the genetic code. Thus, we found a possible explanation for the contemporary codon - amino acid assignment and wide support for an early doublet code. Those explanations have been published in "Journal of Bioinformatics and Computational Biology" under the title "The New Classification Scheme of the Genetic Code, its Early Evolution, and tRNA Usage". Assuming to find these purine - pyrimidine patterns at the

DNA level itself, we examined DNA binding sites for the occurrence of binary patterns. A comprehensive statistic about the largest class of restriction enzymes (type II) has shown a very distinctive purine - pyrimidine pattern. Moreover, we have observed a higher **G+C** content for the protein binding sequences. For both observations we have provided and discussed several explanations published under the title "Common Patterns in Type II Restriction Enzyme Binding Sites" in "Nucleic Acid Research". The identified patterns may help to understand how a protein finds its binding site.

In the last part of my work two submitted articles about the analysis of Boolean functions are presented. Boolean functions are used for the description and analysis of complex dynamic processes and make it easier to find binary patterns within biochemical interaction networks. It is well known that not all functions are necessary to describe biologically relevant gene interaction networks. In the article entitled "Boolean Networks with Biologically Relevant Rules Show Ordered Behavior", submitted to "BioSystems", we have shown, that the class of required Boolean functions can strongly be restricted. Furthermore, we calculated the exact number of hierarchically canalizing functions which are known to be biologically relevant. In our work "The Decomposition Tree for Analysis of Boolean Functions" submitted to "Journal of Complexity", we introduced an efficient data structure for the classification and analysis of Boolean functions. This permits the recognition of biologically relevant Boolean functions in polynomial time.

Zusammenfassung

Die vorliegende kumulative Arbeit besteht aus 6 Artikel, die durch das Thema "Identifikation und Analyse von Mustern in DNA Sequenzen, im genetischen Code und in der transkriptionellen Genregulation" verbunden sind.

Im ersten und zweiten Teil meiner Arbeit wurde eine binäre Kodierung angewandt, um effizient Muster in Nukleotidsequenzen finden zu können. Die Kodierung aller 4 Basen durch ein einzelnes Bit stellte sich als die Einfachste heraus. Es gibt die folgenden Möglichkeiten einer Einzelbitkodierung: Keto (**G,U**) - Amino (**A,C**) Basen, Starke (**G,C**) - Schwache (**A,U**) Basen und Purine (**G,A**) - Pyrimidine (**C,U**). Beim Test aller drei Möglichkeiten erkannten wir, dass bei einer Purin/Pyrimidin Kodierung der Basen die signifikantesten Muster entstanden. Mittels der binären Kodierung (Purine (1) - Pyrimidine (0)) gelang es uns eine neue Darstellung des genetischen Codes zu finden und unter dem Titel "A New Classification Scheme of the Genetic Code" in "Journal of Molecular Evolution", sowie unter dem Titel "A Purine-Pyrimidine Classification Scheme of the Genetic Code" in "BIOForum Europe" zu veröffentlichen. Diese neue Darstellung erlaubte uns eine Reduzierung des bekannten Schemas von bisher 64 auf 32 Felder, bei gleichbleibendem Informationsgehalt. Es stellte sich heraus, dass alle bereits bekannten Muster des genetischen Codes leicht zu erkennen sind und sogar noch weitere hinzukommen. Weiterhin lässt das neue Schema Spekulationen über die Entstehung und Entwicklung der heutigen Translationsmaschinerie und des genetischen Codes zu. Wir fanden eine mögliche Erklärung der heutigen tRNA - Aminosäure Zuordnung und breite Unterstützung für einen früheren *Doublet*

Code. Dies konnten wir in "Journal of Bioinformatics and Computational Biology" unter dem Titel "The New Classification Scheme of the Genetic Code, its Early Evolution, and tRNA Usage" veröffentlichen. Annahmen, dass die Purin - Pyrimidin Kodierung auch Muster auf höherer DNA Ebene zeigen würde, waren richtig. In einer Analyse der DNA Bindemotive untersuchten wir das Vorkommen binärer Muster. Eine umfassende Statistik über die größte Klasse von Restriktionsenzymen ergab tatsächlich ein sehr ausgeprägtes und einfaches Purin-Pyrimidin Muster und einen erhöhten **G+C** Gehalt. Für beide Beobachtungen fanden wir mehrere Erklärungen, welche ausführlich in dem Artikel "Common Patterns in Type II Restriction Enzyme Binding Sites" veröffentlicht in "Nucleic Acid Research" diskutiert wurden. Die gefundenen Muster könnten beispielsweise die Bindung des Proteins unterstützen und das Finden der Bindestelle erleichtern.

Im letzten Teil meiner Arbeit werden zwei eingereichte Artikel zur Analyse Boolescher Funktionen vorgestellt. Boolesche Funktionen lassen die Beschreibung und Analyse komplizierter dynamischer Prozesse zu und erleichtern damit das Finden übergeordneter binärer Muster in biochemischen Interaktionsnetzwerken. Es ist bekannt, dass nicht alle Funktionen nötig sind, um biologisch relevante Geninteraktionsnetzwerke zu beschreiben. Im Artikel mit dem Titel "Boolean Networks with Biologically Relevant Rules Show Ordered Behavior", eingereicht in "BioSystems" konnten wir durch Analyse gemessener Daten zeigen, dass sich die Klasse benötigter Boolescher Funktionen stark einschränken lässt. Weiterhin gelang uns die Berechnung der exakten Anzahl hierarchisch kanalisierender Funktionen, welche oft in realen Geninteraktionsnetzwerken vorkommen. In der Arbeit "The Decomposition Tree for Analysis of Boolean Functions", eingereicht in "Journal of Complexity", stellen wir eine effiziente Datenstruktur zur Klassifizierung und Analyse Boolescher Funktionen vor. Diese erlaubt uns die Erkennung biologisch relevanter Funktionsklassen in Polynomialzeit.

Contents

Abstract	1
Zusammenfassung	3
Introduction	7
New Classification Scheme of the Genetic Code	14
A new classification scheme of the genetic code. Thomas Wilhelm and Swetlana Nikolajewa, 2004. <i>Journal of Molecular Evolution</i> , volume 59, pp. 598-605.	15
A purine-pyrimidine classification scheme of the genetic code. Thomas Wilhelm and Swetlana Nikolajewa, 2004. <i>BIOforum Europe Journal</i> , volume 6, pp. 46-49.	22
The new classification scheme of the genetic code, its early evolution, and tRNA usage. Swetlana Nikolajewa, Maik Friedel, Andreas Beyer and Thomas Wilhelm, 2006. <i>Journal of Bioinformatics and Computational Biology</i> , accepted to the publication on December, 22 nd , 2005, in press.	25
Common Patterns in Type II Restriction Enzyme Binding Sites	38

Common patterns in type II restriction enzyme binding sites. Svetlana Nikolajewa, Andreas Beyer, Maik Friedel, Jens Hollunder and Thomas Wilhelm, 2005. <i>Nucleic Acid Research</i> , volume 33(8), pp. 2726-2733.	39
Pattern Analysis of Gene Regulatory Rules	47
Boolean Networks with biologically relevant rules show ordered behavior. Svetlana Nikolajewa, Maik Friedel, and Thomas Wilhelm. Submitted to the <i>BioSystems</i> on November, 18 th , 2005	47
The Decomposition Tree for analysis of Boolean functions. Maik Friedel, Svetlana Nikolajewa and Thomas Wilhelm. Submitted to the <i>Journal of Complexity</i> on March, 2 nd 2006	62
Discussion	79
Bibliography	89
Anhang	95
Angabe zum Eigenanteil	95
Lebenslauf	96
Publikationen	97
Vorträge	99
Poster	100
Erklärung	101
Danksagung	102

Introduction

*There are 10 kinds of people in the world -
those who understand binary numbers,
and those who don't. Binary humor.*

Since the discovery of the DNA a huge amount of biological data has been produced by molecular biologists, comprising data from genome sequencing, mRNA expression and different protein structures. Most of them are not yet analyzed. To handle these large datasets, new algorithms and methods are needed. The new scientific discipline bioinformatics is required to solve the actual problems of molecular biology, by applying "informatics" techniques (Lucombe *et al.*, 2001). Translating biological information into a digital form allows for the analysis of DNA sequences, the prediction of protein structures and the simulation of macromolecular or metabolic dynamics.

One of the most important tasks in bioinformatics is the identification of smaller and larger patterns, to classify and reduce the biological data and even more important, to explain the principles behind the biological information. Motif discovery in sequential data has widespread applications (Rigoutsos and Floratos, 1998) in predicting the location and structure of genes (Zhang, 2002), in searching for DNA binding sites (Stormo, 2000), in the discovery of drug targets (Hoag, 2006) and in understanding the mechanisms of alternative splicing (Hiller *et al.*, 2005).

Pattern recognition is a scientific field which aim is to classify data based on either *a priori* knowledge or on statistical information extracted from the patterns.

Usually the patterns are groups of experiments, measurements or observations, defining a set of points in an appropriated multidimensional space.

Pattern recognition is not only restricted to bioinformatics, it can also be applied in nearly all fields of science. Other important application areas are image analysis, character recognition, speech analysis, person identification and industrial inspection.

In biology, patterns can be found on every level, ranging from patterns in behavioral biology (top level), down to patterns in molecular genetics (bottom level). At the DNA strand itself patterns can be seen in a "bottom up" or "top down" approach. This means that there are different levels for analyzing DNA patterns. One can either start from the bottom e.g. patterns in the reading frame itself, and end at the highest level e.g. patterns in DNA packing and chromosomal structure (Allen *et al.*, 2006) or vice versa.

The focus of my PhD thesis is the identification and analysis of patterns at different levels within molecular biology, including the organization and evolution of the genetic code, the analysis of short specific DNA binding sequences and the interpretation of important patterns within the dynamics of gene interaction networks. In the first two parts I concentrate on finding patterns using a "bottom-up" approach. Whereas the first chapter of my work deals with the identification and analysis of patterns in the genetic code.

The information concerning heredity, structural and functional features of all living things is stored in deoxyribonucleic acid (DNA). The DNA molecule is a long polymer¹, composed of four different nucleotides: adenine (A), guanine (G), cytosine (C) and thymine (T). This four letters or nucleotide bases are enough to code and carry out the whole genetic information in bacteria, as well as in human. Unlike DNA, RNA (ribonucleic acid) is almost always a single-stranded molecule and has a much shorter chain of nucleotides, where the base thymine is replaced by uracil (U).

¹The DNA of the longest human chromosome is comparable with the distance between Hamburg and Bremen (100 km long) (Calladine *et al.*, 2006).

The genetic information from DNA, through messenger RNA, is translated into amino acid sequences of proteins, according to the rules, known as the *genetic code*. In 1959 Nirenberg began a series of elegant experiments, adding an artificial form of poly-uracil RNA in a cell of *E. coli*. He extracted the radioactively labeled phenylalanine in the resulting protein. This is known to be the first step in the discovery of the genetic code.

During translation a triplet of nucleotides (known as codon) translates into a single amino acid. The **AUG** codon encodes the amino acid methionine and indicates the start of the translation. Whereas the triplets **UAA**, **UAG**, and **UGA** stand for a STOP signal of the translation. Table 1 shows the standard genetic code.

		2nd base					
		U	C	A	G		
1st base	U	UUU <i>Phe</i>	UCU <i>Ser</i>	UAU <i>Tyr</i>	UGU <i>Cys</i>	U C A G	
		UUC <i>Phe</i>	UCC <i>Ser</i>	UAC <i>Tyr</i>	UGC <i>Cys</i>		
		UUA <i>Leu</i>	UCA <i>Ser</i>	UAA <i>Stop</i>	UGA <i>Stop</i>		
		UUG <i>Leu</i>	UCG <i>Ser</i>	UAG <i>Stop</i>	UGG <i>Trp</i>		
	C	CUU <i>Leu</i>	CCU <i>Pro</i>	CAU <i>His</i>	CGU <i>Arg</i>	U C A G	
		CUC <i>Leu</i>	CCC <i>Pro</i>	CAC <i>His</i>	CGC <i>Arg</i>		
		CUA <i>Leu</i>	CCA <i>Pro</i>	CAA <i>Gln</i>	CGA <i>Arg</i>		
		CUG <i>Leu</i>	CCG <i>Pro</i>	CAG <i>Gln</i>	CGG <i>Arg</i>		
	A	AUU <i>Ile</i>	ACU <i>Thr</i>	AAU <i>Asn</i>	AGU <i>Ser</i>	U C A G	
		AUC <i>Ile</i>	ACC <i>Thr</i>	AAC <i>Asn</i>	AGC <i>Ser</i>		
		AUA <i>Ile</i>	ACA <i>Thr</i>	AAA <i>Lys</i>	AGA <i>Arg</i>		
		AUG <i>Met</i>	ACG <i>Thr</i>	AAG <i>Lys</i>	AGG <i>Arg</i>		
	G	GUU <i>Val</i>	GCU <i>Ala</i>	GAU <i>Asp</i>	GGU <i>Gly</i>	U C A G	
		GUC <i>Val</i>	GCC <i>Ala</i>	GAC <i>Asp</i>	GGC <i>Gly</i>		
		GUA <i>Val</i>	GCA <i>Ala</i>	GAA <i>Glu</i>	GGA <i>Gly</i>		
		GUG <i>Val</i>	GCG <i>Ala</i>	GAG <i>Glu</i>	GGG <i>Gly</i>		

Table 1. Table of the standard genetic code. The yellow regions indicate *family codons*, where the encoded amino acid is independent of the third codon position.

For biologists and life scientists one of the most fascinating questions is: Is there an underlying principle for this redundant codon - amino acid assignment (Crick, 1968; Knight, 1999; Di Giulio, 2005)? Many tried to discover such princi-

ples and there are already some patterns known e.g.: the special role of the middle base in codons (Taylor and Coates, 1989; Woese *et al.*, 2000; Knight, 1999; Copley *et al.*, 2005) or the family codon distribution in the genetic code (Lagerkvist, 1978; Halitsky, 2003).

Many different representations have been published, all of them try to find a logical organization of the code. Bashford and Jarvis (2000) presented the genetic code as "...low order polynomials of the 6 coordinates in the 64-dimensional codon weight space". Using the "Leibniz Number" Morimoto (2002) described the genetic code as a cube-shaped periodic table. Reflecting the periodicity in amino acids and symmetrical order, with respected to the xy -plane it allows a partial explanation for some deviations of non-standard genetic codes and for some predictions about potential candidates of non-standard codons to be discovered in the future.

There are also many mathematical models that encode the bases in a binary manner. Based on the coding $\mathbf{A}=00$, $\mathbf{G}=01$, $\mathbf{U}=10$, $\mathbf{C}=11$, Jimenez *et al.* (1996) defined a codon with six binary variables, where the binary code is equivalent to a Boolean hypercube. Stambuk (2000) showed the "universal metric characteristics of the genetic code" on a square with the four bases as vertices, encodes as: $\mathbf{A}=11$, $\mathbf{G}=10$, $\mathbf{U}=00$, $\mathbf{C}=01$. Karasev and Stefanov (2001) proposed a model of topological protein coding, using another binary coding: $\mathbf{A}=11$, $\mathbf{G}=01$, $\mathbf{U}=10$, $\mathbf{C}=00$. The Gray code of He *et al.* (2004) is based on: $\mathbf{A}=01$, $\mathbf{G}=11$, $\mathbf{U}=10$ and $\mathbf{C}=00$, and Sánchez *et al.* (2004) proposed the Hasse diagram of the genetic code, which relies on the correspondence between the codon order and the biochemical properties of amino acids. The Hasse diagram is equivalent to a sixth-dimensional Boolean hypercube with vertices representing the codons, generalizing other Boolean models by using two dual Boolean lattice in the coding. Interestingly, in this structure the hydrophobicity and hydrophilicity of amino acids are reflected by using the Hamming distance between the binary representation of the codons.

The basic principle of the enumerated mathematical models is the binary coding of a single base. Because of the 4 bases in DNA (RNA) two bits are needed to encode a single nucleotide. [Claude E. Shannon \(1948\)](#) first introduced the word *bit*², which is the minimal unit of information. Generally, one bit is the quantity of information required to distinguish two mutually exclusive states. The binary approach in biology also yields useful results in gene expression analysis ([Walker et al., 1999](#)), in transgenic binary expression systems to study embryogenesis ([Noramly et al., 2005](#)), and "... the simple binary- switch nature of asymmetry variation" in phylogenetic analysis offers an attractive focus for comparative studies on evolutionary biology ([Palmer, 2004](#)).

The aim of the second part of my PhD thesis is to find a common sequence motif in a collection of restriction enzyme recognition sequences, to understand the mechanism of the specific protein - DNA binding. A *restriction enzyme* (or *restriction endonuclease*) is an enzyme that cuts double-stranded DNA. Restriction enzymes were first discovered in the late 1960s by [Arber and Linn \(1969\)](#). They isolated two types of enzymes that were responsible for phage growth restriction³ in *E. coli*. Nowadays restriction enzymes are one of the most important tools in biotechnology, because they cut DNA at short specific sites, rather than at random sites along the length of the DNA molecule. Based on the composition of subunits, cofactor requirements, site specificity and mechanism of recognition and cleaving, they are classified into four types. Enzymes of type I, II and III are parts of restriction-modification systems. They additionally contain methyltransferases, adding methyl groups to cytosine or adenine in the host DNA, to prevent cleaving own DNA ([Roberts et al., 2003](#)). The restriction enzymes of type II cleave the DNA within or close to their recognition site and does not require ATP hydrolysis, that makes the type II restriction enzymes the most used enzymes. The orthodox type II REases are homodimers, that recognize palindromic sequences of 4-8 bp. A palindromic sequence is a sequence that reads the same on the complementary strand. In Table 2 five type II restriction enzymes with their

²A bit stands for a binary digit (or binary unit).

³The term restriction comes from this observation.

palindromic sequences are demonstrated. Although the recognition sequences are diverse they show very similar purine - pyrimidine patterns.

Restriction Enzyme	Source	Recognition Sequence*	Purine(1)-pyrimidine(0) pattern
AluI	<i>Arthrobacter luteus</i>	AG↓CT	1100
HaeIII	<i>Haemophilus aegyptius</i>	GG↓CC	1100
BamHI	<i>Bacillus amyloliquefaciens</i>	G↓GA TCC	111000
HindIII	<i>Haemophilus influenzae</i>	A↓AG CTT	111000
EcoRI	<i>Escherichia coli</i>	G↓AA TTC	111000

Table 2. Examples of type II restriction enzymes, taken from [Kimballs Biology Pages](#).

* The arrow ↓ indicates the cleave position of a restriction enzyme.

The third part of my work deals with the pattern analysis of transcriptional gene regulation. A big challenge in the postgenomic era is to understand the cellular phenomena arising from the interaction of genes and proteins ([Hastly *et al.*, 2002](#)). Based on signaling pathways, [Monod and Jacob \(1961\)](#) predicted the fundamental processes of the cell: differentiation and protein regulation. After this prediction many mathematical models appeared, describing the gene regulation. But the simulation of networks on many interacting genes is too complex for making qualitatively high and comprehensive predictions. The Boolean approach of [Kauffman \(1969\)](#) can be used to study simplified gene regulatory networks, consisting of nodes which are genes or proteins and directed connections, representing the interactions between them. In the [Kauffman \(1969\)](#) model a gene can either be 'on' or 'off', and for the description of the gene interactions he used Boolean rules⁴.

The larger the number of genes, the more difficult the Boolean rules involved in the gene regulation. Given six genes, there are already 2^{2^6} or over 18 trillions different rules to activate a seventh gene. [Kauffman \(1993\)](#) suggested that gene

⁴A Boolean function allows to combine information from two or more bits, using binary logical operations: \wedge (AND), \vee (OR) and the unary operation NOT.

regulatory rules must have special patterns, that are responsible for the stable dynamical behavior of cellular regulatory networks, and at the same time restrict the number of appropriated Boolean rules. He extracted an important subclass of Boolean functions, named canalyzing functions. A canalyzing function is a Boolean function, which has at least one input, such that for at least one input value, the output value is fixed (Kauffman *et al.*, 2003). Simulating the ordered and chaotic dynamical behavior of a Boolean model, Szallasi and Liang (1998) discovered the class of hierarchically canalyzing functions, which are a natural extension of canalyzing functions. In hierarchically canalyzing functions the inputs are canalyzing in a hierarchical manner. Harris *et al.* (2002) compiled a set of transcriptional regulatory rules, which were observed in experiments. Analyzing the patterns in these Boolean rules, Kauffman *et al.* (2003) found out, that nearly all of them belong to hierarchically canalyzing functions. From this observations it follows that a simple principle has to underlay the natural occurring gene interaction rules, which support the stability and robustness of gene regulatory networks. In part three of my work we tried to validate this hypothesis.

New Classification Scheme of the Genetic Code

The paper of [Wilhelm and Nikolajewa \(2004a\)](#) provides the first version of the new classification scheme of the genetic code, based on a binary purine - pyrimidine representation of a codon. In the second paper of [Wilhelm and Nikolajewa \(2004b\)](#) the final and optimal form of the new classification scheme is presented, where the column order of the new scheme is fixed. In the third paper [Nikolajewa *et al.* \(2006\)](#) the tRNA anticodon usage pattern is examined, which is related to the codon-reverse codon symmetry in the genetic code. Moreover, we proposed a new hypothesis about the evolution of translation, which we called "reverse recognition conjecture".

J Mol Evol (2004) 59:598–605
DOI: 10.1007/s00239-004-2650-7

JOURNAL OF **MOLECULAR
EVOLUTION**

© Springer Science+Business Media, Inc. 2004

A New Classification Scheme of the Genetic Code

Thomas Wilhelm, Svetlana Nikolajewa

Institute of Molecular Biotechnology, Beutenbergstr. 11, 07745 Jena, Germany

Received: 3 October 2003 / Accepted: 21 May 2004

Abstract. Since the early days of the discovery of the genetic code nonrandom patterns have been searched for in the code in the hope of providing information about its origin and early evolution. Here we present a new classification scheme of the genetic code that is based on a binary representation of the purines and pyrimidines. This scheme reveals known patterns more clearly than the common one, for instance, the classification of strong, mixed, and weak codons as well as the ordering of codon families. Furthermore, new patterns have been found that have not been described before: Nearly all quantitative amino acid properties, such as Woese's polarity and the specific volume, show a perfect correlation to Lagerkvist's codon–anticodon binding strength. Our new scheme leads to new ideas about the evolution of the genetic code. It is hypothesized that it started with a binary doublet code and developed via a quaternary doublet code into the contemporary triplet code. Furthermore, arguments are presented against suggestions that a “simpler” code, where only the mid-base was informational, was at the origin of the genetic code.

Key words: Genetic code — Origin of life — Doublet code — Pattern — Amino acid properties

Introduction

Crick (1968) introduced the notion that the genetic code is simply the result of pure chance or a “frozen accident” and that it therefore does not need any further evolutionary explanation. Later, this view was questioned. Although certain knowledge of the origin and early stages of life is not likely to be obtained, there are some hints of possible evolutionary scenarios of the genetic code. One direction of research (the “top-down approach” [Szathmary 1999]) analyzes patterns in the contemporary code (Knight and Landweber 1998; Szathmary 1999) and tries to infer appropriate chemical and selective forces. The bottom-up approach, on the other hand, is rooted in biochemistry and aims at constructing plausible scenarios for the origin of coding (Topal and Fresco 1976; Maizels and Weiner 1987; Szathmary 1993).

It has been appreciated for a long time that the genetic code assigns similar amino acids to similar codons (Sonneborn 1965; Woese 1965; Zuckerkandl and Pauling 1965; Crick 1968). Two different rationales have been presented: first, mutation (Sonneborn 1965; Zuckerkandl and Pauling 1965) and translation (Woese 1967; Haig and Hurst 1991; Freeland and Hurst 1998) error minimization (or both (Ardell and Sella 2002)) and, second, the tendency of similar amino acids to directly interact with similar RNA sequences (Woese et al. 1966; Yarus 1998, 2000). Landweber and coworkers found further evidence to support both hypotheses. Extending previous work (Haig and Hurst 1991; Freeland and Hurst 1998) by quantifying amino acid similarity, these authors were able to show that “the canonical code is at or very close to a global optimum for error minimization”

Correspondence to: Thomas Wilhelm; email: wilhelm@imb-jena.de

(Freeland et al. 2000). Based on the earlier work of Yarus (cf. Yarus 1998, 2000), by doing a statistical analysis of RNA aptamers (nucleic acid molecules selected to bind specific ligands), they concluded that there is “the strongest support for an intrinsic affinity between any amino acid and its codons” (Knight and Landweber 1998). It has also been proposed that instead of the actual codons, some derivatives of them, such as the anticodons (Dunnill 1966; Jungck 1978) or codon–anticodon duplexes (Alberti 1997), were the original amino acid binding motifs. It could also be that the original amino acid recognition took place at the tRNA acceptor stem (Hopfield 1978) or that the specificity of aminoacylation is determined by the interaction of the tRNA synthetase with its tRNA (Weiner and Maizels 1987). Szathmary (1999) proposed that amino acid RNA allocation took place even before the appearance of tRNA. He also gave a possible evolutionary scenario for the development of an anticodon hairpin to a longer structure with an operational code at the acceptor stem.

Several patterns of the genetic code have been identified, which can be illustrated within the classical scheme.

The Common Scheme of the Genetic Code

The common scheme of the genetic code (Alberts et al. 2002) contains $4^3 = 64$ codons, a three-dimensional matrix where each dimension represents one of the three positions in the triplet code (Fig. 1). Viewed this way, some patterns emerge: The first codon position seems to be correlated with amino acid biosynthetic pathways (Wong 1975; Taylor and Coates 1989) and with their evolution as evaluated by synthetic “primordial soup” experiments (Eigen 1977; Schwemmler 1994). The second position is correlated with the hydrophobic properties of the amino acids (Crick 1968; Wolfenden et al. 1979; Taylor and Coates 1989), and the degeneracy of the third position could be related to the molecular weight or size of the amino acids (Hasegawa and Miyata 1980; Taylor and Coates 1989).

Lagerkvist (1978, 1981) divided the common illustration scheme (Fig. 1) into a left part (containing the first and second columns, i.e., U and C in the second position of the codon, respectively) and a right part (the third and fourth columns, i.e., A and G in the second position). He observed that codon families (the amino acid of a codon family is uniquely determined by the first two nucleotides of a codon; cf. shaded regions in Fig. 1) have a much higher probability to appear in the left part. Furthermore, he found that “strong” codons (the first two nucleotides in the codon are G and/or C) always represent codon families, while “weak” codons (A and/or U as the

first two nucleotides) never do so. “Mixed” codons in the right part of the scheme never represent codon families, whereas mixed codons in the left part always stand for a codon family. Lagerkvist (1978) speculated “that interactions between mixed codons and their anticodons are stronger in the left half of the codon square.”

However, most amino acid properties show no clear pattern in the common scheme of the genetic code. Instead Jungck (1978) used 15 different quantitative measures of amino acid properties such as polarity and molecular volume to demonstrate that these properties are generally more closely correlated with anticodon than with codon dinucleoside monophosphate properties. This supports the hypothesis that the relationship between amino acids and their anticodon dinucleosides was the basis for the origin of the genetic code.

In this article we follow the “top-down approach” toward understanding the organization of the genetic code. We are thereby led to propose a new classification scheme for the code that helps us to identify new patterns which in turn suggest new speculations about its origin.

Results

A New Classification Scheme of the Genetic Code

Figure 2 shows our new scheme for presenting the genetic code. It is based on a binary classification of nucleic acid bases. The two components of all nucleic acids, purines and pyrimidines, are denoted 1 and 0, respectively. The eight rows in Fig. 2 represent the $2^3 = 8$ possible combinations of three binary digits. Since there are two purines (A, G) and two pyrimidines (U, C) for each row, there again exist eight possibilities.

Our first observation is that four (and not eight) columns are sufficient to place all 20 amino acids, as well as the termination codons. Each row contains exactly four different amino acids (including the termination codon). In the standard code, exceptions are the second row, with two leucines, and the AU* start codon in the fourth row. Note that here are also the deviations from the standard code. Interestingly, the yeast mitochondrial code shows no exception: Each row contains exactly four different entries in four different columns. In this respect the yeast mitochondrial code is the most regular one. The fact that in our scheme four columns are sufficient reflects the well-known fact that if the third position is important (in exactly half of our table this is not the case), then it is only decisive if there is either a purine (1) or a pyrimidine (0) (Fitch and Upper 1987), i.e., the third position is analyzed in a binary manner

600

First Letter	Second Letter				Third Letter				
	U	C	A	G					
U	UUU	Phe	UCU	Ser	UAU	Tyr	UGU	Cys	U
	UUC		UCC		UAC		UGC		C
	UUA	Leu	UCA	Ser	UAA	Stop	UGA	Stop	A
	UUG		UCG		UAG		UGG	Trp	G
C	CUU	Leu	CCU	Pro	CAU	His	CGU	Arg	U
	CUC		CCC		CAC		CGC		C
	CUA		CCA	Pro	CAA	Gln	CGA		A
	CUG	Leu	CCG		CAG		CGG	Arg	G
A	AUU	Ile	ACU	Thr	AAU	Asn	AGU	Ser	U
	AUC		ACC		AAC		AGC		C
	AUA	Ile	ACA	Thr	AAA	Lys	AGA	Arg	A
	AUG	Met	ACG		AAG		AGG		G
G	GUU	Val	GCU	Ala	GAU	Asp	GGU	Gly	U
	GUC		GCC		GAC		GGC		C
	GUA	Val	GCA	Ala	GAA	Glu	GGA	Gly	A
	GUG		GCG		GAG		GGG		G

Fig. 1. The common presentation of the standard (“universal”) genetic code. All deviations from this code (Elzanowski and Ostell 2000) are thought to be the result of later mutations (Osawa et al. 1992; Knight and Landweber 2000b; Knight et al. 2001). Shaded regions show codon families.

(Taylor and Coates 1989). This has been explained by Crick’s (1996) wobble hypothesis, wherein the first two nucleotides of the codon pair with their anticodon bases according to Watson–Crick rules, but the third base pairs according to the wobble rules, which say that G can also pair with U, for instance. The third codon position is exclusively analyzed in a binary manner in the mitochondrial codes of yeast, vertebrates, invertebrates, coelenterates, and flatworms, as well as in the codes of mold, protozoa, and mycoplasma/spiroplasma; for the other codes there are a few exceptions (cf. Elzanowski and Ostell 2000). Note that these few exceptions always have a purine at the third position of the codon (e.g., AUA [Ile] and AUG [Met] in the standard code).

Our scheme provides some support for the “adaptive genetic code” hypothesis (Freeland 2002), which states that the code has evolved to minimize the deleterious effects of mutation and translation error (Haig and Hurst 1991; Freeland and Hurst 1998). The purine–pyrimidine binary coding scheme, shown in Fig. 2, exhibits a much greater regularity than a binary coding according to the base pairs (A,U—1; G,C—0). This corresponds to the known fact that transition mutations (e.g., purine A vs. purine G) occur more frequently than transversion mutations (e.g., purine A vs. pyrimidine U).

A second observation concerns the order of the columns. In the first column the first two positions are G and C. These always pair with their anticodon base via three hydrogen bonds, i.e., the first two bases together always guarantee six hydrogen bonds. For

that reason Lagerkvist (1978) called them strong codons. In the second and third columns, the first two bases guarantee five bonds (mixed codons), and in the fourth column just four bonds (weak codons). This pattern corresponds very well to the importance of the third base in the triplet codon: If the first bases are G and/or C (first column), the third base is never important, and in the second and third columns, the third base is important in exactly half the cases (if there is a purine in the second position—lower half of the table). In the fourth column the third base is always necessary for the determination of the correct amino acid. In Fig. 2, the order of codon families is illustrated by the shaded regions. It seems that for the first column, the first two bases alone guarantee sufficient stability in the codon–anticodon pairing to ensure the correct choice of the amino acid. In the case of mixed codons (second and third columns) a codon family is guaranteed if there is a pyrimidine in the second position. Going beyond Lagerkvist’s counting of hydrogen bonds, others have provided some quantitative information about nucleotide binding strengths (Ornstein and Fresco 1983).

A third observation refers to two perfect symmetries in our scheme. The first is the codon–anticodon symmetry: The thick horizontal line in Fig. 2 marks the symmetry axis. For instance, codon CCC (Pro; first column, first row) has the anticodon GGG (Gly; first column, last row), and codon ACG (Thr; third column, fourth row) has the anticodon UGC (Cys; third column, fifth row). The second perfect symmetry is the point symmetry corresponding to Halitsky’s (2003) family–nonfamily symmetry operation (“E–M bifurcation”), indicated by the point in the center of Fig. 2. Halitsky observed that all 32 “family codons” CC*, CU*, UC* GC*, GU*, AC*, CG*, and GG* can be mapped into the 32 “nonfamily codons” UU*, AU*, CA*, UG*, UA*, GA*, AG*, and AA* by exchanging the two amino bases A and C with one another and the two keto bases U and G with one another. For instance, the family codon GUA (Val) is mapped into the nonfamily codon UGC (Cys). Thus, this point symmetry underlies the family–nonfamily symmetry in our scheme (shaded vs. unshaded regions).

A fourth observation concerns the deviations of nonstandard genetic codes. As can be seen in Fig. 2, nearly all deviations occur in codons with a purine at the third position. The only exception is the yeast mitochondrial code, in which CU* does not code for Leu but, rather, for Thr.

Our fifth observation refers to the number of different tRNAs. The mammalian mitochondrial genomes contain one gene for each tRNA, with the exceptions of tRNA Leu and tRNA Ser for which two genes are present. Our new classification scheme for these mitochondrial codes (slight modification of Fig. 2) makes this number obviously: eight tRNAs

Code	Strong codons 6 hydrogen bonds	Mixed codons 5 hydrogen bonds (first G or C)	Mixed codons 5 hydrogen bonds (first U or A)	Weak codons 4 hydrogen bonds
000	Pro CC (C/U)	Leu CU (C/U) 1/1	Ser UC (C/U)	Phe UU (C/U)
001	Pro CC (G/A)	Leu CU (G/A) 2/1	Ser UC (G/A) 0/1	Leu UU (G/A) 0/1
100	Ala GC (C/U)	Val GU (C/U)	Thr AC (C/U)	Ile AU (C/U)
101	Ala GC (G/A)	Val GU (G/A)	Thr AC (G/A)	Met/Ile AU (G/A) 0/5
010	Arg CG (C/U)	His CA (C/U)	Cys UG (C/U)	Tyr UA (C/U)
011	Arg CG (G/A)	Gln CA (G/A)	Trp/Stop UG (G/A) 0/9	Stop UA (G/A) 4/2
110	Gly GG (C/U)	Asp GA (C/U)	Ser AG (C/U)	Asn AA (C/U)
111	Gly GG (G/A)	Glu GA (G/A)	Arg AG (G/A) 6/6	Lys AA (G/A) 0/3

Fig. 2. A new classification scheme of the standard genetic code based on a binary representation of purines (1) and pyrimidines (0). The third base is given in parentheses. When there are differences between the standard code and any other code, the number of deviations from the standard code is indicated. This comparison is based on 16 nonstandard codes (Elzanowski and Ostell 2000). For instance, in the UG(G/A) field, 0/9 indicates that UGG encodes for Trp in all codes, but UGA is not the termination codon in 9 of the 16 nonstandard codes: In 8 different mitochondrial codes UGA encodes Trp, and in the euplotid nuclear code it represents Cys. It is

interesting that at least in some bacteria the 21st amino acid, selenocysteine, can also be encoded by UGA (Osawa et al. 1992; Thanbichler and Böck 2002). Another example is the CU(G/A) field. In the yeast mitochondrion CUG and CUA encode Thr; in the alternative yeast nuclear code, CUG represents Ser. Shaded region show codon families. The point in the center indicates the perfect point symmetry in this scheme, according to Halitsky's (2003) family–nonfamily symmetry operation. The thick horizontal line marks the symmetry axis for codon–anticodon symmetry.

for the eight codon families plus 14 tRNAs for the remaining 14 fields (the two termination codons need no tRNA).

Our sixth observation shows hitherto unknown regularities of amino acid properties. Jungck (1978) collected 15 different measures of amino acid properties, as well as 3 measures for dinucleoside monophosphates. For all of these 18 measures we arranged a table with eight rows and four columns corresponding to the scheme in Fig. 2. For AU(G/A) we took the Met values (e.g., vertebrate mitochondrial code); for UA(G/A), the Tyr values (mitochondrial flatworm code). Then we analyzed all row and column sums. The row sums show a strong monotonicity just for the three dinucleoside monophosphate measures and for the hydrophobicity measure of Levitt (1976). However, amazingly, the column sums of nearly all measures are perfectly correlated with the corresponding codon–anticodon binding strength as defined by Lagerkvist (1978, 1981), in the following simply denoted codon strength. This is demonstrated in Table 1. For this table we averaged the column sums of the second and third columns, giving one “mixed codons” column. As can be seen in Table 1 there are just two exceptions. In the polarity measure of Zimmerman et al. (1968), the deviation is

very weak, and in contradiction to all other measures, here the values for the amino acids vary by orders of magnitude. A problem only arises for the three hydrophobicity measures: The two monotonic measures “Levitt” and “BullBreese” are anticorrelated, and the “Jones” measure is not monotonic. The anticorrelation was found by Jungck (1978), but he did not comment on this.

The fact that the order of the second and third columns is not fixed is also underlined by individual consideration of the two mixed codon columns, instead of the averaging done in Table 1. In about half of the cases the order of the second and third columns should be exchanged to guarantee the strong monotonicity of the amino acid measures as a function of the column number.

The strong correlation between amino acid properties and codon strength implies that the first and second position together, and not one of them alone, must have been important for the amino acid–codon assignment in the evolution of the genetic code.

Evolution of the Genetic Code

What do the observed patterns tell us about the evolution of the genetic code? The so-called biosyn-

602

Table 1. Correlation of codon strength and amino acid properties

Measure	Strong codons	Mixed codons	Weak codons
Dinucleoside monophosphates			
Hydrophilicity			
Weber & Lacey (1978)	1.686	1.434	1.235
Barzilay et al. (1973)	2.72	2.26	2.26
Hydrophobicity (Garel et al. 1973)	2.556	3.413	3.982
Amino acids			
Molec. weight (handbook value)	907	1065.6	1217.5
Molec. volume (Grantham 1974)	381	637.5	906
Refractivity (Jones 1975)	83.86	140.03	186.51
Alpha pK1 (Zimmermann et al. 1968)	16.96	17.11	17.43
Bulkiness (Zimmermann et al. 1968)	93.22	124.345	143.54
Specific volume (McMeekin et al. 1964)	5.26	5.37	5.8
Polarity			
Zimmerman et al. (1968)	107.16	109.58	58.14
Woese et al. (1967)	61.2	59.15	51
Grantham (1974)	71.2	67	56.3
Hydrophobicity			
Jones (1975)	9.18	8.385	16.93
Levitt (1976)	-2.2	1.6	8.8
Bull & Breese (1974)	3880	-165	-6790
Hydrophilicity (Weber & Lacey 1978)	7.02	6.585	5.59
Partition coefficient (Garel et al. 1973)	1.88	5.58	7.6
Sequence frequency (Jungck 1971)	4280	3522	2966

Note. Averaged values (per column, in our scheme of Fig. 2) of quantified dinucleoside monophosphate properties (codon and anticodon values give the same average, because of the codon-anticodon symmetry) and amino acid properties for strong, mixed, and weak codons. Each row represents one of the measures published by Jungck (1978).

thetic theory assumes that the genetic code evolved from a simpler form that encoded fewer amino acids (Crick 1968). A special version of this theory has been given by Wong (1975), who proposes that the genetic code coevolved with the invention of biosynthetic pathways for new amino acids. Although it has been shown that his analyses rest on wrong assumptions (Ronneberg et al. 2000), it is generally accepted that one can discriminate evolutionarily old and new amino acids (Alberts et al. 2002). Of course it could be that the binding allocation between nucleic acid molecules (RNAs or even PNAs [Knight and Landweber 2000b]) and amino acids did not start until all 20 amino acids were available, but it seems simpler to assume that as soon as there were amino acids and nucleic acids available, produced abiotically, both began to bind to each other. It now seems clear that “the code probably underwent a process of expansion from relatively few amino acids to the modern complement of 20” (Knight and Landweber 2000b).

Does our scheme yield some hints as to the evolution of the code? We already noted that the third nucleotide is nearly always (two exceptions in the standard code) analyzed just in a binary manner. Taking this for granted, we can reduce our original 8×8 scheme to an 8×4 scheme (shown in Fig. 2). Looking at this scheme, we observe high redundancy for each second row. Therefore, it is tempting to speculate that there was a period during code evolu-

tion when the third position was not needed at all. Assuming this, we can cancel each second row and are left with a pure doublet code that encodes $4 \times 4 = 16$ amino acids (or 15 plus a termination codon). Perhaps, then, a doublet code preceded the triplet code, as has already been speculated (Jukes 1973; Hayes 1998).

Conceivably, codon expansion from doublet to triplet could have arisen before this or, possibly, not until all 16 amino acids were encoded. If one assumes the latter, then it is interesting to postulate for each doublet the corresponding old amino acid. Met (Wong 1975), Trp, Gln, Asn (Knight and Landweber 2000b), and Tyr (Alberts et al. 2002) seem to be newer amino acids. As mentioned above, Szathmary (1999) proposed an evolutionary mechanism of tRNA formation. In principle, this mechanism could also work starting with doublets instead of triplets. It should be possible to gain experimental evidence for a doublet code by studying amino acid-nucleic acid doublet binding in the same way as has been done for triplets. Knight and Landweber (2000a) showed that Arg triplet codons alone significantly associate with arginine binding sites. Perhaps the doublets show a higher specificity.

However, by proposing a doublet code one faces the frameshifting problem. It seems unthinkable that a sudden transition from a two-letter to a three-letter frame ever occurred. Instead, one can imagine a

gradual evolution with an ancient three-letter reading frame where just the first two letters have been analyzed by an ancient translation machinery. However, one then wonders about such inefficient use of coding space. Perhaps the ancient translation machinery, simply for stereochemical reasons, could not analyze a two-letter frame. In this context it is also interesting to note that even our contemporary code is somehow "inefficient": Already a quaternary doublet code can encode 16 amino acids (or 15 plus a termination codon). For just four (or five) further amino acids a third letter is necessary. Of course, this inefficiency has the advantage of robustness enhancing redundancy.

Szathmary (1992, 2003) proposed a model which yields the result that two different base pairs represent an optimal compromise between the overall copying fidelity and the overall reproduction rate (metabolic efficiency). He assumed that the genetic code was developed before evolution invented proofreading. For higher copying fidelity (due to proofreading, etc.), the model predicts that three different base pairs are better than just two. It is tempting to speculate that in the earliest phases of biological evolution with the lowest copying fidelity, just one base pair could have worked as well. (The copying fidelity is always highest for just one base pair. Nevertheless, Szathmary's simple model gives no one-base pair optimum, but a more detailed model for the metabolic efficiency could do so.) So, perhaps, nucleic acid–amino acid mapping started with a binary code. This is in accordance with earlier speculations that the first genetic material contained only a single base-pairing unit (Crick 1968; Orgel 1968). An important argument in this context is the chemical instability of cytosine, so that it may be difficult to establish a genetic system with G–C base pairing (Levy and Miller 1998). Wächtershäuser (1988) proposed an all-purine precursor of nucleic acids. However, for the sake of self-replication it is more obvious to assume a two-letter code that can give rise to complementary base pairing. Jimenez-Sanchez (1995) argued for an early (binary) A–U coding. Recently, a ribozyme composed of only two different nucleotides has been found by *in vitro* evolution that contained the pyrimidine uracil and the purine 2,6-diaminopurine (Reader and Joyce 2002). Note that uracil is the biosynthetic precursor of the pyrimidines cytosine and thymine (the corresponding precursor of the purines adenine and guanine is hypoxanthine).

Of course, a binary encoding also would be the most aesthetic version from a purely mathematical point of view. A binary triplet code would represent just one column in our scheme (Fig. 2). Given the high redundancy between the rows, it is unlikely that this ever happened. However, an even simpler coding, a binary doublet code, seems conceivable. It is

tempting to speculate which four amino acids, one per two consecutive rows, were the first encoded ones. In the first two rows (two pyrimidines, i.e., 00) Ser seems to be the oldest amino acid, and in the third and fourth rows (10), Ala (Wong 1975). On the other hand, the 01 rows obviously contain no really old amino acid, while the 11 rows contain more than one: Gly, Asp, and Glu (Wong 1975).

One could speculate that the termination marker was important from the very beginning and resulted in coding by the 01 binary doublet. It has been noted that the five amino acids coded by G** (Ala, Val, Gly, Asp, Glu) are all at or near the head of the amino acid synthesis pathways (Taylor and Coates 1989) and also the most abundantly formed ones in abiotic synthesis experiments (Miller 1953, 1987). Furthermore, it has been shown recently by extensive statistical analyses that the frequencies of all five G** amino acids are significantly higher in evolutionary conserved residues, and it has been concluded that "these amino acids may have been the first introduced into the genetic code" (Brooks and Fresco 2002, 2003; Brooks et al. 2002). This is also consistent with physicochemical arguments proposing that the first sense codons had the form G** (Eigen and Schuster 1978). However, Gly is biochemically built from Ser, so Ser can be assumed to be prior. It could be that in the beginning of nucleic acid–amino acid assignment, Asp and Glu competed for the 11 doublet. Of course, code transfer from one amino acid to another one might also have occurred (Wong 1975).

Another scenario consistent with a binary doublet code has been given by Fitch's "ambiguity reduction" hypothesis (Fitch and Upper 1987). It states that early in evolution there was an ambiguity in the charging of amino acids to anticodon acceptors: In the first step just *pyrimidine* codons (*0*), coding for hydrophobic amino acids, and *purine* codons (*1*), coding for hydrophilic amino acids, have been distinguished (binary singulet code). In the second step the more refined binary doublet code (00*, 01*, 10*, 11*) evolved.

The idea that the doublet code was just the second state in the evolution of the genetic code, and that this evolution started with just the midbase as coding, has been worked out by others, who termed it the "simplest" code (McClendon 1986; Schwemmler 1994). However, in this hypothesis both old amino acids Ser (UC*) and Ala (GC*), as well as Asp (GA*) and Glu (GA*), cannot be discriminated. We therefore suggest that the first two positions were equally important from the very beginning. Although our suggestion also does not allow discrimination between the related amino acids Asp and Glu, it nevertheless allows discrimination between the functionally divergent amino acids Ser and Ala. A further argument for the evolutionary importance of the first two nucleotides is the

604

strong correlation observed between codon strength and the amino acid properties.

Conclusion

Taylor and Coates (1989) stated that “many parts of the patterns (of the genetic code) have been seen by others but ... it is the synthesis that adds up to the most interesting ... new insights.” In this spirit, we note that in the work presented here different patterns appear more clearly than in the common scheme of the genetic code. An example is Lagerkvist’s (1978) observation that all strong codons represent codon families, while weak codons do not. Mixed codons represent codon families in half of the cases. Our presentation of the code also highlights new patterns, which were not seen before. As summarized in Table 1, nearly all measures of the amino acid properties correlate strongly with the codon strengths. Furthermore, there is perfect codon–anticodon symmetry as well as point symmetry corresponding to the family–nonfamily symmetry operation (Halitsky 2003) in our scheme.

With regard to evolution, we hypothesize that codon assignments started from a binary doublet code (e.g., hypoxanthin and uracil) and developed later to a quaternary doublet code (A, G, C, U); thereafter, expansion to a triplet code took place. Although the third position is needed for correct amino acid recognition, until now it has nearly always been analyzed in a binary manner. The conclusion that code evolution must have started with doublets and not with a single letter is also underlined by the correlation observed here between the properties of amino acids and the strengths of their codons.

Acknowledgments. We thank two anonymous reviewers for many valuable comments and for referring us to relevant literature and A. Beyer, F. Grosse, M. Friedel, and M.-L. Merten for critical reading of the manuscript. This work was supported by Grant 0312704E from the Bundesministerium für Bildung und Forschung.

References

Alberti S (1997) The origin of the genetic code and protein synthesis. *J Mol Evol* 45:352–358
 Alberts B, Johnson A, Lewis J, Raff M, Roberts K, Walter P (2002) *Molecular biology of the cell*. Garland Science, New York
 Ardell DH, Sella G (2002) No accident: Genetic codes freeze in error-correcting patterns of the standard genetic code. *Phil Trans R Soc Lond B* 357:1625–1642
 Barzilay I, Sussman JL, Lapidot Y (1973) Further studies on the chromatographic behaviour of dinucleoside monophosphates. *J Chromatogr* 79:139–146

Brooks DJ, Fresco JR (2002) Increased frequency of cysteine, tyrosine, and phenylalanine residues since the last universal ancestor. *Mol Cell Prot* 1(2):125–131
 Brooks DJ, Fresco JR (2003) Greater GNN pattern bias in sequence elements encoding conserved residues of ancient proteins may be an indicator of amino acid composition of early proteins. *Gene* 303:177–185
 Brooks DJ, Fresco JR, Lesk AM, Singh M (2002) Evolution of amino acid frequencies in proteins over deep time: Inferred order of introduction of amino acids into the genetic code. *Mol Biol Evol* 19(10):1645–1655
 Bull HB, Breese K (1974) Surface tension of amino acid solutions: a hydrophobicity scale of the amino acid residues. *Arch Biochem Biophys* 161:665–670
 Crick FHC (1966) Codon-anticodon pairing: The wobble hypothesis. *J Mol Biol* 19:548–555
 Crick FHC (1968) The origin of the genetic code. *J Mol Biol* 38:367–379
 Dunnill P (1966) Triplet nucleotide–amino acid pairing: A stereochemical basis for the division between protein and nonprotein amino acids. *Nature* 210:1267–1268
 Eigen M (1977) The hypercycle. A principle of natural self-organization. A: Emergence of the hypercycle. *Naturwissenschaften* 64:541–565
 Eigen M, Schuster P (1978) The hypercycle: A principle of natural self-organization. *Naturwissenschaften* 65:341–368
 Elzanowski A, Ostell J (2000) Genetic codes. <http://www3.ncbi.nlm.nih.gov/htbin-post/Taxonomy/wprintgc?mode=t#SG1>
 Fitch WM, Upper K (1987) The phylogeny of tRNA sequences provides evidence for ambiguity reduction in the origin of the genetic code. *Cold Spring Harbor Symp Quant Biol* 52:759–767
 Freeland SJ (2002) The Darwinian genetic code: An adaptation for adapting? *Genet Program Evol Machin* 3:113–127
 Freeland SJ, Hurst LD (1998) The genetic code is one in a million. *J Mol Evol* 47:238–248
 Freeland SJ, Knight RD, Landweber LF, Hurst LD (2000) Early fixation of an optimal genetic code. *Mol Biol Evol* 17:511–518
 Garel JP, Fillion D, Mandel P (1973) Coefficients de partage d’aminoacides, 1978 nucleobases, nucleosides et nucleotides dans un systeme solvant salin. *J Chromatogr* 78:381–391
 Grantham R (1974) Amino acid difference formula to help explain protein evolution. *Science* 185:862–864
 Haig D, Hurst LD (1991) A quantitative measure of error minimization in the genetic code. *J Mol Evol* 33:412–417
 Halitsky D (2003) Extending the (hexa-)rhombic dodecahedral model of the genetic code: The code’s 6-fold degeneracies and the orthogonal projections of the 5-cube as 3-cube. Contributed paper (983-92-151), American Mathematical Society, and personal communication
 Hasegawa M, Miyata T (1980) On the antisymmetry of the amino acid code table. *Orig Life* 10:265–270
 Hayes B (1998) The invention of the genetic code. *Am Sci* 86:8–14
 Hopfield JJ (1978) Origin of the genetic code: A testable hypothesis based on tRNA structure, sequence, and kinetic proofreading. *Proc Natl Acad Sci USA* 75:4334–4338
 Jimenez-Sanchez A (1995) On the origin and evolution of the genetic code. *J Mol Evol* 41:712–716
 Jones DD (1975) Amino acid properties and side-chain orientation in proteins: A cross correlation approach. *J Theor Biol* 50:167–183
 Jukes TH (1973) Possibilities for the evolution of the genetic code from a preceding form. *Nature* 246:22–26
 Jungck JR (1971) Pre-Darwinian and non-Darwinian evolution of proteins. *Curr Mod Biol* 3:307–318
 Jungck JR (1978) The genetic code as a periodic table. *J Mol Evol* 11:211–224

- Knight RD, Landweber LF (1998) Rhyme or reason: RNA-arginine interactions and the genetic code. *Chem Biol* 5:R215–R220
- Knight RD, Landweber LF (2000a) Guilt by association: The arginine case revisited. *RNA* 6:499–510
- Knight RD, Landweber LF (2000b) The early evolution of the genetic code. *Cell* 101:569–572
- Knight RD, Freeland SJ, Landweber LF (2001) Rewiring the keyboard: Evolvability of the genetic code. *Nat Rev Genet* 2:49–58
- Lagerkvist U (1978) “Two out of three”: An alternative method for codon reading. *Proc Natl Acad Sci USA* 75:1759–1762
- Lagerkvist U (1981) Unorthodox codon reading and the evolution of the genetic code. *Cell* 23:305–306
- Levitt M (1976) A simplified representation of protein conformations for rapid simulation of protein folding. *J Mol Biol* 104:59–107
- Levy M, Miller SL (1998) The stability of the RNA bases: Implications for the origin of life. *Proc Natl Acad Sci USA* 95:7933–7938
- Maizels N, Weiner AM (1987) Peptide-specific ribosomes, genomic tags, and the origin of the genetic code. *Cold Spring Harb Symp Quant Biol* 52:743–749
- McClendon JH (1986) The relationship between the origins of the biosynthetic paths to the amino acids and their coding. *Orig Life* 16:269–270
- McMeekin TL, Groves ML, Hipp NJ (1964) Refractive indices of amino acids, proteins and related substances. In: Stekol JA (ed) *Amino acids and serum proteins*. American Chemical Society, Washington, DC, pp 54–66
- Miller SL (1953) Production of amino acids under possible primitive earth conditions. *Science* 117:528–529
- Miller SL (1987) Which organic compounds could have occurred on the prebiotic earth? *Cold Spring Harb Symp Quant Biol* 52:17–27
- Orgel LE (1968) Evolution of the genetic apparatus. *J Mol Biol* 38:381–393
- Ornstein RL, Fresco JR (1983) Correlation of T_m, sequence, and H of complementary RNA helices and comparison with DNA helices. *Biopolymers* 22:2001–2016
- Osawa S, Jukes TH, Watanabe K, Muto A (1992) Recent evidence for the evolution of the genetic code. *Microbiol Rev* 56(1):22–264
- Reader JS, Joyce GF (2002) A ribozyme composed of only two different nucleotides. *Nature* 420:841–844
- Ronneberg TA, Landweber LF, Freeland SJ (2000) Testing a biosynthetic theory of the genetic code: Fact or artifact?. *Proc Natl Acad Sci USA* 97:13690–13695
- Schwemmler W (1994) Reconstruction of cell evolution: A periodic system of cells. CRC Press, Boca Raton, FL
- Sonneborn TM (1965) Degeneracy of the genetic code: extent, nature, and genetic implications. In: Bryson V, Vogel HJ (eds) *Evolving genes and proteins*. Academic Press, New York, pp 297–377
- Szathmary E (1992) What is the optimum size for the genetic alphabet? *Proc Natl Acad Sci USA* 89:2614–2618
- Szathmary E (1993) Coding coenzyme handles: A hypothesis for the origin of the genetic code. *Proc Natl Acad Sci USA* 90:9916–9920
- Szathmary E (1999) The origin of the genetic code. *Trends Genet* 15:223–229
- Szathmary E (2003) Why are there four letters in the genetic alphabet? *Nat Rev Genet* 4:995–1001
- Taylor FJR, Coates D (1989) The code within the codons. *BioSystems* 22:177–187
- Thanbichler M, Böck A (2002) The function of SECIS RNA in translational control of gene expression in *Escherichia coli*. *EMBO J* 21:6925–6934
- Topal MD, Fresco JR (1976) Base pairing and fidelity in codon-anticodon interaction. *Nature* 263:289–293
- Wächtershäuser G (1988) An all-purine precursor of nucleic acids. *Proc Natl Acad Sci USA* 85:1134–1135
- Weber AL, Lacey JC Jr (1978) Genetic code correlations: amino acids and their anticodon nucleotides. *J Mol Evol* 17:273–284
- Weiner AM, Maizels N (1987) tRNA-like structures tag the 3' ends of genomic RNA molecules for replication: Implications for the origin of protein synthesis. *Proc Natl Acad Sci USA* 84:7383–7390
- Woese CR (1965) On the evolution of the genetic code. *Proc Natl Acad Sci USA* 54:1546–1552
- Woese CR (1967) The genetic code: The molecular basis for genetic expression. Harper and Row, New York
- Woese CR, Dugre DH, Saxinger WC, Dugre SA (1966) The molecular basis for the genetic code. *Proc Natl Acad Sci USA* 55:966–974
- Woese CR, Dugre DH, Dugre SA, Kondo M, Saxinger WC (1967) On the fundamental nature and evolution of the genetic code. *Cold Spring Harbor Symp Quant Biol* 31:723–736
- Wolfenden RV, Cullis PM, Southgate CCF (1979) Water, protein folding, and the genetic code. *Science* 206:575–577
- Wong JT-F (1975) A co-evolution theory of the genetic code. *Proc Natl Acad Sci USA* 72:1909–1912
- Yarus M (1998) Amino acids as RNA ligands: A direct-RNA-template theory for the code's origin. *J Mol Evol* 47:109–117
- Yarus M (2000) RNA-ligand chemistry: A testable source for the genetic code. *RNA* 6:475–484
- Zuckermandl E, Pauling L (1965) Evolutionary divergence and convergence in proteins. In: Bryson V, Vogel HS (eds) *Evolving genes and proteins*. Academic Press, New York, pp 97–167
- Zimmerman JM, Eliezer N, Simna R (1968) *J Theor Biol* 21:170–201

A Purine-Pyrimidine Classification Scheme of the Genetic Code



Thomas Wilhelm, Svetlana Nikolajewa

Although containing the same information, our new classification scheme of the genetic code is simpler than the common representation as a three-dimensional matrix: it contains just 32 instead of 64 fields. Moreover, it shows known patterns in the code more clearly than the common scheme. Above all, with the help of our new scheme we could identify new patterns never seen before. This gives rise to some speculations about the origin and early evolution of the genetic code. We hypothesize that coding started in a binary doublet manner and developed via a quaternary doublet code to our contemporary quaternary triplet code. Most interestingly, it may be possible to discover traces of the old binary coding in present-day genomes.

The genetic code specifies how the information contained in the nucleic acids DNA and RNA is translated into the correct sequence of amino acids building the highly specific proteins. Up to the three termination codons UGA, UA(G/A) (standard code), each nucleotide triplet stands for exactly one amino acid, the methionin codon AUG is also the start codon. The genetic code is comma-free and non-overlapping. It is usually represented as a three-dimensional matrix in which the four rows stand for the first base and the four columns for the second base. To show the third dimension (the

third base) in the plane figure, each of the 16 boxes is again divided into four fields, giving together 64 entries (Fig.1).

For a long time one assumed that the genetic code is universal for all life forms on earth. Today there are at least 16 slightly deviating different codes known (www.ncbi.nlm.nih.gov/Taxonomy/Utils/wprintgc.cgi). However, it is generally believed that all these deviations are later descendants of the earlier standard code. Not surprisingly, non-standard codes are only found in small genomes, nearly all of them in mitochondria known to have by far the smallest genomes.

Since the early days of the discovery of the genetic code non-random patterns have been searched in the code for providing information about its origin and early evolution. In 1965 Nirenberg finished his famous project of deciphering the code. At that time most scientists believed that the code is the result of pure chance and hence does not need any further evolutionary explanation. Crick [1] formulated the corresponding "frozen accident" hypothesis which was widely accepted for many years. However, today it is assumed that at least some hints of possible evolutionary scenarios can be found in our contemporary code. The top-down approach, which we are following here, analyzes patterns in the code and tries to infer appropriate chemical and selective forces. The bottom-up approach, on the other hand, is rooted in biochemistry and aims at constructing plausible scenarios for the origin of coding.

It has been appreciated for a long time that the genetic code assigns similar amino acids to similar codons. Two different rationales have been presented: first, mutation and translation error minimization [2], and second, similar amino acids tend to directly interact with similar RNA sequences [3]. It was stated that "the canonical code is at or very close to a global optimum for error minimization" [4]. It has also been proposed that instead of the actual codons, some of their derivatives, such as the anticodons or codon-anticodon duplexes were the original amino acid binding motifs. It is also possible that the original amino acid recognition took place at the tRNA acceptor stem. Szathmari [5] proposed that amino acid-RNA allocation took place even

		2nd base					
		U	C	A	G		
1st base	U	UUU Phe UUC Phe UUA Leu UUG Leu	UCU Ser UCC Ser UCA Ser UCG Ser	UAU Tyr UAC Tyr UAA Stop UAG Stop	UGU Cys UGC Cys UGA Stop UGG Trp	3rd base	U C A G
	C	CUU Leu CUC Leu CUA Leu CUG Leu	CCU Pro CCC Pro CCA Pro CCG Pro	CAU His CAC His CAA Gln CAG Gln	CGU Arg CGC Arg CGA Arg CGG Arg		U C A G
	A	AUU Ile AUC Ile AUA Ile AUG Ile	ACU Thr ACC Thr ACA Thr ACG Thr	AAU Asn AAC Asn AAA Lys AAG Lys	AGU Ser AGC Ser AGA Arg AGG Arg		U C A G
	G	GUU Val GUC Val GUA Val GUG Val	GCU Ala GCC Ala GCA Ala GCG Ala	GAU Asp GAC Asp GAA Glu GAG Glu	GGU Gly GGC Gly GGA Gly GGG Gly		U C A G

Fig. 1: The common representation of the standard genetic code (mRNA triplets in the mRNA reading direction (5'→3')). Shaded regions show codon families.

before the appearance of tRNA. He also gave a possible evolutionary scenario for the development of an anticodon hairpin to a longer structure with an operational code at the acceptor stem.

However, the first codon position seems to be correlated with amino acid biosynthetic pathways and to their evolution as evaluated by synthetic “primordial soup” experiments. The second position is correlated with the hydrophatic properties of the amino acids, and the degeneracy of the third position could be related to the molecular weight or size of the amino acids [6]. Lagerkvist [7] observed that codon families (the amino acid of a codon family is uniquely determined by the first two nucleotides of a codon) have a much higher probability to appear in the left part of the common illustration scheme (cf. Fig. 1). He also found that “strong” codons (the first two nucleotides in the codon are G and/or C) always represent codon families, while “weak” codons (A and/or U as the first two nucleotides) never do so. “Mixed” codons in the right part of the scheme never represent codon families, whereas mixed codons in the left part always stand for a codon family.

The New Classification Scheme of the Genetic Code

Most amino acid properties show no clear pattern in the common scheme of

the genetic code. Recently we proposed a new classification scheme [8 and www.imb-jena.de/~sweta/genetic_code], based on a binary purine(1)-pyrimidine(0) coding (Fig. 2). It shows known regularities more clearly than the common scheme and it even highlights some new patterns.

Code	Strong codons 6 hydrogen bonds	Mixed codons 5 hydrogen bonds	Mixed codons 5 hydrogen bonds	Weak codons 4 hydrogen bonds
000	Pro CC (C/U)	Ser UC (C/U)	Leu CU (C/U) 1/1	Phe UU (C/U)
001	Pro CC (A/G)	Ser UC (A/G) 0/1	Leu CU (A/G) 2/1	Leu UU (A/G) 0/1
100	Ala GC (C/U)	Thr AC (C/U)	Val GU (C/U)	Ile AU (C/U)
101	Ala GC (A/G)	Thr AC (A/G)	Val GU (A/G)	Ile / Met AU (A/G) 0/5
010	Arg CG (C/U)	Cys UG (C/U)	His CA (C/U)	Tyr UA (C/U)
011	Arg CG (A/G)	Stop / Trp UG (A/G) 0/9	Gln CA (A/G)	Stop UA (A/G) 4/2
110	Gly GG (C/U)	Ser AG (C/U)	Asp GA (C/U)	Asn AA (C/U)
111	Gly GG (A/G)	Arg AG (A/G) 6/6	Glu GA (A/G)	Lys AA (A/G) 0/3

Fig. 2: The purine(1)-pyrimidine(0) classification scheme of the standard genetic code. The third base is given in parenthesis. If there are differences between the standard code and any other code, the number of deviations from the standard code is indicated. For instance, in the UG(G/A) field, 0/9 indicates that UGG encodes for Trp in all codes, but UGA is not the termination codon in 9 of the 16 non-standard codes. In some bacteria the 21st amino acid, selenocysteine, can also be encoded by UGA. Shaded regions show codon families. The point in the center indicates the perfect point symmetry corresponding to Halitsky's family – nonfamily symmetry operation [9]. The thick horizontal line marks the symmetry axis for codon-anticodon symmetry.

There are three possible variants of a binary coding scheme for the genetic code: One could group the bases (i) according to base-pairs (A,U = 1, G,C = 0), (ii) according to keto- and aminobases (G,U = 1, A,C = 0), and (iii) according to purines and pyrimidines (A,G = 1, C,U = 0). In such a simplified code eight different binary triplets exist: 000, 001, ..., 111. Each of these binary triplets represents eight different codons, e.g. in our coding scheme 000 stands for CCC, CCU, ..., UUU. The purine-pyrimidine coding is superior to the other two variants, because it is the only one that allows the genetic code to be represented using just four columns (Fig. 2). The reason for this vast simplification in our scheme is that for the third position in each triplet it only matters if it is a purine or a pyrimidine.

Given the primary purine-pyrimidine coding, we have again two different possibilities to sort the first two bases per row: one can use either of the remaining two binary codings, according to base-pairs or according to keto- and aminobases as a sort criterion inside the rows. We have chosen the base-pairs for sorting inside rows, because only this reveals the following regularities of the genetic code: (i) All codon families group together, i.e. they are not scattered all over the table. (ii) More importantly, the codon strength classification directly corresponds to the columns in our scheme (cf. Fig. 2). Thus, in the first column the first two bases complementary pair with 6 hydrogen bonds, in the sec-

ond and third column with 5, and in the fourth column with just 4 hydrogen bonds. For all these reasons our classification scheme of the genetic code is superior to all similar ones.

Our new scheme shows some fascinating regularities. We can, for instance, better understand the number of different tRNAs in some organisms. In the simplest case one should expect one tRNA per coding field in our scheme. Exactly this happens in the case of vertebrate mitochondria. It is known that animal mitochondria contain exactly 22 different tRNAs. In vertebrate mitochondria UA1 and AG1 are stop codons. Thus there are exactly 22 fields for amino acids left: the 8 codon families plus 14 remaining fields. Interestingly, the 22 tRNAs in animal mitochondria correspond 1:1 to these 22 fields.

The amino acids of the nine "strong groups" (mutually evolutionary conserved, based on the alignment score matrix PAM250, cf. http://bioinfolab.unl.edu/em-lab/documents/clustalx_doc/clustalw.txt) very closely group together in our scheme, more closely than in the standard scheme. That means neighboring amino acids in our scheme have a higher probability to be aligned to each other in genome comparisons than neighboring amino acids in the standard scheme.

Our new scheme also led us to detect hitherto unknown regularities of amino acid properties in the genetic code. Jungck [10] collected 15 different measures of amino acid properties. For all of these we arranged a table with 8 rows and 4 columns corresponding to our scheme. Amazingly, the column sums of nearly all measures are perfectly correlated to the corresponding codon-anticodon binding strength. For instance, the first column harbours more polar amino acids, the last column less polar ones and the mixed codon fields are in between. Similarly, the bulkiness and the specific volume increases continuously from the first to the last column.

Evolution of the Genetic Code

The observed regularities inspire to some speculations about the early evolution of the genetic code. Thus the strong correlation between amino acid properties and codon strength implies that the first two positions together (and not the second position alone as speculated by others) must have been important for the amino acid – codon assignment in the early evolution of the code. It therefore also could be that just the first two nucleotides of a codon (or anticodon) show specific binding affinity to the corre-

sponding amino acid (maybe important in the process of the code formation).

Nowadays one assumes that "the code probably underwent a process of expansion from relatively few amino acids to the modern complement of 20" [11]. Can we find some hints in our scheme indicating coding of less than 20 amino acids in ancient times? Indeed, there is a high redundancy for each second row. This gives rise to the speculation that in the early days of code evolution just the first two bases of the triplet were coding. The reading frame, however, arguably always comprised three letters. In any way, a quaternary doublet can encode at most 16 amino acids, or 15 plus one termination codon (some bacteria exist that do not possess any stop codon). In this context it is interesting to note that Asn, Gln, Met, Trp, and Tyr seem to be newer amino acids.

Since the discovery of the genetic code it is speculated that the first genetic material contained only a single base-pairing unit [1]. Recently, for the first time a ribozyme was found composed of only one purine and one pyrimidine [12]. Assuming a binary doublet code, it is tempting to speculate which four amino acids, one per two consecutive rows, were the first encoded ones. In the first two rows Ser seems to be the oldest amino acid, and in the third and fourth row Ala. The 01-rows obviously contain no really old amino acid while the 11-rows contain more than one: Gly, Asp, Glu. However, Gly is biochemically built from Ser, so Ser can be assumed as prior. It could be that in the beginning of nucleic acid – amino acid assignment Asp and Glu competed for the 11-doublet. Of course, code transfer from one amino acid to another might also have occurred.

Conclusions

We have found a concise scheme for the genetic code that is superior to similar schemes for different reasons. It shows nice patterns and symmetries and even so far unknown regularities in the code. We are now studying the fascinating question whether we still can find traces of doublet coding or even binary coding in contemporary genomes.

References are available from the authors.

Dr. Thomas Wilhelm
Svetlana Nikolajewa
Theoretical Systems Biology
Institute of Molecular Biotechnology
Beutenbergstr. 11
07745 Jena, Germany
wilhelm@imb-jena.de
sweta@imb-jena.de

December 9, 2005 12:54 WSPC/INSTRUCTION FILE jbcbl

Journal of Bioinformatics and Computational Biology
© Imperial College Press

**THE NEW CLASSIFICATION SCHEME OF THE GENETIC CODE,
ITS EARLY EVOLUTION, AND tRNA USAGE***

SWETLANA NIKOLAJEWA
MAIK FRIEDEL
ANDREAS BEYER
THOMAS WILHELM

*Theoretical Systems Biology, Institute of Molecular Biotechnology Beutenbergstr.11, Jena,
D-07745, Germany
wilhelm@imb-jena.de*

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

We present a new classification scheme of the genetic code. In contrast to the standard form it clearly shows five codon symmetries: codon-anticodon, codon-reverse codon, and sense-antisense symmetry, as well as symmetries with respect to purine-pyrimidine (A vs. G, U vs. C) and keto-aminobase (G vs. U, A vs. C) exchanges. We study the number of tRNA genes of 16 archaea, 81 bacteria and 7 eucaryotes to analyze whether these symmetries are reflected in corresponding tRNA usage patterns. Two features are especially striking: reverse stop codons do not have their own tRNAs (just one exception in human), and **A**** anticodons are significantly suppressed. Our classification scheme of the genetic code and the identified tRNA usage patterns support recent speculations about the early evolution of the genetic code. In particular, pre-tRNAs might have had the ability to bind their codons in two directions to the corresponding codons.

Keywords: Genetic code; evolution; tRNA

1. Introduction

The genetic code specifies how the information contained in the nucleic acids is translated into the correct sequence of amino acids. It is usually represented as shown in Figure 1. Since the early days of the discovery of the genetic code patterns have been searched for gaining insights into its origin and early evolution⁸. It is known that the genetic code assigns similar amino acids to similar codons. Two different rationales have been presented: first, mutation and translation error minimization^{3,10}, and second, similar amino acids tend to directly interact with similar RNA sequences⁴². It has also been stated that instead of the actual codons,

*This work has been supported by the Bundesministerium für Bildung und Forschung Grant 0312704E.

December 9, 2005 12:54 WSPC/INSTRUCTION FILE jbec1

2 Svetlana Nikolajewa, Maik Friedel, Andreas Beyer and Thomas Wilhelm

First base	Second base								Third base
	U		C		A		G		
U	UUU	Phe	UCU	Ser	UAU	Tyr	UGU	Cys	U
	UUC	Phe	UCC	Ser	UAC	Tyr	UGC	Cys	C
	UUA	Leu	UCA	Ser	UAA	Stop	UGA	Stop	A
	UUG	Leu	UCG	Ser	UAG	Stop	UGG	Trp	A
C	CUU	Leu	CCU	Pro	CAU	His	CGU	Arg	U
	CUC	Leu	CCC	Pro	CAC	His	CGC	Arg	C
	CUA	Leu	CCA	Pro	CAA	Gln	CGA	Arg	A
	CUG	Leu	CCG	Pro	CAG	Gln	CGG	Arg	A
A	AUU	Ile	ACU	Thr	AAU	Asn	AGU	Ser	U
	AUC	Ile	ACC	Thr	AAC	Asn	AGC	Ser	C
	AUA	Ile	ACA	Thr	AAA	Lys	AGA	Arg	A
	AUG	Met	ACG	Thr	AAG	Lys	AGG	Arg	A
G	GUU	Val	GCU	Ala	GAU	Asp	GGU	Gly	U
	GUC	Val	GCC	Ala	GAC	Asp	GGC	Gly	C
	GUA	Val	GCA	Ala	GAA	Glu	GGA	Gly	A
	GUG	Val	GCG	Ala	GAG	Glu	GGG	Gly	A

Fig. 1. The common representation of the standard genetic code (codon families are shaded).

some of their derivatives, such as the anticodons^{9,14} or codon-anticodon duplexes² were the original amino acid binding motifs. Recently, a new mechanism has been proposed for the association of amino acids with their codons and the origin of the genetic code⁶. It could explain two other long-known regularities of the genetic code. The first codon position seems to be correlated with amino acid biosynthetic pathways and to their evolution as evaluated by synthetic “primordial soup” experiments^{32,39}. The second position is correlated with the hydrophobic properties of the amino acids. Codons with **U** as the second base code for the most hydrophobic amino acids and those having **A** as the second base are associated with the most hydrophilic amino acids³². Lagerkvist^{16,17} observed that codon families (the amino acid of a codon family is determined by the first two nucleotides of a codon alone) have a much higher probability to appear in the left part of the common illustration (Fig. 1).

Recently, it was found that special purine - pyrimidine patterns of DNA binding sites facilitate recognition by restriction enzymes²¹. Here we show that also the genetic code is largely determined by purine - pyrimidine coding.

The following section introduces a new classification scheme of the genetic code based on the purine - pyrimidine coding, which demonstrates different codon symmetries that do not appear in the standard scheme. Section 3 presents an analysis of tRNA frequencies in 104 species (tRNA usage patterns), corresponding to the five symmetries in the new scheme. For each codon the number of genes coding

December 9, 2005 12:54 WSPC/INSTRUCTION FILE jbcbl

The New Classification Scheme of the Genetic Code, its Early Evolution, and tRNA Usage 3

	Strong <i>6 hydrogen bonds</i>	Mixed <i>5 hydrogen bonds</i>	Mixed <i>5 hydrogen bonds</i>	Weak <i>4 hydrogen bonds</i>
000	<i>Pro</i> CC (C/U) Proline	<i>Ser</i> UC (C/U) Serine	<i>Leu</i> CU (C/U) Leucine	<i>Phe</i> UU (C/U) Phenylalanine
001	<i>Pro</i> CC (A/G) Proline	<i>Ser</i> UC (A/G) Serine	<i>Leu</i> CU (A/G) Leucine	<i>Leu</i> UU (A/G) Leucine
100	<i>Ala</i> GC (C/U) Alanine	<i>Thr</i> AC (C/U) Threonine	<i>Val</i> GU (C/U) Valine	<i>Ile</i> AU (C/U) Isoleucine
101	<i>Ala</i> GC (A/G) Alanine	<i>Thr</i> AC (A/G) Threonine	<i>Val</i> GU (A/G) Valine	<i>Ile/Met</i> AU (A/G) Isoleucine/Methionine
010	<i>Arg</i> CG (C/U) Arginine	<i>Cys</i> UG (C/U) Cysteine	<i>His</i> CA (C/U) Histidine	<i>Tyr</i> UA (C/U) Tyrosine
011	<i>Arg</i> CG (A/G) Arginine	<i>Stop/Trp</i> UG (A/G) Tryptophan	<i>Gln</i> CA (A/G) Glutamine	<i>Stop</i> UA (A/G)
110	<i>Gly</i> GG (C/U) Glycine	<i>Ser</i> AG (C/U) Serine	<i>Asp</i> GA (C/U) Aspartic acid	<i>Asn</i> AA (C/U) Asparagine
111	<i>Gly</i> GG (A/G) Glycine	<i>Arg</i> AG (A/G) Arginine	<i>Glu</i> GA (A/G) Glutamic acid	<i>Lys</i> AA (A/G) Lysine

Fig. 2. The purine(1)-pyrimidine(0) classification scheme of the genetic code. The third base is given in parenthesis. Shaded regions show codon families. The dashed horizontal line marks the symmetry axis for codon-anticodon symmetry and the dashed vertical line the mirror symmetry of purine ($G \leftrightarrow A$)-pyrimidine ($C \leftrightarrow U$) exchange. The point in the center indicates the point symmetry corresponding to keto-aminobase exchanges ($G \leftrightarrow U$, $A \leftrightarrow C$).

for the tRNA with the complimentary anticodon (according to the Watson-Crick base pairing) is counted. Note that this analysis differs from the codon adaptation index^{12,28}, which additionally takes cytoplasmic tRNA concentration into account. The most striking features of tRNA usage allow us to extend our earlier speculations concerning the evolution of the genetic code³⁷.

2. The New Classification Scheme of the Genetic Code

In contrast to the common representation of the genetic code our scheme is based on a binary encoding of the four bases **A**, **G**, **U**, **C**. There are three possibilities of a binary base coding⁴¹, according to:

- (i) weak-strong bases (**A,U** = 1; **G,C** = 0),
- (ii) keto- and aminobases (**G,U** = 1; **A,C** = 0), and
- (iii) purines and pyrimidines (**A,G** = 1; **C,U** = 0).

In such a simplified code eight different binary triplets exist: 000, 001, ..., 111. Each of these binary triplets represents eight different codons, e.g. in our purine-pyrimidine coding scheme 000 stands for **CCC**, **CCU**, ..., **UUU**. The purine-pyrimidine coding is superior to the other two variants, because it is the only one

December 9, 2005 12:54 WSPC/INSTRUCTION FILE jbec1

4 Svetlana Nikolajewa, Maik Friedel, Andreas Beyer and Thomas Wilhelm

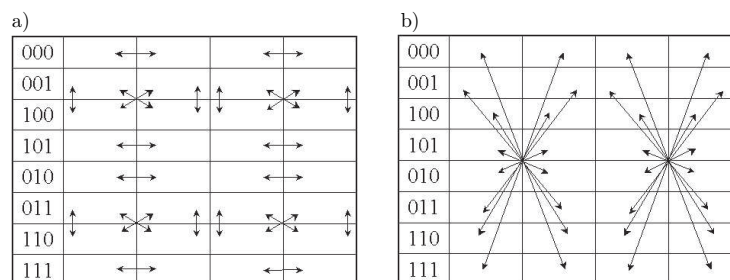


Fig. 3. The codon - reverse codon pattern (a) and the sense - antisense codon pattern (b) in the purine - pyrimidine scheme of the genetic code. For instance, codon **GAU** is reverse to the codon **UAG** and codon **CGU** is the antisense codon of **ACG**. The codon symmetries are indicated by arrows.

that allows the genetic code to be represented using just four columns (Fig. 2). The reason for this vast simplification in our scheme is that for the third codon position it only matters if it is a purine or a pyrimidine. Interestingly, the only two exceptions are the start (**AUG**) and stop (**UGA**) codons. Given the purine-pyrimidine coding, there are two possibilities to sort the first two bases per row: one can use either of the remaining two binary codings, according to the weak and strong bases or according to keto- and aminobases, as a sort criterion inside the rows. We have chosen the weak-strong splitting to sort rows, because only this reveals the following regularities of the genetic code. First, all codon families group together, i.e. they are not scattered all-over the table. Secondly, the codon strength classification directly corresponds to the columns in our scheme (Fig. 2). Thus, in the first column the first two bases complementary pair with 6 hydrogen bonds (strong codons), in the second and third column with 5 (mixed codons), and in the fourth column with just 4 hydrogen bonds (weak codons).

In addition to its simplicity the new scheme uniquely shows 5 codon symmetries (Fig. 2 and 3) that are not obvious in other representations of the genetic code. Figure 2 reveals that the recently proposed family-nonfamily symmetry operation¹³, exchanging the amino bases (**A** ↔ **C**) and the keto bases (**G** ↔ **T**), corresponds to the point symmetry in our scheme. Moreover, the horizontal mirror symmetry corresponds to the codon-anticodon symmetry (weak (**A** ↔ **U**) and strong base (**G** ↔ **C**) exchanges) and the vertical mirror symmetry represents the purine-pyrimidine exchange symmetry (**G** ↔ **A**, **C** ↔ **U**). Figure 3a shows the symmetric codon-reverse codon pattern and Figure 3b the sense-antisense codon pattern. Note that the last four patterns cannot be seen in the usual presentation of the genetic code (Fig. 1).

In our recently presented new classification scheme of the genetic code³⁷ there was one ambiguity left concerning the amino acid arrangement: the order of the

December 9, 2005 12:54 WSPC/INSTRUCTION FILE jbecb1

The New Classification Scheme of the Genetic Code, its Early Evolution, and tRNA Usage 5

second and third column was arbitrary. We now present four reasons for choosing the column order (mixed codons) as shown in Fig.2:

- (i) The codon-reverse codon symmetry, and
- (ii) the sense-antisense symmetry are revealed only by the chosen order.
- (iii) In each quadrant of the scheme the second position of the corresponding triplets is the same.
- (iv) Strongly conserved groups of amino acids³⁴ are subsets of exactly one quadrant, e.g. the amino acids *M, I, L, V* belong to the upper right block in the table. The other conserved strong groups belonging to one block are *MILF, STA, NEQK, NHQK, NDEQ, HY*. The only exceptions are *QHRK* (*R (Arg)* is in another quadrant) and *FYW* (in three quadrants). In other words, reverse codon pairs tend to code for evolutionary similar amino acids, and each quadrant is enriched for amino acids with similar biochemical properties.

The new scheme of the genetic code has now its optimal form (Fig. 2). It shows five different triplet symmetries, including two additional symmetries that could not be seen in our first version of the scheme³⁷.

3. Patterns of tRNA usage

We studied tRNA usage of 16 archaea, 81 bacteria and 7 eucaryotes, using all information from the public database Genomic tRNA Compilation²⁹. Different tables corresponding to the identified codon symmetries were composed, each containing all codons together with their symmetric codons. Table 1 shows the tRNA usage of all organisms, corresponding to the codon-reverse codon symmetry. Rows are sorted by the number of tRNA genes for a given anticodon (highest priority archaea, second priority bacteria).

The order in Table 1 shows best the following main observations. The first interesting pattern of tRNA usage refers to reverse STOP codons.^a Of course, no species has a tRNA with an anticodon complementary to any termination codon. Intriguingly, there is also no tRNA with an anticodon for a reverse STOP codon. The only exception is *H. sapiens* with one tRNA^{A^{sn}} with the anticodon **A^{sn}TT**. The lack of specific tRNAs does not imply that no tRNA exists which can recognize reverse STOP codons. For instance, using base pairing allowed by Crick's wobble rules⁷, tRNA with the **GTT** anticodon can recognize the reverse STOP codon **AAT**.

The second striking pattern in tRNA usage is the significant suppression of tRNAs with **A** at the first anticodon position (Tab. 1). **A**** anticodons are fully excluded in archaea. In bacteria and eucaryotes there are some exceptions, but it can be observed that **AY*** anticodons do not appear in any species.

^atRNA genes specifically recognizing initiation codon (*Met*) are significantly overrepresented.

December 9, 2005 12:54 WSPC/INSTRUCTION FILE jbecb1

6 Svetlana Nikolajewa, Maik Friedel, Andreas Beyer and Thomas Wilhelm

Table 1. Codon-reverse codon pairs and the corresponding number of tRNA genes.

Amino acid pairs	Codon pairs	Anticodon pairs	Number of tRNA genes		
			archaea(16)	bacteria(81)	eucaryotes(7)
Cys	TGT	ACA	0	0	0
Phe	TTT	AAA	0	0	0
Tyr	TAT	ATA	0	0	1
Ser	TCT	AGA	0	0	28
Ile	ATA	TAT	0	5	16
Asp ↔ Stop	GAT ↔ TAG	ATC ↔ CTA	0 ↔ 0	0 ↔ 0	0 ↔ 0
Ser ↔ Stop	AGT ↔ TGA	ACT ↔ TCA	0 ↔ 0	0 ↔ 0	0 ↔ 0
Asn ↔ Stop	AAT ↔ TAA	ATT ↔ TTA	0 ↔ 0	0 ↔ 0	1 ↔ 0
Val ↔ Leu	GTT ↔ TTG	AAC ↔ CAA	0 ↔ 12	0 ↔ 93	18 ↔ 29
Ala ↔ Ser	GCT ↔ TCG	AGC ↔ CGA	0 ↔ 12	1 ↔ 64	36 ↔ 13
Gly ↔ Trp	GGT ↔ TGG	ACC ↔ CCA	0 ↔ 14	0 ↔ 111	0 ↔ 19
Pro ↔ Ser	CCT ↔ TCC	AGG ↔ GGA	0 ↔ 16	0 ↔ 99	16 ↔ 1
Ile ↔ Leu	ATT ↔ TTA	AAT ↔ TAA	0 ↔ 16	0 ↔ 107	16 ↔ 21
His ↔ Tyr	CAT ↔ TAC	ATG ↔ GTA	0 ↔ 16	0 ↔ 118	0 ↔ 55
Thr ↔ Ser	ACT ↔ TCA	AGT ↔ TGA	0 ↔ 16	1 ↔ 114	18 ↔ 21
Leu ↔ Phe	CTT ↔ TTC	AAG ↔ GAA	0 ↔ 16	8 ↔ 113	20 ↔ 25
Arg ↔ Cys	CGT ↔ TGC	ACG ↔ GCA	0 ↔ 16	114 ↔ 104	18 ↔ 46
Ala	GCG	CGC	12	28	15
Glu	GAG	CTC	12	30	19
Gly	GGG	CCC	12	46	13
Arg	CGC	GCG	14	12	0
Val	GTG	CAC	14	31	18
Glu ↔ Lys	GAA ↔ AAG	TTC ↔ CTT	14 ↔ 12	122 ↔ 59	21 ↔ 29
Pro	CCC	GGG	15	66	0
Thr	ACA	TGT	15	115	19
Val ↔ Leu	GTC ↔ CTG	GAC ↔ CAG	15 ↔ 12	92 ↔ 76	0 ↔ 10
Leu	CTC	GAG	16	90	2
His	CAC	GTG	16	106	16
Lys	AAA	TTT	16	120	25
Arg	AGA	TCT	16	121	21
Ala ↔ Pro	GCC ↔ CCG	GGC ↔ CCG	16 ↔ 12	73 ↔ 49	0 ↔ 11
Gly ↔ Arg	GGA ↔ AGG	TCC ↔ CCT	16 ↔ 12	113 ↔ 81	18 ↔ 14
Ala ↔ Thr	GCA ↔ ACG	TGC ↔ CGT	16 ↔ 12	118 ↔ 66	17 ↔ 18
Asp ↔ Gln	GAC ↔ CAG	GTC ↔ CTG	16 ↔ 13	121 ↔ 43	21 ↔ 26
Ser ↔ Arg	AGC ↔ CGA	GCT ↔ TCG	16 ↔ 14	105 ↔ 30	21 ↔ 20
Pro ↔ Thr	CCA ↔ ACC	TGG ↔ GGT	16 ↔ 15	109 ↔ 107	29 ↔ 0
Asn ↔ Gln	AAC ↔ CAA	GTT ↔ TTG	16 ↔ 17	132 ↔ 115	41 ↔ 23
Gly ↔ Arg	GGC ↔ CCG	GCC ↔ CCG	17 ↔ 11	116 ↔ 74	23 ↔ 8
Ile ↔ Leu	ATC ↔ CTA	GAT ↔ TAG	17 ↔ 17	106 ↔ 107	1 ↔ 12
Met ↔ Val	ATG ↔ GTA	CAT ↔ TAC	45 ↔ 16	285 ↔ 118	33 ↔ 12

Another observation concerning tRNA usage is the significant suppression of **A*A** self-reverse codons. In no archaea and in no bacteria any tRNA has such an anticodon. In archaea the anticodon **TAT** is the only one without own tRNAs that is not a STOP anticodon or an **A**** anticodon. Interestingly, this is the only anticodon which according to Crick's wobble rules⁷ allows recognition of a STOP codon (**TAG**).

December 9, 2005 12:54 WSPC/INSTRUCTION FILE jbec1

The New Classification Scheme of the Genetic Code, its Early Evolution, and tRNA Usage 7

4. The reverse recognition conjecture

In this section we present a hypothesis that consistently explains the observed suppression of anticodons for reverse STOP codons. We conjecture that the observed tRNA usage patterns reflect important features of the ancient translation machinery. Maybe, in the early days of translation, pre-tRNAs were able to recognize codons in both directions (Fig. 4). In order to guarantee termination (i.e., to avoid incorrect elongation) the reverse stop codons had (and have) no own tRNA.

In agreement with others^{6,24,40}, we hypothesized in our previous work that the contemporary triplet code developed from an ancient doublet code³⁷. However, in order to avoid the frameshift problem one has to assume a triplet reading frame also in doublet coding times³⁷. In agreement, the triplet reading frame was recently substantiated because unpaired RNA loops with 7 and 8 nucleotides are the most stable ones⁴⁰. Nevertheless, one still wonders about such an information wasting, where the third base would not carry any information at all. We speculate that in the early days of translation pre-tRNAs could fit in two opposite directions to the corresponding mRNA (Fig. 4). This would resolve the wasting problem: if a codon could be recognized in both directions all bases would carry information, although in a given codon-anticodon pairing only two bases are analyzed. Three different facts support our speculation. First, ancient pre-tRNAs presumably only consisted of the anticodon loop, lacking the D- and T-loops²⁷. Such pre-tRNAs would have been (almost) symmetrical and could thus bind in two directions. If the reverse recognition model is correct, the resulting polypeptide should be relatively independent of the pre-tRNA binding direction. This is supported by the special role of the central triplet base^{38,40}. It is well-known that the second base has the strongest interaction with the bases of 16S RNA (the universally conserved and essential bases A1492, A1493, and G530^{22,30}). Moreover, the middle base of the anticodon has particularly strong interactions with the correct aminoacyl-tRNA synthetase during amino acid attachment²⁰. The second base is exceptional also in another respect: it is correlated to the main physical property of amino acids, the hydrophobicity³². The third fact supporting our “reverse recognition conjecture” is the above observation that reverse codon pairs generally encode evolutionary similar amino acids³⁴. We suppose that this observation is a relict from old “reverse recognition times”, where the reverse recognition should have a minimal effect on the resulting polypeptide.

It was speculated that the translation machinery of the last universal common ancestor (LUCA) is most similar to that of archaea³⁵, so we expect that tRNA usage patterns in archaea reflect ancient translation. What could be the reason for the forbidden **A**** anticodon-tRNAs? Three different explanations can be given. First, sometimes **A** (and the simple derivative inosine **I**) at the third codon position misleadingly pair with the first anticodon position²³. In order to prevent such a mistranslation **A**** anticodon-tRNAs could have been forbidden. A second possible explanation is the strong preference of **G** (instead of **A**) at the first anticodon po-

December 9, 2005 12:54 WSPC/INSTRUCTION FILE jbcbl

8 Svetlana Nikolajewa, Maik Friedel, Andreas Beyer and Thomas Wilhelm

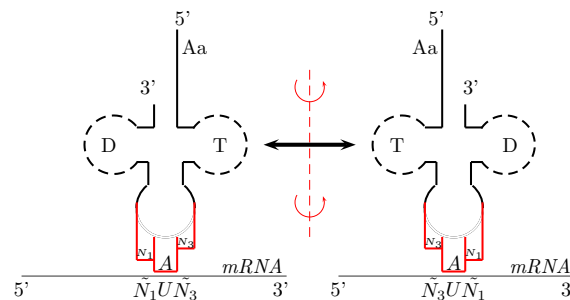


Fig. 4. Possible ancient codon-anticodon recognition with doublet coding and reverse pairing. The figure shows schematically the binding of the same pre-tRNA in normal and reverse direction to two different (reverse) codons. The proximity of the anticodon 'fingers' to the codon represents the accuracy by which the respective nucleotides are recognized¹¹.

sition in order to recognize the corresponding pyrimidines. In the recently proposed evolution of wobble rules³⁵ **G** (at the first anticodon position) always recognizes **U** and **C**, in all discussed evolution stages. **A**** anticodons, in contrast, could not recognize any base in the early stages³⁵. This would also be in agreement with earlier speculations about a binary coding scheme with just one purine and one pyrimidine^{26,37}. Interestingly, Table 1 reveals that nearly all 16 **G**** anticodons have corresponding tRNAs in all species. The third explanation is based on an observation concerning initiation codons. Translation in eukaryotes can be initiated from codons other than **AUG**. A well documented case (including direct protein sequencing) is the **GUG** start of a ribosomal P2A protein of the fungus *Candida albicans*¹. Other examples can be found in the NCBI taxonomy database^{4,36}. Interestingly, all 9 different initiation codons have **U** at the second position (**AUG** (standard), **AUA**, **AUU**, **AUC**, **GUG**, **GUA**, **UUG**, **UUA**, **CUG**). Maybe, in earlier times ***U*** codons generally could initiate translation, starting with ***A*** anticodons. We speculate that the forbidden **A**** anticodons should protect the transcript against wrong translation initiation which would lead to a frameshift.

Moreover, we note that the three termination codons in the standard genetic code all have a purine at the second position. The alternative termination codons in non-standard codes are also ***R*** codons (**AGA** and **AGG** in vertebrate mitochondria^{4,36}). Maybe, in "binary coding times"³⁷ ***Y*** codons could initiate translation, whereas ***R*** codons could terminate translation. This additionally supports our speculations of possible reverse codon recognition. Up to now the possibility of reverse recognition provides the only explanation that consistently integrates all of our observations. This model might be used as a plausible framework onto which research into translation evolution may be devised.

December 9, 2005 12:54 WSPC/INSTRUCTION FILE jbec1

The New Classification Scheme of the Genetic Code, its Early Evolution, and tRNA Usage 9

5. Discussion

We presented a new classification scheme of the genetic code. It has now its optimal form, no ambiguities in codon order are left. The scheme clearly shows all five different codon symmetries. We also studied the occurrence of tRNA genes in archaea, bacteria and eukaryotic species. tRNA usage, ordered according to the codon-reverse codon symmetry, shows three interesting facts:

- (i) some reverse codons are significantly underrepresented, most strikingly, there are no specific tRNAs for reverse STOP codons;
- (ii) **A**** anticodons are significantly repressed, **G**** anticodons are significantly utilized, and
- (iii) **A*A** self-reverse anticodons are totally excluded in archaea and bacteria.

This led us to extend our earlier speculations on doublet coding³⁷. We conjecture that in earlier times codon recognition could also have been carried out in the reverse direction with first recognizing the second base (Fig. 4). Our hypothesis is related to the recently proposed evolution scheme of the genetic code, where it was suggested that "... triplet codons gradually evolved from two types of ambiguous doublet codons, those in which the first two bases of each three-base window were read ('prefix' codons) and those in which the last two bases of each window were read ('suffix' codons).⁴⁰" In contrast to this model our reverse recognition conjecture implies a parallel-stranded duplex structure of the two relevant codon-anticodon base pairs. Although such parallel structures are difficult to find in natural nucleic acids they have been observed in DNA⁵ and mRNA³³, and a corresponding crystal structure has been reported³¹. However, because RNA is unstable and difficult to synthesize, it was proposed that the first genetic material used a simpler backbone than ribose¹⁵. For such molecules the pairing strand direction is probably not as constraint as in DNA or RNA.

Of course, all discussed tRNA usage patterns depend on the completeness of the known tRNAs. If a significant number of tRNAs is still unknown, this might modify these patterns. However, the fact that tRNAs are systematically searched by robust computer algorithms^{18,19,25} makes it very unlikely that such a significant number of tRNAs will be found in the future.

Acknowledgments

We thank R. Brockmann for critically reading the manuscript and an anonymous referee for valuable comments. This work was supported by Grant no. 0312704E of the Bundesministerium fuer Bildung und Forschung, Germany.

References

1. Abramczyk D, Tchorzewski M, and Grankowski N, Non-AUG translation initiation of mRNA encoding acidic ribosomal P2A protein in *Candida albicans*, *Yeast*, **20**:1045–1052, 2003.

December 9, 2005 12:54 WSPC/INSTRUCTION FILE jbcbl

10 Svetlana Nikolajewa, Maik Friedel, Andreas Beyer and Thomas Wilhelm

2. Alberti S, The origin of the genetic code and protein synthesis, *J Mol Evol*, **45**:352–358, 1997.
3. Ardell DH, Sella G, No accident: genetic codes freeze in error-correcting patterns of the standard genetic code, *Phil Trans R Soc Lond B*, **357**:1625–1642, 2002.
4. Benson DA, Karsch-Mizrachi I, Lipman DJ, Ostell J, Rapp BA, and Wheeler DL, GenBank, *Nucleic Acids Res*, **28**:15–18, 2000.
5. Borisova OF, Shchylolkina AK, Chernov BK, and Tchurikov NA, Relative stability of AT and GC pairs in parallel DNA duplex formed by a natural sequence, *FEBS Lett*, **322**:304–306, 1993.
6. Copley SD, Smith E, and Morowitz HJ, A mechanism for the association of amino acids with their codons and the origin of the genetic code, *Proc Natl Acad Sci U S A*, **102**:4442–4447, 2005.
7. Crick FHC, Codon-anticodon pairing: the wobble hypothesis, *J Mol Biol*, **19**:548–555, 1966.
8. Crick FHC, The origin of the genetic code, *J Mol Biol*, **38**:367–379, 1968.
9. Dunnill P, Triplet nucleotide-amino-acid pairing; a stereochemical basis for the division between protein and non-protein amino-acids, *Nature*, **210**:1265–1267, 1966.
10. Freeland SJ, Knight RD, Landweber LF, and Hurst LD, Early fixation of an optimal genetic code, *Mol Biol Evol*, **17**:511–518, 2000.
11. Freeland SJ and Hurst LD, The genetic code is one in a million, *J Mol Evol*, **47**:238–248, 1998.
12. Friberg M, von Rohr P, Gonnet G, Limitations of codon adaptation index and other coding DNA-based features for prediction of protein expression in *Saccharomyces cerevisiae*, *Yeast*, **21**:1083–1093, 2004.
13. Halitsky D, Extending the (hexa-)rhombic dodecahedral model of the genetic code: the code's 6-fold degeneracies and the orthogonal projections of the 5-cube as 3-cube. Contributed paper (983-92-151), *American Mathematical Society; and personal communication*, 2003.
14. Jungck JR, The genetic code as a periodic table, *J Mol Evol*, **11**:211–224, 1978.
15. Knight RD, Landweber LF, The early evolution of the genetic code, *Cell*, **101**:569–572, 2000.
16. Lagerkvist U, “Two out of three”: An alternative method for codon reading, *Proc Natl Acad Sci USA*, **75**:1759–1762, 1978.
17. Lagerkvist U, Unorthodox codon reading and the evolution of the genetic code, *Cell*, **23**:305–306, 1981.
18. Laslett D and Canback B, ARAGORN, a program to detect tRNA genes and tmRNA genes in nucleotide sequences, *Nucleic Acids Res*, **32**:11–16, 2004.
19. Lowe TM and Eddy SR, tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence, *Nucleic Acids Res*, **25**:955–964, 1997. <http://lowelab.ucsc.edu/GtRNAdb/>
20. McClain WH, Schneider J, Bhattacharya S, and Gabriel K, The importance of tRNA backbone-mediated interactions with synthetase for aminoacylation, *Proc Natl Acad Sci U S A*, **95**:460–465, 1998.
21. Nikolajewa S, Beyer A, Friedel M, Hollunder J, and Wilhelm T, Common patterns in type II restriction enzyme binding sites, *Nucleic Acids Res*, **33**:2726–2733, 2005.
22. Ogle JM, Brodersen DE, Clemons WM Jr, Tarry MJ, Carter AP, and Ramakrishnan V, Recognition of cognate transfer RNA by the 30S ribosomal subunit, *Science*, **292**:897–902, 2001.
23. Osawa S, Jukes TH, Watanabe K, and Muto A, Recent evidence for evolution of the genetic code, *Microbiol Rev*, **56**: 229–264, 1992.

December 9, 2005 12:54 WSPC/INSTRUCTION FILE jbcbl

The New Classification Scheme of the Genetic Code, its Early Evolution, and tRNA Usage 11

24. Patel A, The triplet genetic code had a doublet predecessor, *J Theor Biol*, **233**:527–532, 2005.
25. Pavese A, Conterio F, Bolchi A, Dieci G, and Ottonello S, Identification of new eukaryotic tRNA genes in genomic DNA databases by a multistep weight matrix analysis of transcriptional control regions, *Nucleic Acids Res*, **11**:1247–1256, 1994.
26. Reader JS and Joyce GF, A ribozyme composed of only two different nucleotides, *Nature*, **420**:841–844, 2002.
27. Rodin S, Ohno S, Rodin A, Transfer RNAs with complementary anticodons: could they reflect early evolution of discriminative genetic code adaptors? *Proc Natl Acad Sci USA*, **90**:4723–4727, 1993.
28. Sharp PM, Li WH, The codon Adaptation Index—a measure of directional synonymous codon usage bias, and its potential applications, *Nucleic Acids Res*, **15**:1281–1295, 1987.
29. Sprinzl M, Vassilenko KS, Emmerich J, Bauer F, Compilation of tRNA sequences and sequences of tRNA genes. Last update 2003. www.uni-bayreuth.de/departments/biochemie/trna/.
30. Stahl G, McCarty GP, Farabaugh PJ, Ribosome structure: revisiting the connection between translational accuracy and unconventional decoding, *Trends Biochem Sci*, **27**:178–183, 2002.
31. Sunami T, Kondo J, Kobuna T, Hirao I, Watanabe K, Miura K, Takenaka A, Crystal structure of d(GCGAAAGCT) containing a parallel-stranded duplex with homo base pairs and an anti-parallel duplex with Watson-Crick base pairs, *Nucleic Acids Res*, **30**:5253–5260, 2002.
32. Taylor FJ and Coates D, The code within the codons, *BioSystems*, **22**:177–187, 1989.
33. Tchurikov NA, Chistyakova LG, Zavilgelsky GB, Manukhov IV, Chernov BK, and Golova YB, Gene-specific silencing by expression of parallel complementary RNA in *Escherichia coli*, *J Biol Chem*, **275**:26523–26529, 2000.
34. Thompson JD, Higgins DG, and Gibson TJ, CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice, *Nucleic Acids Res*, **22**:4673–4680, 1994.
35. Tong KL, Wong JT, Anticodon and wobble evolution, *Gene*, **333**:169–177, 2004.
36. Wheeler DL, Chappey C, Lash AE, Leipe DD, Madden TL, Schuler GD, Tatusova TA, and Rapp BA, Database resources of the National Center for Biotechnology Information, *Nucleic Acids Res*, **28**:10–14, 2000.
37. Wilhelm T and Nikolajewa SL, A new classification scheme of the genetic code, *J Mol Evol*, **59**:598–605, 2004.
38. Woese CR, *The genetic code: The molecular basis for Genetic Expression*, Harper & Row, New York, 1967.
39. Wong J T, A co-evolution theory of the genetic code, *Proc Natl Acad Sci USA*, **72**:1909–1912, 1975.
40. Wu HL, Bagby S, van den Elsen JM, Evolution of the genetic triplet code via two types of doublet codons, *J Mol Evol*, **61**:54–64, 2005.
41. Yagil G, The over-representation of binary DNA tracts in seven sequenced chromosomes, *BMC GENOMICS*, **5**:(1):19, 2005.
42. Yarus M, Amino acids as RNA ligands: a direct-RNA-template theory for the code's origin, *J Mol Evol*, **47**:109–117, 1998.

December 9, 2005 12:54 WSPC/INSTRUCTION FILE jbec1

12 *Swetlana Nikolajewa, Maik Friedel, Andreas Beyer and Thomas Wilhelm*



Swetlana Nikolajewa received the Bachelor and Master degrees, both in applied mathematics, from Rostov State University (RSU), Russia, in 1997 and 1999, respectively. From 2003 she is with the Institute of Molecular Biotechnology, Jena, Germany, where she does doctoral study at the Theoretical Systems Biology Group.



Maik Friedel is student of Bioinformatics at the Friedrich-Schiller University, Jena, Germany. He is doing his diploma thesis at the Theoretical Systems Biology Group, Institute of Molecular Biotechnology, Jena, Germany.



Andreas Beyer received his degree in Applied Systems Science and his Ph.D. from the University of Osnabrück, Germany, in 1999 and 2002, respectively. Since 2002 he is working in the group of Thomas Wilhelm at the Institute of Molecular Biotechnology, Jena, Germany as a post-doc.



Thomas Wilhelm received his diploma degree in Biophysics, and his Ph.D. in Theoretical Biophysics from Humboldt-University Berlin, Germany, in 1992 and 1997, respectively. He is the head of the Theoretical Systems Biology Group at the Institute of Molecular Biotechnology, Jena, Germany.

Common Patterns in Type II Restriction Enzyme Binding Sites

The article of [Nikolajewa *et al.* \(2005\)](#) provides the statistical analysis of nucleotide patterns in restriction enzyme binding sites of type II, according to all possible binary coding schemes. The significant patterns are discussed in detail.

Published online May 11, 2005

2726–2733 *Nucleic Acids Research*, 2005, Vol. 33, No. 8
doi:10.1093/nar/gki575

Common patterns in type II restriction enzyme binding sites

Svetlana Nikolajewa, Andreas Beyer, Maik Friedel, Jens Hollunder and Thomas Wilhelm*

Institute of Molecular Biotechnology, Beutenbergstrasse 11, D-07745 Jena, Germany

Received April 1, 2005; Revised and Accepted April 26, 2005

ABSTRACT

Restriction enzymes are among the best studied examples of DNA binding proteins. In order to find general patterns in DNA recognition sites, which may reflect important properties of protein–DNA interaction, we analyse the binding sites of all known type II restriction endonucleases. We find a significantly enhanced GC content and discuss three explanations for this phenomenon. Moreover, we study patterns of nucleotide order in recognition sites. Our analysis reveals a striking accumulation of adjacent purines (R) or pyrimidines (Y). We discuss three possible reasons: RR/YY dinucleotides are characterized by (i) stronger H-bond donor and acceptor clusters, (ii) specific geometrical properties and (iii) a low stacking energy. These features make RR/YY steps particularly accessible for specific protein–DNA interactions. Finally, we show that the recognition sites of type II restriction enzymes are underrepresented in host genomes and in phage genomes.

INTRODUCTION

Protein–DNA interactions play a fundamental role in cell biology. For instance, the highly specific interactions between transcription factors and DNA are essential for proper gene expression regulation (1). The ‘immune system’ of bacteria and archaea relies on restriction endonucleases (REases) recognizing short sequences in foreign DNA with remarkable specificity and cleaving the target on both strands (2–4). REases are indispensable tools in molecular biology and biotechnology (5–7) and have been studied intensively because of their extraordinary importance for gene analysis and cloning work. In addition, they are important model systems for studying the general question of highly specific protein–nucleic acid interactions (2). REases also serve as examples for investigating structure–function relationships and for understanding the

evolution of functionally similar enzymes with dissimilar sequences (3).

Based on subunit composition, cofactor requirements, site specificity and mode of action REases have been classified into four types (8). Enzymes of types I, II and III are parts of restriction–modification (RM) systems, which additionally contain methyltransferases (MTases) adding methyl groups to cytosine or adenine in the host DNA. Type IV REases have no cognate MTases; they recognize and cleave sequences with already modified bases (9) and show only weak specificity (8). RM systems occur ubiquitously among bacteria and archaea (10–12). Their principal biological function is the protection of host DNA against foreign DNA, such as phages and conjugative plasmids (13). Other possible functions are to increase diversity by promoting recombination (13,14) and to act as selfish elements (15,16).

Here we study the recognition sequences of all known type II REases. The main criterion for classifying a restriction enzyme as type II is that it cleaves specifically within or close to its recognition site and does not require ATP hydrolysis. The orthodox type II REase is a homodimer recognizing a palindromic sequence of 4–8 bp. The possible advantage of symmetric recognition sites has already been discussed by the discoverers of restriction enzymes (17). They argued economically that it is ‘much cheaper to specify two identical subunits each capable of recognizing’ the half of the symmetrical sequence than to specify ‘a larger protein capable of recognizing the entire sequence’. This may explain the overwhelming majority of palindromic recognition sequences. However, there are other subtypes too—for instance, type IIA REases that recognize asymmetric sequences (8). Recently, the first example of a type II enzyme (MspI) where a monomer and not a dimer binds to a palindromic DNA sequence (18) has been found.

Much has been written about the evolution of REases. When elaborating on this topic Chinen *et al.* (19) wondered ‘Why are these recognition sequences so diverse?’ Here we show that these sequences are not as diverse as may appear at first sight. Typical patterns can be identified when focusing on purines and pyrimidines. This is apparent from Table 1, which shows the recognition sequences of all restriction enzymes with known three-dimensional structure.

*To whom correspondence should be addressed. Tel: +49 3641 65 6208; Fax: +49 3641 65 6191; Email: wilhelm@imb-jena.de

© The Author 2005. Published by Oxford University Press. All rights reserved.

The online version of this article has been published under an open access model. Users are entitled to use, reproduce, disseminate, or display the open access version of this article for non-commercial purposes provided that: the original authorship is properly and fully attributed; the Journal and Oxford University Press are attributed as the original place of publication with the correct citation details given; if an article is subsequently reproduced or disseminated not in its entirety but only in part or as a derivative work this must be clearly indicated. For commercial re-use, please contact journals.permissions@oupjournals.org

Table 1. All type II restriction enzymes with known three-dimensional structure and their cognate DNA recognition sequences [PDB, (20)]

Enzyme	Source	Recognition sequence ^a	Purine (1)–pyrimidine (0) pattern
MspI	<i>Moraxella</i> species	CCGG	0011
FokI	<i>Flavobacterium okeanoicoles</i>	GGATG	11101
EcoRII	<i>Escherichia coli</i>	CCWGG	00W11
EcoRI	<i>E. coli</i>	GAATC	111000
BamHI	<i>Bacillus amyloliquefaciens</i>	GGATCC	111000
HindIII	<i>Haemophilus influenzae</i>	AAGCTT	111000
BglII	<i>Bacillus globigii</i>	AGATCT	111000
BstYI	<i>Bacillus stearothermophilus</i>	RGATCY	111000
EcoRV	<i>E. coli</i>	GATATC	110100
Cfr10I	<i>Citrobacter freundii</i>	RCCGGY	100110
NaeI	<i>Nocardia aerocolonigenes</i>	GCCGGC	100110
NgoMIV	<i>Neisseria gonorrhoeae</i>	GCCGGC	100110
HincII	<i>H. influenzae Rc</i>	GTyrAC	100110
Bse634I	<i>Bacillus</i> species 634	RCCGGY	100110
MunI	<i>Mycoplasma</i> species	CAATTG	011001
PvuII	<i>Proteus vulgaris</i>	CAGCTG	011001
BsoBI	<i>B. stearothermophilus</i>	CYCGRG	000111
EcoO109I	<i>E. coli</i>	RGGNCCY	111N000
BglI	<i>B. globigii</i>	GCCNNNNNGGC	100NNNNN110

The corresponding purine (1)–pyrimidine (0) coding shows that 11/00 is a common pattern in all binding sites.

^aRecognition sequence representations use the standard abbreviations (21) to represent ambiguity. R = G or A; K = G or T; S = G or C; B = not A (C or G or T); D = not C (A or G or T); Y = C or T; M = A or C; W = A or T; H = not G (A or C or T); V = not T (A or C or G) and N = A or C or G or T.

MATERIALS AND METHODS

All restriction enzyme binding sites were taken from REBASE [last update March 3, 2005 (10)]. Almost all (98%) known REase recognition sequences belong to type II enzymes. We separated the type II binding sites into symmetric and asymmetric sequences, with just 0.96% belonging to the latter class.

The statistical analysis of sequence patterns is based on counting the frequency of all possible substrings up to a length of 4 bp in the symmetric and asymmetric binding sequences (see Supplementary Table S2). In addition to counting substrings of the actual nucleotide sequence, we also counted substrings according to two different binary coding schemes: purine–pyrimidine coding and ketobase–aminobase coding. For the substring analyses of symmetric sequences we consider only the first half of each sequence (the second half is redundant).

Using a binomial distribution, we calculated *P*-values that quantify the probability of finding the respective subsequence in a randomized set of binding sites at least as often as in the original binding sites. The *P*-values take account of the relative abundance of each letter (A, G, R, N etc.) in the binding sites (see Supplementary Table S1).

Analysis of dinucleotide H-bond donor and acceptor clusters

We selected B-DNA crystal structures from PDB (20) with X-ray diffraction resolution ≤ 1.5 Å. Only structures with

Watson–Crick base-pairing, without mismatches and without additional ligands were taken into account. The selected PDB entries are 1D8G, 1D8X, 1D23, 1D49, 1EN3, 1EN8, 1ENN, 232D and 295D. The first and last nucleotides in each sequence were omitted from the analysis.

We calculated the average distance between two canonical (22) H-bond donors (and between two acceptors, respectively), each one belonging to one of two adjacent bases. Donor and acceptor pairs must be oriented towards the major or minor groove; pairs with one partner on the major and one partner on the minor groove were omitted. The DNA backbone was not considered for this analysis. Reported distances are averages for the nine selected crystal structures (see Supplementary Table S3). For each dinucleotide base pair we summed all corresponding reciprocal distance values and thus obtained a quantitative measure for H-bond donor and acceptor clusters of each dinucleotide base pair in the major or minor groove (see Supplementary Table S3). The resulting value integrates the number of acceptors/donors and their distance. Simply counting the number of donor and acceptor pairs gives similar results.

Analysis of DNA geometry and flexibility

We analysed four different datasets for the dinucleotide parameters roll, tilt and twist, and three datasets for shift, slide and rise (see Supplementary Table S4). Olson *et al.* (23) analysed the flexibility in all these six parameters deduced from protein–DNA and pure DNA crystal complexes (yielding two datasets: OlsDNA and OlsProt-DNA). Scipioni *et al.* (24) deduced the flexibility in roll, tilt and twist from scanning force microscopy images (dataset Scip). Recently (25), all six parameters were calculated from an extensive analysis of structural databases (dataset Per). These authors also found an excellent agreement between database analysis and corresponding molecular dynamics simulations.

RESULTS

Currently, a total of 3726 different REases from 281 bacterial and 26 archaeal genomes are known (REBASE, last update March 3, 2005). The class type II alone comprises 3654 different REases, recognizing 257 different binding sites (the remainder are isoschizomers). Among these are 176 symmetric sequences (mostly recognized by homodimers) and 81 asymmetric sequences. We statistically analysed all type II binding sites and additionally the small datasets of type I, type III and homing endonucleases.

High GC content in DNA binding sites

Our first observation is the significantly enhanced GC content in all type II binding sites: 68% GC and 32% AT. Ambiguous letters (N, R, Y, K and M) were not taken into account (for the complete statistics of base compositions of type II binding sites, see Supplementary Table S1). In contrast, the mean GC content of the host genomes as well as that of the bacteriophages is on average $\sim 50\%$. The GC content of the binding sites thus deviates significantly from this genome-wide average ($P < 10^{-300}$). We argue that this significantly enhanced GC content reflects biological functionality of the binding sites. Three different facts could play a role in this

2728 *Nucleic Acids Research*, 2005, Vol. 33, No. 8

context. (i) In order to protect themselves, hosts have to methylate the specific binding sites in their own genomes. This happens by methylation of either adenine or cytosine. There are two different methylation sites in cytosine [yielding N4-methylcytosine (m4) and C5-methylcytosine (m5)], but only one methylation site in adenine [yielding N6-methyladenine (m6)] (26). All the known results of methylation sensitivity experiments are collected in REBASE (10). We have counted all m4, m5 and m6 methylations that reliably prevent DNA cutting and found 146, 1350 and 524 methylations, respectively. Evolution may therefore have favoured cytosines (over adenines) in RM binding sites. (ii) GC-rich sequences are more stable than AT-rich sequences because of the better stacking interactions. Furthermore, G and C always form three H-bonds in complementary base-pairing and therefore have a higher binding strength than A and T, which pair with two H-bonds. MTases and endonucleases (like other DNA binding proteins) recognize sequences on a bound double strand better than those on open DNA without H-bonds between the two strands at the 'open' site. However, the third fact seems to be the most relevant reason for the high GC content. (iii) One A–T base pair allows for five canonical H-bonds between the bases and the recognizing amino acids, whereas the G–C base pair allows for up to six H-bonds (22), which may be beneficial for protein binding. Generally, type II restriction enzymes exhaust the hydrogen bonding potential of their recognition sequence. In contrast, homing endonucleases do not fully exhaust the hydrogen bonding potential. In support of this notion, the mean GC content in homing

enzyme binding sites is only 46% (see Supplementary Table S8).

As a generalization one might hypothesize that an enhanced GC content may be an important property of protein binding DNA sequences whenever high specificity is needed. It was found that GC-rich DNA sequences have a higher CAP-binding affinity than AT-rich sites (27) (CAP—*Escherichia coli* catabolite gene activator protein).

Enhanced occurrence of RR/YY dinucleotides in DNA binding sites

We separated the type II enzyme recognition sequences into symmetric and asymmetric sequences. In the case of the former we analysed only the first half of the sequence. For these two subsets we counted the occurrence of subsequences up to size 4 and calculated the corresponding *P*-values (see Materials and Methods and Supplementary Table S2). The most abundant dinucleotides are GG and CC. However, owing to the high GC content (which affects the *P*-value) the most significant dinucleotide is GA ($P < 10^{-69}$ in the symmetric dataset). Other substrings, such as CTG ($P < 10^{-57}$ in the symmetric dataset) are similarly significant. A much clearer picture is obtained by considering substrings according to the two different binary coding schemes: purine–pyrimidine coding and ketobase–aminobase coding. Table 2 shows that the two dinucleotides RR and YY are the most significant patterns in the large symmetric dataset. In the much smaller asymmetric set, RRR, YYY and YYYY are even more significant, but

Table 2. Purine–pyrimidine and ketobase–aminobase patterns in type II restriction enzyme recognition sequences

Pattern	Symmetrical recognition sequences		Keto (1)–amino (0)		Asymmetrical recognition sequences		Keto (1)–amino (0)	
	Purine (1)–pyrimidine (0) Frequency	<i>P</i> -value	Frequency	<i>P</i> -value	Purine (1)–pyrimidine (0) Frequency	<i>P</i> -value	Frequency	<i>P</i> -value
00	1758	6.6E–63	1097	0.61	529	5.1E–12	294	1
01	817	1	1060	1	214	1	379	0.59
10	903	1	1278	0.01	348	0.98	524	2.0E–15
11	1743	1.7E–29	1389	0.01	501	4.7E–14	380	0.69
000	348	5.5E–08	78	1	288	1.5E–24	62	1
001	328	1.8E–08	250	9.3E–06	81	1	160	0.07
010	89	1	250	9.3E–06	79	1	210	1.0E–08
011	165	0.99	302	3.3E–10	102	0.99	129	0.92
100	269	0.04	194	0.41	140	0.79	142	0.52
101	105	1	117	1	104	0.99	156	0.16
110	264	0.00	271	1.8E–05	193	1.0E–05	210	3.1E–08
111	310	8.3E–13	132	1	231	1.5E–15	128	0.95
0000					150	3.2E–27	14	1
0001	3	0.59	2	0.92	24	0.99	31	0.99
0010					26	0.99	91	3.4E–08
0011	1	0.94	3	0.42	47	0.74	53	0.36
0100	4	0.36	1	0.98	32	0.99	31	0.99
0101					9	1	34	0.99
0110			1	0.90	35	0.92	81	2.4E–05
0111			5	0.01	39	0.90	27	0.99
1000	8	0.01	1	0.98	78	0.00	14	1
1001					18	1	83	8.2E–06
1010	1	0.94	2	0.68	36	0.99	89	2.3E–07
1011	7	0.01	5	0.01	45	0.73	44	0.86
1100	3	0.54	4	0.21	82	2.7E–05	24	0.99
1101	2	0.74	2	0.41	52	0.34	109	2.0E–13
1110					88	1.4E–07	91	1.2E–07
1111			2	0.20	94	2.3E–10	20	1

In the pur–pyr coding 1 stands for purine (A, G, R) and 0 for pyrimidine (T, C, S), and in the keto–amino coding 1 stands for a ketobase (G, T, K) and 0 for an aminobase (A, C, M).

RR and YY also stand out. In addition, Table 2 shows that there is no comparably significant ketobase–aminobase pattern. Thus, purine–pyrimidine classification seems to be biologically more important than the ketobase–aminobase categorization. This is also underlined by the fact that among all type II recognition sites the number of Rs and Ys (ambiguous binding sites) is about a factor of 26 higher than the number of Ks and Ms (Supplementary Table S1). REases sometimes allow for some degree of ambiguity, as long as the required purine–pyrimidine pattern is ensured.

The high statistical significance of two and more consecutive purines (or pyrimidines) in type II enzyme binding sites points to biological relevance. We present evidence for three mechanisms that are potentially responsible for the observed enrichment of this pattern.

(i) *H-bond donor and acceptor clusters.* RR/YY steps provide on average stronger H-bond donor (example in Figure 1) and acceptor clusters than other dinucleotides (see Materials and Methods and Supplementary Table S3). Close proximity of acceptor pairs (or donor pairs) on the DNA allows for the establishment of bifurcated H-bonds, which are stronger than canonical single donor–single acceptor interactions. This feature of RR/YY steps potentially facilitates the recognition by and binding of interacting proteins (28). Supplementary Table S3 shows that the average cluster strength of RR/YY steps is higher than that of all other steps. The only (very weak) exception are acceptor clusters in the minor groove, resulting from low strength of the GG/CC step. However, this is counterbalanced by the strong acceptor cluster in the major groove and the donor clusters in the major and minor groove of the GG/CC step. Figure 1 shows an example of a single amino acid (of EcoRI) that potentially interacts with three consecutive purines (GAA) and establishes a bifurcated H-bond.

However, there is growing evidence that specific protein–DNA binding is accomplished not only by specific chemical contacts, but also by suitable geometrical arrangement of the

DNA and by its propensity to adopt a deformed conformation facilitating the protein binding (29). The following points (ii and iii) show that both properties are better fulfilled by two adjacent purines (or pyrimidines) than by other dinucleotides.

(ii) *Geometrical arrangement.* RR/YY steps allow for a special geometrical arrangement of the DNA (see Materials and Methods and Supplementary Table S4). RR/YY steps are characterized by (a) minimal slide values, without exception; (b) strong tilt in the negative direction [dataset Per deviates somewhat, but ‘tilt is a parameter very sensitive to the choice of calculation method’ (30) and, thus, the consistency of the other three datasets seems remarkable]; and (c) a positive roll in all datasets, which implies positive bending towards the major groove (25). The only exception is the AA/TT step in the Scip dataset. However, AA/TT is by far the least significant dinucleotide of all RR/YY steps (Supplementary Table S2).

(iii) *Stacking energy.* RR/YY steps have a low stacking energy (25) and seem therefore well suited to the often necessary conformational changes during specific protein binding (23,31). Moreover, the stacking energy of all RR/YY steps is anticorrelated with the statistical significance of the RR/YY subsequences (Supplementary Tables S2 and S4). AA/TT has the highest stacking energy and the lowest significance, whereas GATC has the lowest stacking energy and the highest significance.

Probably, all three possible reasons for an enhanced frequency of RR/YY steps in type II REase binding sites together play a role in the corresponding specific DNA recognition.

In asymmetric binding sequences longer chains of purines or pyrimidines, such as RRR, YYY and YYYYY, are even more significant than RR/YY steps. This could indicate that such substrings are preferred in binding sites. Some dinucleotide parameters, such as stacking energy, more or less add up in longer sequences. On the other hand, a negative correlation between motions at a given base pair step and neighbouring steps was found for most helical coordinates (32).

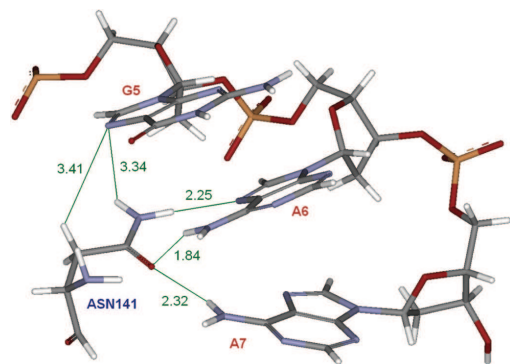


Figure 1. Example of an interaction between an H-bond donor cluster (resulting from two adjacent purines AA) and an H-bond acceptor (bifurcated hydrogen bond). The figure shows binding of residue Asn141 from EcoRI to the DNA subsequence 5'-D(GAA)-3' (only one strand shown). Green lines indicate potential hydrogen donor–acceptor pairs; distances are in angstroms. The structure is according to PDB entry 1CKQ. Note the bending towards the major groove, which reduces the distances between the H-bond donors of the two adenines.

Binding sites are underrepresented in host and phage genomes

The typical features of type II restriction enzyme binding sites, high GC content and overrepresentation of RR/YY steps, could also be linked to the frequency of these sites in the host and/or phage genomes. To address this question we analysed the genome of *E. coli* K12 and the known genomes of its phages (33). All four bases are almost equally abundant in both the *E. coli* genome and the genomes of its phages. Based on this information we can estimate the expected frequency of any given sequence in a randomized genome. Enrichments of sequences are quantified as the ratio of observed versus expected frequency. In addition we calculated weighted ratios, taking into account the number of different enzymes recognizing the same sequence (Supplementary Table S5).

Three findings arise from this analysis: (i) most binding sites are underrepresented in both the host and the phage genomes (possible explanations are that phages try to escape REases and that hosts minimize the methylation effort); (ii) under(over)representation in host and phage genomes is correlated; and (iii) under(over)representation is correlated with GC content and RR/YY frequency (most underrepresented sequences contain only GC and always contain RR/YY steps). This

2730 *Nucleic Acids Research*, 2005, Vol. 33, No. 8

correlation again underlines the biological importance of these two features.

DISCUSSION

We presented a statistical analysis of all known DNA recognition sites of type II restriction enzymes. This collection comprises by far the largest group of reliably known specific protein binding sites on DNA. There is hardly any sequence similarity among restriction enzymes (34). REases often use uncommon DNA binding motifs (35), but sometimes also typical structures already known from transcription factors, such as FokI and NaeI, which both use a helix–turn–helix motif. The typical features of type II REase binding sites such as high GC content and many RR/YY steps may also be relevant for other DNA recognition sequences. We have also analysed all known binding sites of type I and type III restriction enzymes and of homing endonucleases (Supplementary Tables S6–S8). However, we found no statistically significant motifs, which is probably due to the small number of sequences of these types. Homing endonucleases are known to bind less specifically (10,36). This lack of specificity could be another explanation for the lack of statistically significant patterns among this class of binding sites. Table 3 shows examples of other DNA binding proteins along with their recognition sequences. Nearly all of them contain RR/YY steps. The average GC content of these sequences is 54%.

We presented three different possible explanations for the amplified occurrence of two neighbored purines (or pyrimidines) in the recognition sites. One argument is that these give stronger H-bond donor and acceptor clusters than any other adjacent base pair and therefore facilitate hydrogen bonds to amino acids. For instance, EcoRV (binding GATATC) establishes multiple contacts to the first 2 bp and the last 2 bp, but none to the middle 2 bp (60).

Evolutionary relatedness of REases recognizing similar sequences would be a completely different explanation for our observed patterns. Although only a few REase crystal structures have been solved so far, it became clear from additional bioinformatics studies that REases belong to at least four unrelated and structurally distinct superfamilies: PD-(D/E)XK, PLD, HNH and GIY-YIG (34). The largest one [PD-(D/E)XK] comprises the two major classes α (EcoRI-like) and β (EcoRV-like) (2). Enzymes belonging to the same superfamily sometimes also have similar recognition sequences. For instance, Eco29kI, NgoMIII and MraI, which are related to the GIY-YIG superfamily, all bind to CCGCGG (61). HpyI (CATG), NlaIII (CATG), SphI (GCATGC), NspHI (RCATGY), NspI (RCATGY), MboII (GAAGA) and KpnI (GGTACC) belong to the HNH superfamily (62), and SsoII (CCNGG), EcoRII (CCWGG), NgoMIV (GCCGGC), PspGI (CCWGG) and Cfr10I (RCCGGY) to the EcoRI branch (63). It has already been argued that these enzymes diverged early in evolution, presumably from a type IIP enzyme that recognized

Table 3. Examples of gene regulatory proteins that recognize specific short DNA sequences

DNA binding protein	Recognition sequence (or consensus motif)	Purine (1)–pyrimidine (0) pattern	References
p53	RRRCW ₆ GYYYRRRCW ₂ GYYY	1110W ₂ 10001110W ₂ 1000	(38)
MADS box	CCW ₆ GG	00W ₆ 11	(39)
ERSE	CCAATN ₆ CCACG	00110N ₆ 00101	(40)
Ski oncoprotein	GTCTAGAC	10001110	(41)
GAL4	CGGN ₅ TN ₅ CCG	011N ₅ 0N ₅ 001	(42)
GAL4 <i>in vitro</i>	WGGN ₁₀₋₁₂ CCG	W11N ₁₀₋₁₂ 001	(42)
nkx-2.5	CWTTAATTN	0W001100N	(43)
Bicoid	TCTAATCCC	000110000	(44)
AP-2	GCCCCAGGC	100001110	(45)
Stat5-RE	TTCN ₃ GAA	000N ₃ 111	(46)
GRE	AGAACAN ₃ TGTTCT	111101N ₃ 010000	(46)
SRF	CCW ₂ AW ₃ GG	00W ₂ 1W ₃ 11	(47)
MCM1	CCYW ₃ N ₂ GG	000W ₃ N ₂ 11	(47)
NFκB	GGGACTTTCC	111100000	(48)
<i>pur</i> repressor	ANGCAANCGNTTNCNT	1N1011N01N00N0N0	(49)
YY1	GGCCATCTTG	1100100001	(50)
NF-1/CTF-1	TGGN ₆ GCCAA	011N ₆ 10011	(51)
PPAR	AGGAAACTGGA	11111100111	(52)
NFAT	ATTGGAAA	10011111	(53)
CREA	GCGGAGACCCAG	1011111000011	(54)
C/EBP	CCAAT	00110	(55)
PacC	GCCARG	100111	(56)
TTK finger1	GAT	110	(57)
TTK finger2	AGG	111	(57)
Zif finger1	GCG	101	(57)
Zif finger2	TGG	011	(57)
GLI finger4	TTGGG	00111	(57)
GLI finger5	GACC	1100	(57)
<i>E. coli</i> sigma factors (binding in –35 region)			(58–60)
σ70 (primary)	CTTGA	00011	
σ32 (heat shock)	CTTGAA	000111	
σ60 (nitr. reg. gene)	CTGGNA	0011N1	
σ54 (nit. ox. stress)	TTGG CACG	0011 0101	
σ28 (exter. stress)	CTAAA	00111	

CCxGG or xCCGGx (63). We are not aware of any systematic study of recognition sequence similarity versus membership in superfamilies. However, it is conceivable that sequence similarity (or the corresponding purine–pyrimidine pattern) is evolutionarily conserved. Some positive correlation between amino acid similarity and recognition sequence similarity of restriction enzymes has already been found (64). However, REases are extremely divergent and mostly structurally and evolutionarily unclassified (34). Even related enzymes binding to similar DNA sequences may differ much in the details of protein–DNA interaction. Comparing the cocrystal structures of the related enzymes BamHI and EcoRI, it has been inferred that none of the interactions could have been anticipated from the other structure (65). Lukacs and Aggarwal (66) studied the structures of two related enzyme pairs BglII (AGATCT) versus BamHI (GGATCC) and MnlI (CAATTG) versus EcoRI (GAATTC), which both differ in only the outer base of the binding site. For the first pair they found ‘surprising diversity’ in how the common base pairs are recognized, whereas the enzymes of the second pair recognize their common inner and middle base pairs in a nearly identical manner.

The problem of recognition and binding of a protein to its specific DNA sequence is far from being solved. Heitman and Model (35) substituted amino acids in the binding domain of EcoRI such that some of the original 12 hydrogen bonds contacting the base pairs of the recognition sequence could not be established by the mutant. This change did not affect the binding specificity of EcoRI, but only its enzymatic activity. It was concluded that the hydrogen bonds revealed by the crystal structure are insufficient to fully account for substrate recognition, and additional amino acids must contact the DNA to help discern the substrate (35). The authors argued that protein–DNA interactions can be influenced by sequence-dependent variation of the structure of the DNA backbone [originally suggested by Dickerson (67)], and that the EcoRI enzyme could recognize its cognate sequence because it adopts its unusual bound conformation more readily than other DNA sequences. It was concluded that even with a detailed cocrystal structure it is exceedingly difficult to determine which interactions contribute to sequence-specific DNA recognition (35). Moreover, it has been found that protein binding to DNA is modulated by sequence context outside the recognition site (68) and that different endonucleases have different context preferences (69).

Our work suggests that sometimes only the purine–pyrimidine pattern matters for recognition by a certain biomolecule. Note that R and Y are most frequent among the ambiguous letters in restriction enzyme binding sites. In such cases the exact base would be irrelevant as long as it is a purine (or pyrimidine). Several such examples are already known. For instance, during translation the third base of the codon is nearly always analysed in this binary manner (in the yeast mitochondrial code this is always the case) (70). Another example is the sequential contact model for EcoRI, proposing that during the transition from DNA binding to DNA scission, the contacts to the pyrimidines could either precede or follow the purine contacts observed in the crystal structure (35). It is known that a change in just 1 bp of the cognate site can reduce the ratio k_{cat}/K_m for DNA cleavage by a factor of $>10^6$ (71). Thus, a transition exchange might generally have a less dramatic effect than a transversion exchange. Such a smaller

effect of a transition exchange could also be observed in corresponding pausing experiments (72), which might be important for protein engineering.

SUPPLEMENTARY MATERIAL

Supplementary Material is available at NAR Online.

ACKNOWLEDGEMENTS

We thank two anonymous referees for valuable comments. This work has been supported by the Bundesministerium für Bildung und Forschung (Grant 0312704E). Funding to pay the Open Access publication charges for this article was provided by the Institute of Molecular Biotechnology.

Conflict of interest statement. None declared.

REFERENCES

- Beyer, A., Hollunder, J., Nasheuer, H.-P. and Wilhelm, T. (2004) Post-transcriptional expression regulation in the yeast *Saccharomyces cerevisiae* on a genomic scale. *Mol. Cell. Proteomics*, **3**, 1083–1092.
- Pingoud, A. and Jeltsch, A. (2001) Structure and function of type II restriction endonucleases. *Nucleic Acids Res.*, **29**, 3705–3727.
- Bujnicki, J.M. (2003) Crystallographic and bioinformatic studies on restriction endonucleases: inference of evolutionary relationships in the ‘midnight zone’ of homology. *Curr. Protein Pept. Sci.*, **4**, 327–337.
- Pingoud, A.M. (2004) Restriction endonucleases. In Gross, H.J. (ed.), *Nucleic Acids and Molecular Biology*. Springer-Verlag, Berlin, Heidelberg, Vol. 14, pp. 442.
- Chandrasegaran, S. and Smith, J. (1999) Chimeric restriction enzymes: what is next? *Biol. Chem.*, **380**, 841–848.
- Williams, R.J. (2001) Isolation and characterization of an unknown restriction endonuclease. *Methods Mol. Biol.*, **160**, 431–442.
- Jenkins, G.J., Williams, G.L., Beynon, J., Ye, Z., Baxter, J.N. and Parry, J.M. (2002) Restriction enzymes in the analysis of genetic alterations responsible for cancer progression. *Br. J. Surg.*, **89**, 8–20.
- Roberts, R.J., Belfort, M., Bestor, T., Bhagwat, A.S., Bickle, T.A., Bitinaite, J., Blumenthal, R.M., Degtyarev, S.Kh., Dryden, D.T., Dybvig, K. et al. (2003) A nomenclature for restriction enzymes, DNA methyltransferases, homing endonucleases and their genes. *Nucleic Acids Res.*, **31**, 1805–1812.
- Bickle, T.A. and Kruger, D.H. (1993) Biology of DNA restriction. *Microbiol. Rev.*, **57**, 434–450.
- Roberts, R.J., Vincze, T., Posfai, J. and Macelis, D. (2005) REBASE—restriction enzymes and DNA methyltransferases. *Nucleic Acids Res.*, **33**, D230–D232.
- Roberts, R.J. and Halford, S.E. (1993) Type II restriction endonucleases. In Linn, S.M., Lloyd, R.S. and Roberts, R.J. (eds), *Nucleases*. Cold Spring Harbor Laboratory Press, Cold Spring Harbor, New York, NY, pp. 35–88.
- Raleigh, E.A. and Brooks, J.E. (1998) In De Bruijn, F.J., Lupski, J.R. and Weinstock, G.M. (eds), *Bacterial Genomes*. Chapman and Hall, New York, pp. 78–92.
- Arber, W. (1979) Promotion and limitation of genetic exchange. *Science*, **205**, 361–365.
- Price, C. and Bickle, T.A. (1986) A possible role for DNA restriction in bacterial evolution. *Microbiol. Sci.*, **3**, 296–299.
- Naito, T., Kusano, K. and Kobayashi, I. (1995) Selfish behavior of restriction-modification systems. *Science*, **267**, 897–899.
- Kobayashi, I. (2001) Behavior of restriction-modification systems as selfish mobile elements and their impact on genome evolution. *Nucleic Acids Res.*, **29**, 3742–3746.
- Kelly, T.J. and Smith, H.O. (1970) A restriction enzyme from *Hemophilus influenzae* II. Base sequence of the recognition site. *J. Mol. Biol.*, **51**, 393–409.
- Xu, Q.S., Kucera, R.B., Roberts, R.J. and Guo, H.C. (2004) An asymmetric complex of restriction endonuclease MspI on its palindromic DNA recognition site. *Structure*, **12**, 1741–1747.

2732 *Nucleic Acids Research*, 2005, Vol. 33, No. 8

19. Chinen,A., Naito,Y., Handa,N. and Kobayashi,I. (2000) Evolution of sequence recognition by restriction-modification enzymes: selective pressure for specificity decrease. *Mol. Biol. Evol.*, **17**, 1610–1619.
20. Berman,H.M., Westbrook,J., Feng,Z., Gilliland,G., Bhat,T.N., Weissig,H., Shindyalov,I.N. and Bourne,P.E. (2000) The protein data bank. *Nucleic Acids Res.*, **28**, 235–242.
21. Nomenclature Committee of the International Union of Biochemistry. (1985) Nomenclature for incompletely specified bases in nucleic acid sequences. Recommendations 1984. *Eur. J. Biochem.*, **150**, 1–5.
22. Saenger,W. (1984) *Principles of Nucleic Acid Structure*. Springer-Verlag, NY.
23. Olson,W.K., Gorin,A.A., Lu,X.-J., Hock,L.M. and Zhurkin,V.B. (1998) DNA sequence-dependent deformability deduced from protein–DNA crystal complexes. *Proc. Natl Acad. Sci. USA*, **95**, 11163–11168.
24. Scipioni,A., Anselmi,C., Zuccheri,G., Samori,B. and De Santis,P. (2002) Sequence-dependent DNA curvature and flexibility from scanning force microscopy images. *Biophys. J.*, **83**, 2408–2418.
25. Pérez,A., Noy,A., Lanksa,F., Luque,F.J. and Orozco,M. (2004) The relative flexibility of B-DNA and A-RNA duplexes: database analysis. *Nucleic Acids Res.*, **32**, 6144–6151.
26. Bujnicki,J.M. (2001) Understanding the evolution of restriction-modification systems: clues from the sequence and structure comparisons. *Acta Biochim. Pol.*, **48**, 935–967.
27. Gartenberg,M.R. and Crothers,D.M. (1988) DNA sequence determinants of CAP-induced bending and protein binding affinity. *Nature*, **333**, 824–829.
28. Parra,R.D., Furukawa,M., Gong,B. and Zeng,X.C. (2001) Energetics and cooperativity in three-center hydrogen bonding interactions. I. Diacetamide-X dimers (X=HCN, CH₃OH). *J. Chem. Phys.*, **115**, 6030–6035.
29. Lanksa,F. (2004) DNA sequence-dependent deformability—insights from computer simulations. *Biopolymers*, **73**, 327–339.
30. Lu,X.J. and Olson,W.K. (1999) Resolving the discrepancies among nucleic acid conformational analyses. *J. Mol. Biol.*, **285**, 1563–1575.
31. Rozenberg,H., Rabinovich,D., Frolow,F., Hegde,R.S. and Shakked,Z. (1998) Structural code for DNA recognition revealed in crystal structures of papillomavirus E2-DNA targets. *Proc. Natl Acad. Sci. USA*, **95**, 15194–15199.
32. Zacharias,M. and Sklenar,H. (2000) Conformational deformability of RNA: a harmonic mode analysis. *Biophys. J.*, **78**, 2528–2542.
33. Hallin,P.F. and Ussery,D. (2004) CBS Genome Atlas Database: a dynamic storage for bioinformatic results and sequence data. *Bioinformatics*, **20**, 3682–3686.
34. Chmiel,A.A., Bujnicki,J.M. and Skowronek,K.J. (2005) A homology model of restriction endonuclease SfiI in complex with DNA. *BMC Struct. Biol.*, **5**, 2.
35. Heitman,J. and Model,P. (1990) Substrate recognition by the EcoRI endonuclease. *Proteins*, **7**, 185–197.
36. Jurica,M.S. and Stoddard,B.L. (1999) Homing endonucleases: structure, function and evolution. *Cell. Mol. Life Sci.*, **55**, 1304–1326.
37. Bian,J. and Sun,Y. (1997) p53CP, a putative p53 competing protein that specifically binds to the consensus p53 DNA binding sites: a third member of the p53 family? *Proc. Natl Acad. Sci. USA*, **94**, 14753–14758.
38. Parenicova,L., de Folter,S., Kieffer,M., Horner,D.S., Favalli,C., Busscher,J., Cook,H.E., Ingram,R.M., Kater,M.M., Davies,B. et al. (2003) Molecular and phylogenetic analyses of the complete MADS-box transcription factor family in Arabidopsis: new openings to the MADS world. *Plant Cell*, **15**, 1538–1551.
39. Yoshida,H., Haze,K., Yanagi,H., Yura,T. and Mori,K. (1998) Identification of the *cis*-acting endoplasmic reticulum stress response element responsible for transcriptional induction of mammalian glucose-regulated proteins. Involvement of basic leucine zipper transcription factors. *J. Biol. Chem.*, **273**, 33741–33749.
40. Nicol,R. and Stavnezer,E. (1998) Transcriptional repression by v-Ski and c-Ski mediated by a specific DNA binding site. *J. Biol. Chem.*, **273**, 3588–3597.
41. Vashee,S., Xu,H., Johnston,S.A. and Kodadek,T. (1993) How do ‘Zn2 cys6’ proteins distinguish between similar upstream activation sites? Comparison of the DNA-binding specificity of the GAL4 protein *in vitro* and *in vivo*. *J. Biol. Chem.*, **268**, 24699–24706.
42. Chen,C.Y. and Schwartz,R.J. (1995) Identification of novel DNA binding targets and regulatory domains of a murine tinman homeodomain factor, nfx-2.5. *J. Biol. Chem.*, **270**, 15628–15633.
43. Burz,D.S., Rivera-Pomar,R., Jäckle,H. and Hanes,S.D. (1998) Cooperative DNA-binding by Bicoid provides a mechanism for threshold-dependent gene activation in the *Drosophila* embryo. *EMBO J.*, **17**, 5998–6009.
44. Nakayama,M., Takahashi,K., Kitamura,T., Murakami,O., Shirato,K. and Shibahara,S. (2000) Transcriptional control of adrenomedullin induction by phorbol ester in human monocytic leukemia cells. *Eur. J. Biochem.*, **267**, 3559–3566.
45. Stoecklin,E., Wissler,M., Moriggi,R. and Groner,B. (1997) Specific DNA binding of Stat5, but not of glucocorticoid receptor, is required for their functional cooperation in the regulation of gene transcription. *Mol. Cell. Biol.*, **17**, 6708–6716.
46. Nurrish,S.J. and Treisman,R. (1995) DNA binding specificity determinants in MADS-box transcription factors. *Mol. Cell. Biol.*, **15**, 4076–4085.
47. Karin,M., Yamamoto,Y. and Wang,Q.M. (2004) The IKK NF- κ B system: a treasure trove for drug development. *Nature Rev. Drug Discov.*, **3**, 17–26.
48. Pabo,C.O. and Sauer,R.T. (1984) Protein–DNA recognition. *Annu. Rev. Biochem.*, **53**, 293–321.
49. Sakamuro,D. and Prendergast,G.C. (1999) New Myc-interacting proteins: a second Myc network emerges. *Oncogene*, **18**, 2942–2954.
50. Wenzelides,S., Altmann,H., Wendler,W. and Winnacker,E.L. (1996) CTF5—a new transcriptional activator of the NF1/CTF family. *Nucleic Acids Res.*, **24**, 2416–2421.
51. Mandard,S., Muller,M. and Kersten,S. (2004) Peroxisome proliferator-activated receptor alpha target genes. *Cell. Mol. Life Sci.*, **61**, 393–416.
52. Northrop,J.P., Ho,S.N., Chen,L., Thomas,D.J., Timmerman,L.A., Nolan,G.P., Admon,A. and Crabtree,G.R. (1994) NF-AT components define a family of transcription factors targeted in T-cell activation. *Nature*, **369**, 497–502.
53. Cubero,B. and Scazzocchio,C. (1994) Two different, adjacent and divergent zinc finger binding sites are necessary for CRE-mediated carbon catabolite repression in the proline gene cluster of *Aspergillus nidulans*. *EMBO J.*, **13**, 407–415.
54. Lékstrom-Himes,J. and Xanthopoulos,K.G. (1998) Biological role of the CCAAT/enhancer-binding protein family of transcription factors. *J. Biol. Chem.*, **273**, 28545–28548.
55. Espeso,E.A. and Penalva,M.A. (1996) Three binding sites for the *Aspergillus nidulans* PacC zinc-finger transcription factor are necessary and sufficient for regulation by ambient pH of the isopenicillin N synthase gene promoter. *J. Biol. Chem.*, **271**, 28825–28830.
56. Alberts,B., Johnson,A., Lewis,J., Raff,M., Roberts,K. and Walter,P. (2002) *Molecular Biology of the Cell*, 4th edn. Garland Publishing, NY.
57. Harley,C.B. and Reynolds,R.P. (1987) Analysis of *E. coli* promoter sequences. *Nucleic Acids Res.*, **15**, 2343–2361.
58. Jishage,M., Iwata,A., Ueda,S. and Ishihama,A. (1996) Regulation of RNA polymerase sigma subunit synthesis in *Escherichia coli*: intracellular levels of four species of sigma subunit under various growth conditions. *J. Bacteriol.*, **178**, 5447–5451.
59. Gardner,A.M., Gessner,C.R. and Gardner,P.R. (2003) Regulation of the nitric oxide reduction operon (norRVW) in *Escherichia coli*. Role of NorR and sigma54 in the nitric oxide stress response. *J. Biol. Chem.*, **278**, 10081–10086.
60. Winkler,F.K., Banner,D.W., Oefner,C., Tsernoglou,D., Brown,R.S., Heathman,S.P., Bryan,R.K., Martin,P.D., Petratos,K. and Wilson,K.S. (1993) The crystal-structure of EcoRV endonuclease and of its complexes with cognate and non-cognate DNA fragments. *EMBO J.*, **12**, 1781–1795.
61. Bujnicki,J.M., Radlinska,M. and Rychlewski,L. (2001) Polyphyletic evolution of type II restriction enzymes revisited: two independent sources of second-hand folds revealed. *Trends Biochem. Sci.*, **26**, 9–11.
62. Saravanan,M., Bujnicki,J.M., Cymerman,I.A., Rao,D.N. and Nagaraja,V. (2004) Type II restriction endonuclease R.KpnI is a member of the HNH nuclease superfamily. *Nucleic Acids Res.*, **32**, 6129–6135.
63. Pingoud,V., Sudina,A., Geyer,H., Bujnicki,J.M., Lurz,R., Luder,G., Morgan,R., Kubareva,E. and Pingoud,A. (2005) Specificity changes in the evolution of type II restriction endonucleases—a biochemical and bioinformatic analysis of restriction enzymes that recognize unrelated sequences. *J. Biol. Chem.*, **280**, 4289–4298.

Nucleic Acids Research, 2005, Vol. 33, No. 8 2733

64. Jeltsch,A., Kröger,M. and Pingoud,A. (1995) Evidence for an evolutionary relationship among type-II restriction endonucleases. *Gene*, **160**, 7–16.
65. Newman,M., Strzelecka,T., Dörner,L.F., Schildkraut,I. and Aggarwal,A.K. (1995) Structure of BamHI endonuclease bound to DNA: partial folding and unfolding on DNA binding. *Science*, **269**, 656–663.
66. Lukacs,C.M. and Aggarwal,A.K. (2001) BglIII and MunI: what a difference a base makes. *Curr. Opin. Struct. Biol.*, **11**, 14–18.
67. Dickerson,R.E. (1983) Base sequence and helix structure variation in B and A DNA. *J. Mol. Biol.*, **166**, 419–441.
68. Beveridge,D.L., Barreiro,G., Byun,K.S., Case,D.A., Cheatham,T.E.,III, Dixit,S.B., Giudice,E., Lankas,F., Lavery,R., Maddocks,J.H. *et al.* (2004) Molecular dynamics simulations of the 136 unique tetranucleotide sequences of DNA oligonucleotides. I. Research design and results on d(CpG) steps. *Biophys. J.*, **87**, 3799–3813.
69. Engler,L.E., Sapienza,P., Dörner,L.F., Kucera,R., Schildkraut,I. and Jen-Jacobson,L. (2001) The energetics of the interaction of BamHI endonuclease with its recognition site GGATCC. *J. Mol. Biol.*, **307**, 619–636.
70. Wilhelm,T. and Nikolajewa,S. (2004) A new classification scheme of the genetic code. *J. Mol. Evol.*, **59**, 598–605.
71. Taylor,J.D. and Halford,S.E. (1989) Discrimination between DNA sequences by the EcoRV restriction endonuclease. *Biochemistry*, **28**, 6198–6207.
72. Jeltsch,A., Alves,J., Wolfes,H., Maass,G. and Pingoud,A. (1994) Pausing of the restriction endonuclease EcoRI during linear diffusion on DNA. *Biochemistry*, **33**, 10215–10219.

Pattern Analysis of Gene Regulatory Rules

The first paper of Nikolajewa *et al.* provides a minimal formula representation and the exact number of hierarchically analyzing functions. We have also shown that the naturally occurring rules belong to two simple subclasses of hierarchically analyzing functions, that support the stable dynamical behavior of gene regulatory networks. In the second paper Friedel *et al.* the new data structure *Decomposition Tree* is presented. It is useful for the classification and analysis of Boolean functions.

Boolean Networks with biologically relevant rules show ordered behavior^{*}

S. Nikolajewa^a, M. Friedel^a and T. Wilhelm^{a,*}

^a*Theoretical Systems Biology, Leibniz Institute for Age Research - Fritz Lipmann Institute (former Institute of Molecular Biotechnology), Beutenbergstr. 11, Jena, D-07745, Germany*

Abstract

It was found recently that natural gene regulatory systems are governed by hierarchically canalyzing functions (HCFs), a special subclass of Boolean functions. Here we study the HCF class in detail. We present a new minimal logical expression for all HCFs. Based on this formula, we calculate the cardinality of the HCF class. Moreover, we define HCF subclasses and calculate their cardinality as well. Using the well-known critical connectivity condition $2K_{cp}(1-p) = 1$, we discuss order-chaos transitions of Boolean networks (BNs) regulated by functions of given HCF subclasses. Finally, analysing real gene regulatory rules we show that nearly all of the biologically relevant functions belong to the simplest HCF subclasses. This restriction is important for reverse engineering of transcription regulatory networks and for ensemble approach studies in systems biology. It is shown that Boolean networks with functions belonging to the biologically realized HCF subclasses show ordered behavior.

Key words: Boolean Network, canalyzing function, hierarchically canalyzing function, nested canalyzing function

1 Introduction

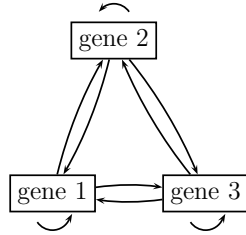
One of the outstanding problems in contemporary systems biology is the understanding of the multifariously interwoven networks underlying cellular reg-

^{*} This work has been supported by the Bundesministerium für Bildung und Forschung Grant 0312704E.

^{*} Corresponding author. Adress: IMB, Beutenbergstr. 11, Jena, D-07745, Germany; tel:+49-3641 65 6208, fax:+49-3641 65 6191

Email address: wilhelm@imb-jena.de (T. Wilhelm).

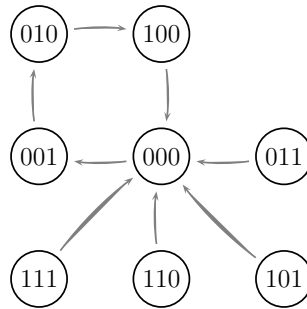
ulation. Boolean networks (BNs) (Kauffman, 1969, 1993) play a prominent role to elucidate and simulate cellular regulatory systems. In these simple models the nodes (for instance genes) are either on or off. Consider, for example, a BN representing a gene regulatory system comprising three genes, with the wiring diagram and the corresponding Boolean updating rules:



$$\begin{aligned} x_1^{t+1} &= \bar{x}_1^t \wedge x_2^t \wedge \bar{x}_3^t \\ x_2^{t+1} &= \bar{x}_1^t \wedge \bar{x}_2^t \wedge x_3^t \\ x_3^{t+1} &= \bar{x}_1^t \wedge x_2^t \wedge \bar{x}_3^t \end{aligned}$$

The BN can also be defined by a truth table or a state space flow diagram, mapping each possible input state to the corresponding output state:

x_1^t	x_2^t	x_3^t	x_1^{t+1}	x_2^{t+1}	x_3^{t+1}
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	0	0
1	1	0	0	0	0
1	1	1	0	0	0



This system has one attractor of length 4.

Up to now BNs represent the only class of dynamical models which led to non-trivial results about cellular organization on a large scale (Szallasi and Liang, 1998). Accordingly, many authors study BNs (Albert and Barabasi, 2000; Bornholdt and Sneppen, 2000; Glass and Hill, 1998; Huang, 2001; Stauffer, 1987; Lähdesmäki et al., 2003; Huang et al., 2005; Shmulevich et al., 2005), others try to evaluate in detail the inexactness due to such an abstract approach (Buchler et al., 2003; Setty et al., 2003).

It was speculated that real genetic networks do not use all Boolean rules with the same probability (Kauffman, 1993; Shmulevich et al., 2003; Gat-Viks and Shamir, 2003). Harris et al. (2002) collected the updating rules of 139 different real genes. A corresponding analysis confirmed earlier speculations: it was shown that nearly all of these rules belong to the class of canalizing functions

(CFs) (Kauffman, 1993), also denoted as canalizing functions (Shmulevich et al., 2004) or forcing functions (Stauffer, 1987). A Boolean function is canalizing if already one input alone can determine the output. The other inputs play a role only if this canalizing input takes its non-canalizing value (Kauffman, 1993). Moreover, the cardinality of the CF class was calculated (Just et al., 2004). A later analysis of Harris' data revealed that 133 of the 139 rules belong to a special subclass of CFs: to hierarchically canalizing functions (HCFs), also known as nested canalizing functions (Kauffman et al., 2003), a class first introduced some years ago by Szallasi and Liang (1998).

Section 2 deals with HCFs in general. We present the minimal logical expression for HCFs. Based on this result we calculate the number of HCFs. Moreover, we define subclasses of HCFs and calculate their cardinality as well. Based on the well-known critical connectivity formula $2K_c p(1-p) = 1$ (Derida and Pomeau, 1986), the order-chaos transitions are discussed for BNs regulated by functions of a given HCF subclass. Section 3 discusses biological applications. Analyzing Harris' data (Harris et al., 2002) we show that 128 of the 133 hierarchically canalizing gene regulatory rules belong to the two simplest HCF subclasses. It is shown that BNs with functions of these biologically relevant subclasses show ordered behavior.

2 Hierarchically Canalizing Functions (HCF)

Some years ago the idea of CFs was extended to hierarchically canalizing functions (Szallasi and Liang, 1998). In HCFs all inputs are canalizing in a hierarchical manner: if the first input takes on its non-canalizing value, a second input is canalizing for the remaining states. If the second input takes on the non-canalizing value, a third input is canalizing, etc. HCFs represent an important subclass of CFs. It was shown that HCFs enhance order even more than simple CFs (Szallasi and Liang, 1998; Kauffman et al., 2004). Studying the transcriptional regulation of real genes, it was found that just 6 of 139 are not HCFs (Kauffman et al., 2003). Thus, in genetic regulatory networks there seems to be a very strong tendency towards HCFs. In the following we give a formal definition of HCFs and present the corresponding minimal logical formula representation. Moreover, we calculate the exact number of hierarchically canalizing functions (depending on the number of inputs), define HCF subclasses and calculate their cardinality as well.

Let k denote the number of inputs of a Boolean function $f(x) = f(x_1, \dots, x_k)$ and \mathcal{B}^k the set of all Boolean functions on k variables. The symbol σ stands for a possible negation of Boolean variables, so x^σ can be x or \bar{x} . The input x_i is called essential if $f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_k) \neq f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_k)$.

Definition 1 (Hierarchically Canalizing Function) *Let f be a canaliz-*

ing function with canalizing input x_i and canalizing input value a_i . If the function $g(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_k) = f(x_1, \dots, x_{i-1}, \bar{a}_i, x_{i+1}, \dots, x_k)$ on $k - 1$ inputs is canalizing, then the function $f(x)$ is a 2-times canalizing function. A Boolean function f on k essential inputs is called hierarchically canalizing if $f(x)$ is k -times canalizing with the canalizing inputs x_1, \dots, x_k and the canalizing input values a_1, \dots, a_k .

2.1 Minimal formula representation of a Hierarchically Canalizing Function

A Boolean function $f(x)$ on k essential inputs is a hierarchically canalizing function if and only if $f(x)$ can be written with k different inputs and $k - 1$ binary operations \wedge or \vee , where the operation priority is ordered from left to right ($\odot \in \{\wedge, \vee\}$):

$$f_i(x_1, x_2, \dots, x_k) = x_{i_1}^{\sigma_{i_1}} \odot (x_{i_2}^{\sigma_{i_2}} \odot (\dots \odot (x_{i_{k-1}}^{\sigma_{i_{k-1}}} \odot x_{i_k}^{\sigma_{i_k}})) \dots). \quad (1)$$

This can constructively be proven as follows: let $f(x_1, x_2, \dots, x_k)$ be a hierarchically canalizing function with the first canalizing input x_1 , then there exists a function $f_2 = g(x_2, \dots, x_k)$ such that f can be written as $f = x_1^{\sigma_1} \odot f_2(x_2, x_3, \dots, x_k)$, $\odot \in \{\wedge, \vee\}$. From the HCF definition it follows that function $f_2 \in \mathcal{B}^{k-1}$ is canalizing again. Repeating this procedure k times leads to the minimal logical formula (1).

Example 2 *TGF- β is a gene regulated by a HCF. The transforming growth factor- β controls growth differentiation and apoptosis of cells. It is regulated by five transcription factors (Harris et al., 2002): the negative regulator SnoN, the receptor proteons Smad2/4 and Smad3/4, histone deacetylases HDAC, and the nuclear transcriptional corepressor N-CoR. The gene updating rule can be defined by a binary string (11111110000000111111111101111) or the minimal logical formula $TGF^{(t+1)} = Smad2/4^{(t)} \wedge (not\ SnoN^{(t)} \vee Smad3/4^{(t)} \vee not\ N - CoR^{(t)} \vee not\ HDAC/Sin3^{(t)})$.*

2.2 Number of Hierarchically Canalizing Functions

Szallasi and Liang (1998) numerically calculated the number of HCFs up to $k = 5$ inputs. Here we present the exact formula for arbitrary k . The number of hierarchically canalizing functions is

$$N = 2 + 2k + \sum_{i=2}^k \binom{k}{i} a_i 2^{i+1}, \quad (2)$$

$$\begin{aligned}
 &\text{with } a_2 = 1 \\
 &a_3 = 1 + \binom{3}{2}a_2 \\
 &a_4 = 1 + \binom{4}{2}a_2 + \binom{4}{3}a_3 \\
 &\dots \\
 &a_k = 1 + \sum_{j=2}^{k-1} \binom{k}{j}a_j.
 \end{aligned}$$

This number is deduced by counting all different HCF representations (1) for a given number of inputs k . For each k two constant functions (0,0,...,0 and 1,1,...,1) and $2k$ functions that depend on just one input exist (first two summands in (2)). Each summand in (2) counts the number of functions with i essential inputs, a_i denotes the number of different parenthesis patterns (Table 1), separating \wedge and \vee - operators. For each parenthesis pattern, there are two possibilities of operator assignments: starting with \wedge or with \vee .

i	parenthesis pattern	operator assignments		a_i
2	$(x_1 \odot x_2)$	$x_1 \vee x_2; x_1 \wedge x_2$	1	$a_2 = 1$
3	$x_1 \odot x_2 \odot x_3$	$x_1 \vee x_2 \vee x_3; x_1 \wedge x_2 \wedge x_3$	1	$a_3 = 1 + \binom{3}{2}a_2$
	$(x_1 \odot x_2) \odot x_3$	$(x_1 \vee x_2) \wedge x_3; (x_1 \wedge x_2) \vee x_3$		
	$x_1 \odot (x_2 \odot x_3)$	$x_1 \vee (x_2 \wedge x_3); x_1 \wedge (x_2 \vee x_3)$	$\binom{3}{2}$	
	$x_2 \odot (x_1 \odot x_3)$	$x_2 \vee (x_1 \wedge x_3); x_2 \wedge (x_1 \vee x_3)$		
4	$x_1 \odot x_2 \odot x_3 \odot x_4$...	1	$a_4 = 1 + \binom{4}{2}a_2 + \binom{4}{3}a_3$
	$(x_1 \odot x_2) \odot x_3 \odot x_4$...	$\binom{4}{2}$	
	
	$x_1 \odot (x_2 \odot (x_3 \odot x_4))$...	$\binom{4}{3}$	
i	$a_i = 1 + \sum_{j=2}^{i-1} \binom{i}{j}a_j$

Table 1

Number of different possibilities to set parentheses (a_i) in minimal logical expressions of HCFs with i essential inputs.

The number of different σ patterns (negations of inputs) is 2^i . Therefore, the number of different HCFs on i essential inputs is $a_i * 2 * 2^i$.

2.3 Subclasses of Hierarchically Canalizing Functions

The definition of HCF subclasses S_l^k is based on the minimal formula representation of a HCF (1), containing k essential inputs and $k - 1$ logical operations in a fixed order. These operations can be encoded by a binary number of length $k - 1$, where 1 and 0 correspond to OR and AND, respectively. For example, the operations of function $\bar{x}_1 \wedge (x_2 \vee (\bar{x}_3 \wedge x_4))$ are encoded by 010 (or decimal $l = 3$), thus this function belongs to the class S_3^4 .

Definition 3 (S_l^k : subclasses of HCFs) Let l_b be the binary representation of the decimal number l : $l_b = l_0l_1 \dots l_{k-2}$ codes for the operations order

in the minimal logical HCF formula (1). All hierarchically analyzing functions $f_i(x_1, \dots, x_k) = x_{i_1}^{\sigma_{i_1}} \odot_1 (x_{i_2}^{\sigma_{i_2}} \odot_2 (\dots (x_{i_{k-1}}^{\sigma_{i_{k-1}}} \odot_{k-1} x_{i_k}^{\sigma_{i_k}}) \dots))$, where \odot_j is \wedge , if $l_{j-1} = 0$, or \odot_j is \vee , if $l_{j-1} = 1$ with the same operation order $l_b = l_0 l_1 \dots l_{k-2}$ belong to the S_l^k class.

Based on the different operator patterns the class of HCFs on k inputs can be divided into 2^{k-1} subclasses: $S_0^k, S_1^k, \dots, S_{2^{k-1}-1}^k$. The class S_0^k contains all rules $x_1^{\sigma_1} \wedge x_2^{\sigma_2} \wedge \dots \wedge x_k^{\sigma_k}$. In class S_1^k the last operation is \vee and all other operations are \wedge , for instance $x_k^{\sigma_k} \wedge x_{k-1}^{\sigma_{k-1}} \wedge \dots \wedge (x_2^{\sigma_2} \vee x_1^{\sigma_1})$. Functions with only \vee belong to the class $S_{2^{k-1}-1}^k$.

Appendix A shows an example of all subclasses of HCFs on $k = 4$ inputs. In appendix B a procedure to calculate the cardinality of the HCF subclasses S_l^k is given.

2.4 Order and chaos of Boolean Networks with S_l^k functions (S_l^k networks)

The phase transition between the ordered and chaotic regimes can be defined with help of the average sensitivity

$$\bar{S} = 2Kp(1 - p), \quad (3)$$

where K is the average connectivity of the BN and p is the probability of choosing 1 rather than 0 for the transition function output values (Derrida and Pomeau, 1986; Shmulevich et al., 2005). For $\bar{S} < 1$ the network is in the ordered regime, for $\bar{S} > 1$ the network shows chaotic behavior. The number of ones of a given S_l^k function is $2l + 1$ (proof is given in appendix C). Therefore, $p = \frac{2l+1}{2^k}$. A BN where each node is regulated by a S_l^k function (S_l^k network) has the average sensitivity:

$$\bar{S}(k, l) = 2k \frac{2l+1}{2^k} \left(1 - \frac{2l+1}{2^k}\right). \quad (4)$$

Figure 1a shows $\bar{S}(k, l)$ and figure 1b $\bar{S}(k)$ for some given l .

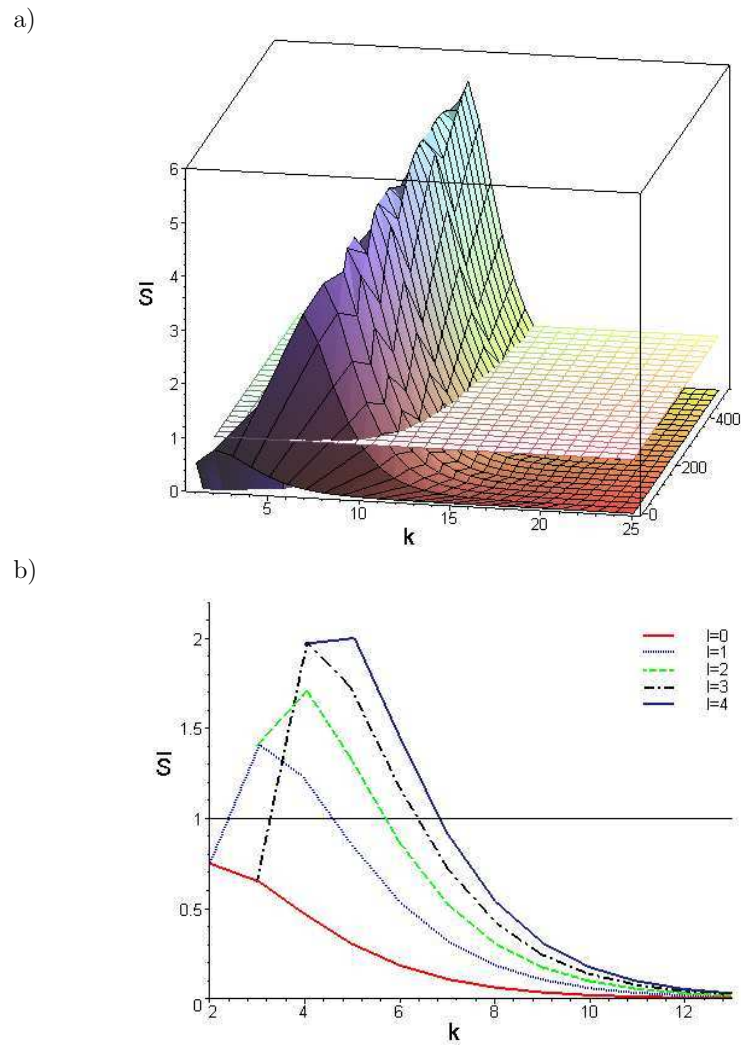


Fig. 1. Sensitivity of S_l^k networks. a) $\bar{S}(k, l)$. The cut between $\bar{S}(k, l)$ and $\bar{S} = 1$ defines the order - chaos transition. b) $\bar{S}(k)$ for $l = 0, 1, 2, 3, 4$

S_0^k networks are always stable, because $\bar{S} < 1$ for arbitrary k . Generally, for S_l^k networks intervals of chaotic behavior exist (Table 2). Interestingly, for many l (for instance $l = 1, 3$) ordered behavior is shown for small and large k and chaotic behavior for medium k .

l	chaos	order
0 and $2^{k-1} - 1$	—	$k \geq 2$
1 and $2^{k-1} - 2$	$3 \leq k \leq 4$	$k = 2, k \geq 5$
2 and $2^{k-1} - 3$	$3 \leq k \leq 5$	$k \geq 6$
3 and $2^{k-1} - 4$	$4 \leq k \leq 6$	$k = 3, k \geq 7$
...		
$2^{k-2} - 2$ and $2^{k-2} + 1$	$k \geq 4$	$k = 3$
$2^{k-2} - 1$ and 2^{k-2}	$k \geq 3$	—

Table 2
Ordered and chaotic regimes for S_l^k networks.

3 Biological Importance of Hierarchically Canalizing Functions

Analyzing natural gene regulatory rules it was first found that all of them belong to the class of canalizing functions (Harris et al., 2002). Later it was shown that 133 of the 139 analyzed functions are also hierarchically canalizing (Kauffman et al., 2003). We translated all these 133 HCFs into the corresponding minimal logical expressions and found that nearly all belong to two special subclasses: S_0^k 66, 39% and S_1^k 29, 41%.

Based on the average sensitivity (4) it can be shown that BNs, which are made up of $2/3 S_0^k$ and $1/3 S_1^k$ rules, show always ordered behavior. The maximum sensitivity (worst case), deduced from (4), for S_0^k and S_1^k is $\bar{S}^{max}(k = 2, 0) = \frac{3}{4}$ and $\bar{S}^{max}(k = 3, 1) = \frac{45}{32}$, respectively. The biological realistic combination of them leads to $\frac{2}{3} * \frac{3}{4} + \frac{1}{3} * \frac{45}{32} = \frac{31}{32} < 1$. Thus, a BN with this fraction of functions is stable even in the most critical case of two inputs for all S_0^k functions and three inputs for all S_1^k functions.

HCFs can most easily be realized by gene regulatory systems. Here only one single operator (\wedge or \vee) per transcription factor exists, no additional intermediate TFs are needed. In contrast, non-canalizing functions seem to be unfavorable for the design of gene regulatory systems. Consider, for instance, the simplest non-canalizing function x_1 XOR x_2 (Table 3): the result is false, if both inputs have the same value. The implementation of XOR needs intermediate transcription factors, which implies synchronization problems (Buchler et al., 2003). The XOR-gate cannot be implemented by the minimal formula (1).

The specific \wedge, \vee pattern in the minimal expression of a given HCF provides interesting information about the output control. In the simplest cases all operations are the same. If the minimal logical expression contains only AND (S_0^k), then all (transcription) factors have the same importance and cannot be replaced by other factors. In the 'only OR case' ($S_{2^{k-1}-1}^k$) the factors are

inputs		1st step		2nd step
x_1	x_2	$x_1 \wedge \bar{x}_2$	$\bar{x}_1 \wedge x_2$	$(x_1 \wedge \bar{x}_2) \vee (\bar{x}_1 \wedge x_2) = x_1 \text{ XOR } x_2$
0	0	0	0	0
0	1	0	1	1
1	0	1	0	1
1	1	0	0	0

Table 3
Implementation of XOR based on AND and OR gates.

independent from each other, each factor alone can determine the output. Generally, the number of ANDs and ORs and their position in the minimal expression (1) provide information about (gene) regulation: the number of \vee and \wedge corresponds to the number and size of factor groups, respectively, that determine the output independently of each other (four such groups in example 2).

4 Discussion

We presented a minimal logical expression for Boolean functions of the biologically important HCF class. Based on this result the exact number of HCFs was calculated. Moreover we defined meaningful subclasses of HCFs, calculated their cardinality as well, and discussed the stability of S_1^k networks. Analyzing biological data on gene regulation we found that nearly all genes are regulated by functions of the two simplest subclasses of HCFs: S_0^k and S_1^k . This is important for corresponding ensemble approach studies (Kauffman, 2004a,b; Kauffman et al., 2004) and for reverse engineering of gene regulatory networks (Akutsu et al., 2000, 1999; D’haeseleer et al., 2000; Yeung et al., 2003). The smaller the number of possible functions, the fewer data is needed for reverse engineering. Table 4 demonstrates that S_0^k and S_1^k comprise just a small subset of all HCFs.

We have also shown that Boolean networks with functions belonging to the biologically observed subclasses (Harris et al., 2002) are always stable. This corresponds to a resently found numerical result (Rämö et al. , 2005). The minimal logical expression of HCFs has different additional advantages for the analysis of gene regulation and beyond. Membership in the HCF class can be proven in polynomial time for any Boolean function, thus for these functions the famous minimization problem (important for chip design, for instance) is solved. For genes regulated by HCFs the minimal expression helps to quantify the importance of the different transcription factors. Moreover, groups of TFs that operate together can easily be identified. These combined

k	S_0^k	S_1^k	HCF	CF	$Total$
1	0	0	4	4	4
2	4	4	14	14	16
3	8	24	96	120	256
4	16	96	1050	3514	65536
5	32	320	15036	1292276	4294967296

Table 4

The number of Boolean functions on k inputs in different biologically meaningful subclasses in comparison to the total number of Boolean functions (2^{2^k}).

operations could be proven by clustering of corresponding mRNA expression data.

Acknowledgment. We thank S. Harris for providing us the data on gene regulation.

APPENDIX A. The 8 subclasses of HCFs with $k = 4$. For the sake of simplicity possible negations are neglected. Negations only shift the true points and do therefore not alter the number of the true/false points. The second line of the table shows the binary representation l_b of the corresponding operator pattern, given in the third line.

n	Class				S_0^k $x_1(x_2(x_3x_4))$	S_1^k $x_1(x_2(x_3 \vee x_4))$	S_2^k 010 $x_1(x_2 \vee (x_3x_4))$	S_3^k 011 $x_1(x_2 \vee (x_3 \vee x_4))$	S_4^k 100 $x_1 \vee (x_2(x_3x_4))$	S_5^k 101 $x_1 \vee (x_2(x_3 \vee x_4))$	S_6^k 110 $x_1 \vee (x_2 \vee (x_3x_4))$	S_7^k 111 $x_1 \vee (x_2 \vee (x_3 \vee x_4))$
	x_1	x_2	x_3	x_4								
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0	0	1
2	0	0	1	0	0	0	0	0	0	0	0	1
3	0	0	1	1	0	0	0	0	0	0	1	1
4	0	1	0	0	0	0	0	0	0	0	1	1
5	0	1	0	1	0	0	0	0	0	1	1	1
6	0	1	1	0	0	0	0	0	0	1	1	1
7	0	1	1	1	0	0	0	0	1	1	1	1
8	1	0	0	0	0	0	0	0	1	1	1	1
9	1	0	0	1	0	0	0	1	1	1	1	1
10	1	0	1	0	0	0	0	1	1	1	1	1
11	1	0	1	1	0	0	1	1	1	1	1	1
12	1	1	0	0	0	0	1	1	1	1	1	1
13	1	1	0	1	0	1	1	1	1	1	1	1
14	1	1	1	0	0	1	1	1	1	1	1	1
15	1	1	1	1	1	1	1	1	1	1	1	1
Number of ones					1	3	5	7	9	11	13	15

APPENDIX B. The number of functions in HCF subclasses S_l^k (k essential inputs).

The number of different functions in the S_0^k (and $S_{2^{k-1}-1}^k$) class is easily counted: each negation just shifts the only 1 (or 0). Because there are 2^k negations (each input can be negated or not), there are 2^k different S_0^k (and $S_{2^{k-1}-1}^k$) functions. There are $\binom{k}{2}$ possibilities to choose the two inputs which are combined by \vee . Therefore, the cardinality of the S_1^k class is $\binom{k}{2}2^k$. The determination of the number of exchange possibilities for an arbitrary HCF subclass is based on the corresponding operator pattern given by the minimal logical expression. The pattern is divided into blocks of equal operations (the order of operations is fixed). Then the number of exchanges leading to different functions is calculated for each block. Finally, these numbers are multiplied. The procedure to calculate the number of different functions of a given S_l^k class is:

Input: k is the number of essential inputs; $l_b = l_0l_1 \dots l_{k-2}$, ($l_i \in \{0, 1\}$) denotes the subclass.

Output: N is the number of elements in the S_l^k class.

```

PROCEDURE N( $k, l_b$ );
   $N := 2^k$ ;
   $\hat{k} := k$ ;
   $n_{op} := 2$ ;
   $i := k - 3$ ;
  repeat
    if  $l_{i+1} = l_i$  then  $n_{op} := n_{op} + 1$ 
    else
       $N := N * \binom{\hat{k}}{n_{op}}$ ;
       $\hat{k} := \hat{k} - n_{op}$ ;
       $n_{op} := 1$ ;
     $i := i - 1$ ;
  until ( $i < 0$ )
  return  $N$ ;
END;
```

APPENDIX C. The number of true points of any function of the S_l^k class is $2l + 1$.

The binary representation $l_b = l_0 l_1 \dots l_{k-2}$ of the decimal number $l = l_0 2^{k-2} + l_1 2^{k-3} + \dots + l_{k-2} 2^0$ codes for the order of the $k - 1$ operations. If the first bit $l_0 = 1$ (first operator is \vee), the first input is canalyzing to 1, which implies $\frac{2^k}{2}$ ones in the truth table. Generally, each $l_i = 1$ yields $\frac{2^k}{2^{i+1}}$ ones in the truth table. Because the last input is always canalyzing to 1 we have one additional 1 in the truth table output. Summing the total number of ones leads to $l_0 \frac{2^k}{2} + l_1 \frac{2^k}{2^2} + \dots + l_{k-2} \frac{2^k}{2^{k-1}} + 1 = 2l + 1$.

References

- Albert, R., Barabasi, A.-L., 2000. Dynamics of complex systems: Scaling laws for the period of Boolean networks. *Phys. Rev. Lett.* 84, pp. 5660-5663.
- Akutsu, T., Miyano, S., Kuhara, S., 2000. Algorithms for inferring qualitative models of biological networks *Pacific Symposium on Biocomputing* 5, 293-304.
- Akutsu, T., Miyano, S., Kuhara, S., 1999. Identification of genetic networks from a small number of gene expression patterns under the Boolean network model. *Pacific Symposium on Biocomputing* 4, pp. 17-28.
- Bornholdt, S., Sneppen, K., 2000. Robustness as an Evolutionary Principle. *Proc. R. Soc. London B.* 267, pp. 2281-2286.
- Buchler, N.E., Gerland, U., Hwa, T., 2003. On schemes of combinatorial transcription logic. *Proc. Natl. Acad. Sci.* 100, pp. 5136-5141.
- Derrida, B., Pomeau, Y., 1986. Random networks of automata - a simple annealed approximation. *Europhysics letters* 1(2), pp. 45-49.
- D'haeseleer P., Liang S., Somogyi R., 2000. Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics.* 16(8), pp. 707-726.
- Gat-Viks, I., Shamir, R., 2003. Chain functions and scoring functions in genetic networks. *Bioinformatics* 19(1), pp. 108-117.
- Glass, L., Hill, C., 1998. Ordered and Disordered Dynamics in Random Networks. *Europhys. Letts.* 41, pp. 599-604.
- Harris, S., Sawhill, B., Wuensche, A., Kauffman, S., 2002. A Model of Transcriptional Regulatory Networks Based on Biases in the Observed Regulation Rules. *Complexity* 7(4), pp. 23-40.
- Huang, S., 2001. Genomics, complexity and drug discovery: insights from Boolean network models of cellular regulation. *Pharmacogenomics* 2(2), pp. 203-221.
- Huang, S., Eichler, G., Bar-Yam, Y., Ingber, D.E., 2005. Cell fates as high-dimensional attractor states of a complex gene regulatory network. *Phys. Rev. Lett.* 94, pp. 128701.
- Just, W., Shmulevich, I., Konvalina, J., 2004. The number and probability of canalizing functions. *Physica D.* 197, pp. 211-221.
- Kauffman, S., 1969. Metabolic Stability and Epigenesis in Randomly Constructed Genetic Nets, *J. Theor. Biol.* 22, pp. 437-467.
- Kauffman, S., 1993. *The Origins of Order: Self-Organization and Selection in Evolution.* Oxford University Press, New York.
- Kauffman, S., 2004. A proposal for using the ensemble approach to understand genetic regulatory networks. *J. Theor. Biol.* 230(4), pp. 581-590.
- Kauffman, S., 2004. The ensemble approach to understand genetic regulatory networks. *Physica A,* 340, pp. 733-740.
- Kauffman, S., Peterson, C., Samuelsson, B., and Troein, C., 2004. Genetic networks with canalizing Boolean rules are always stable. *Proc. Natl. Acad. Sci.* 101, pp. 17102-17107

- Kauffman,S., Peterson,C., Samuelsson,B., Troein,C., 2003. Random Boolean Network Models and the Yeast Transcriptional Network. *Proc. Natl. Acad. Sci.* 100, pp. 14796-14799.
- Lähdesmäki,H., Shmulevich,I., Yli-Harja,O., 2003. On learning gene regulatory networks under the Boolean network model. *Machine Learning* 52, pp. 147-167.
- Rämö, P., Kesseli,J., and Yli-Harja,O. 2005. Stability of functions in Boolean models of gene regulatory networks. *Chaos* 15, 034101.
- Setty,Y., Mayo,A.E., Surette,M.G., Alon,U., 2003. Detailed map of a cis-regulatory input function. *Proc. Natl. Acad. Sci.* 100, pp. 7702-7707.
- Shmulevich,I., Lähdesmäki,H., Dougherty,E.R., Astola,J., Zhang,W., 2003. The role of certain Post classes in Boolean network models of genetic networks. *Proc. Natl. Acad. Sci. USA* 100(19), pp. 10734-10739.
- Shmulevich,I., Lähdesmäki,H., and Egiazarian,K., 2004. Spectral Methods for Testing Membership in Certain Post Classes and the Class of Forcing Functions. *IEEE Signal Processing Let.* 11(2), pp. 289-292.
- Shmulevich, I., Kauffman,S.A., Aldana,M., 2005. Eukaryotic cells are dynamically ordered or critical but not chaotic. *Proc. Natl. Acad. Sci.* 102, pp. 13439-13444.
- Stauffer,D., 1987. On forcing functions in Kauffman random boolean networks. *J. Stat. Phys.* 46(3-4), pp. 789-794.
- Szallasi,Z., Liang,S., 1998. Modeling the Normal and Neoplastic Cell Cycle With Realistic Boolean Genetic Networks: Their Application for Understanding Carcinogenesis and Assessing Therapeutic Strategies. in: *Pacific Symp. on Biocomputing* 3, pp. 66-76.
- Yeung,M.K.S., Tegner,J., Collins,J.J., 2003. Reverse engineering gene networks using singular value decomposition and robust regression. *PNAS* 99, pp. 6163-6168.

The Decomposition Tree for Analyses of Boolean Functions^{*}

M. Friedel, S. L. Nikolajewa, T. Wilhelm^{*}

*Leibniz Institute for Age Research - Fritz Lipmann Institute, Beutenbergstr. 11,
D-07745 Jena, Germany*

Abstract

We present a new data structure for analyses of Boolean functions, called Decomposition Tree (DT), and demonstrate different applications. In each node of the DT appropriate bit string decomposition fragments are combined by a logical operator. The DT has 2^n nodes in worst case, what implies exponential complexity for problems where the whole tree has to be considered. However, it is important to note that many problems are simpler. We show that these can be handled in an efficient way using the DT. Nevertheless, many problems are of exponential complexity and cannot be made simpler, for instance the calculation of prime implicants. Using our general DT structure, for this important task we present the fastest known exact algorithm which we are aware of (lower time complexity than the Quine-McCluskey algorithm).

Key words: Boolean functions, Decomposition Tree, monotonic function, linear function, canalyzing function, symmetry, prime implicant, logic minimization
PACS: 02.10.Ab

1 Introduction

Boolean functions (BFs) have widespread applications in nearly all fields of science and engineering. The Reduced Ordered Binary Decision Diagram

^{*} Supported by Grant 0312704E of the Bundesministerium für Bildung und Forschung

^{*} Corresponding author.

Email addresses: maikfr@fli-leibniz.de (M. Friedel),
sweta@fli-leibniz.de (S. L. Nikolajewa), wilhelm@fli-leibniz.de (T. Wilhelm).

(ROBDD) is probably the most powerful data structures known so far for the manipulation of large logic functions [4]. It provides a compact representation of Boolean expressions, and there are efficient algorithms for performing all kinds of logical operations on ROBDDs [1]. However, the ROBDD construction itself is based on the equivalence of bit-strings.

In this paper we present a new data structure for BFs, called Decomposition Tree (DT). This tree provides a unified approach to tackle many different BF problems. We take as input format the truth table, represented by a bit string of length 2^k , for a given function with k variables. In all our calculations appropriate bit string decomposition fragments are combined by a logical operator (in each node of the DT). The application of different operators allows us to classify a given function into different subclasses. For many problems only functions of a particular subclass are needed. In molecular biology, for instance, gene regulatory networks are simulated with analyzing [15] and hierarchically analyzing Boolean functions [26]. Such functions have also been used to study such diverse problems as decision structures in social systems [17], the convergence behavior of nonlinear filters [27], or artificial life [16]. Monotonic functions play a special role in game theory, computational learning, harmonic analysis and signal processing. Nonlinear functions are essential for cryptographic transformations [12,23]. Functions with unate properties are used in the design of conventional cryptosystems [12] and functions with special symmetry characteristics are important for circuit restructuring [13].

Each Boolean function can be represented by its disjunctive normal form (DNF). A lot of BF research is devoted to minimal DNFs [9,28–31]. The generation of prime implicants (PIs) of a given function is an important first step to calculate its minimal DNF. Early interest in PIs [22] was mainly inspired by this problem. Meanwhile different other applications have been found. PIs are used for alternative representations of Boolean expressions in various problems of artificial intelligence [24], to implement Assumption-Based Truth Maintenance, to characterize diagnoses, to compile formulas for Transcranial Magnetic Stimulation and to implement circumscription [14,16]. PIs play a role in expert system development to find all irredundant rules from a given rule system and in Electronic Design Automation [11]. We show that one can simply generate all PIs of a given BF by using our Decomposition Tree and the AND-operator for the manipulation of the appropriate bit-strings.

In the first part of the paper (section 2) we introduce the Decomposition Tree. In the following application part (section 3) we first demonstrate our approach to tackle the subclass classification problem (section 3.1). In section 3.2 we present an efficient recursive algorithm to compute prime implicants. It is shown that our PI algorithm has a lower time complexity than the well-known algorithm of Quine and McCluskey [10,19,22] which also uses the truth

table input format. In Section 3.3 we demonstrate how the DT can be used to construct the ROBDD of a given BF.

2 Decomposition Tree

Let $f : \{0, 1\}^k \rightarrow \{0, 1\}$ be a Boolean function on k variables. The Decomposition Tree is based on the Decomposition Set.

Definition 1 (D_i -decomposition) D_i -decomposition of function f in input x_i is a segmentation of f into two functions f_0^i and f_1^i , which are defined by the positive and negative values of the input x_i :

$$D_i : \begin{aligned} f_0^i &= f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_k) \\ f_1^i &= f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_k) \end{aligned} \quad (1)$$

The bit string representations of f_0^i, f_1^i are called decomposition fragments of the D_i -decomposition.

The previous definition can be generalized to decompositions in more inputs.

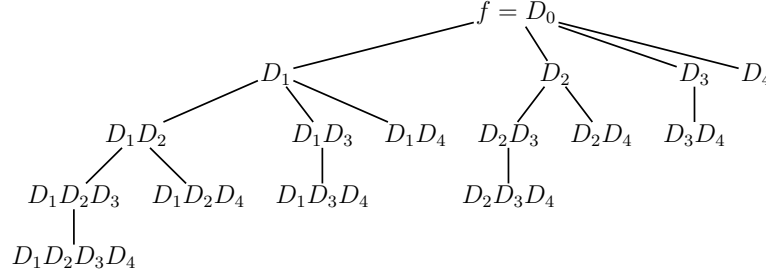
Definition 2 ($D_i D_j$ -decomposition) Given the D_i - and D_j -decompositions, $i < j \in \{1, \dots, k\}$, the $D_i D_j$ -decomposition is a combination of the D_i and D_j decompositions:

$$D_i D_j : \begin{aligned} f_{00}^{ij} &= f(x_1, \dots, \overset{i}{0}, \dots, \overset{j}{0}, \dots, x_k) \\ f_{01}^{ij} &= f(x_1, \dots, \overset{i}{0}, \dots, \overset{j}{1}, \dots, x_k) \\ f_{10}^{ij} &= f(x_1, \dots, \overset{i}{1}, \dots, \overset{j}{0}, \dots, x_k) \\ f_{11}^{ij} &= f(x_1, \dots, \overset{i}{1}, \dots, \overset{j}{1}, \dots, x_k) \end{aligned} \quad (2)$$

The decomposition can be extended to an arbitrary input combination of size $l \leq k : D_{i_1} D_{i_2} \dots D_{i_l}$, $i_1 < i_2 < \dots < i_l \in \{1, \dots, k\}$. It has 2^l bit string fragments of length 2^{k-l} . The Decomposition Set D contains 2^k possible decompositions: $\{D_0, D_1, D_2, \dots, D_k, D_1 D_2, \dots, D_1 D_2 D_3 \dots D_k\}$, where $D_0 \equiv f$.

Definition 3 (Decomposition Tree) The Decomposition Tree (DT) is an ordered tree of all possible decompositions from the set D , where duplicate decompositions are eliminated. The root D_0 of the DT is the function f .

Example 4 General Decomposition Tree for $k = 4$.



To detect a special pattern in a given Boolean function (e.g. membership in the subclass of monotonic functions or prime implicants) we combine all decomposition fragments with a special Boolean operation. The operator can be any logical operation, for instance AND, OR, XOR etc. The result of applying this operator over the decomposition fragments is a Boolean function, which we call *operator-combination* (\odot -combination).

Definition 5 (\odot -combination) *Without loss of generality, given a decomposition $D_1D_2 \dots D_l$ of function f . The \odot -combination is a Boolean function $g : \{0, 1\}^{k-l} \rightarrow \{0, 1\}$ defined by applying the \odot -operator over the decomposition fragments:*

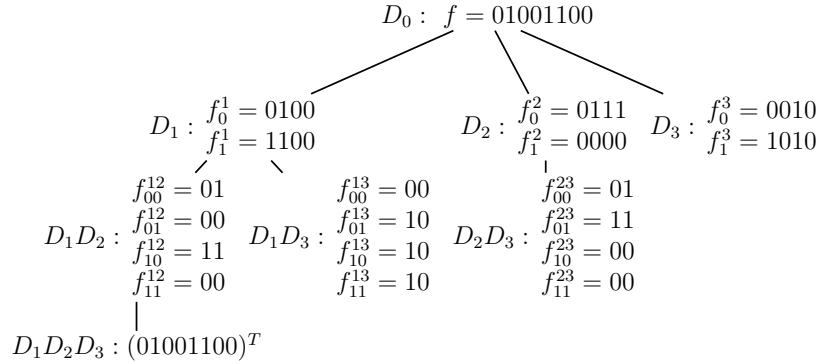
$$g(x_{l+1}, \dots, x_k) = \underbrace{f_{00\dots 0}^{12\dots l} \odot f_{00\dots 1}^{12\dots l} \odot \dots \odot f_{11\dots 1}^{12\dots l}}_{2^l}. \quad (3)$$

This can also be written as:

$$\begin{aligned} & f(0, 0, \dots, 0, x_{l+1}, \dots, x_k) \\ & \odot f(0, 0, \dots, 1, x_{l+1}, \dots, x_k) \\ & \dots \\ & \odot f(1, 1, \dots, 1, x_{l+1}, \dots, x_k) \\ & \hline & = g(x_{l+1}, \dots, x_k) \end{aligned} \quad (4)$$

We use as input format of a function the truth table of length 2^k for k variables.

Example 6 *We decompose the function $x_1\bar{x}_2\vee\bar{x}_2x_3$ with the bit string 01001100 (truth table). The decomposition tree is:*



Different properties of this function can simply be detected with the decomposition tree: the function has two positive unates in x_1 and x_3 , because fragments $f_0^1 \leq f_1^1$ and $f_0^3 \leq f_1^3$. Moreover, fragment $f_1^2 \equiv 0$, thus the function is canalizing (forcing) from 1 to 0 in input x_2 , what means that it has the special representation $f = \bar{x}_2 \wedge h(x_1, x_3)$, with $h(x_1, x_3) = x_1 \vee x_3$.

Generally, depending on the problem (the pattern which is searched for in a given BF), one mostly does not need to consider the whole tree (2^n nodes). For instance, for prime implicant calculation the tree can be cut if the \wedge -combination gives the constant function $g = 0$ (see 3.2). Even more, each node of the DT can always be calculated independently of the other nodes. It follows that many problems can be solved in polynomial time. For instance, the special class $\tilde{x}_1\tilde{x}_3$ of implicants of a given BF with $k = 4$ can be detected by only considering the node D_2D_4 (see 3.2). Another example are quadratic Boolean functions where only D_iD_j nodes have to be considered.

3 Applications

3.1 Classification of Boolean Functions

There are five characteristic classes of BFs [9]: 0(1)-preserving, self-dual, monotonic and linear functions. The first three ones can simply be detected from the truth table. However, it is more difficult to decide whether a given BF belongs to one of the last two classes. In the following we show how the classification problem can be solved with the DT for these two and other classes of BFs.

3.1.1 Monotonic Boolean function

Definition 7 Let $a = (a_1, \dots, a_k)$ and $b = (b_1, \dots, b_k)$ be different k -element binary vectors. One says that a precedes b , denoted as $a \prec b$, if $a_i \leq b_i$ for $1 \leq i \leq k$. A Boolean function $f(x_1, \dots, x_k)$ is called monotonic if for any two vectors a and b such that $a \prec b$, the relation $f(a) \leq f(b)$ holds.

Detection: If the first and last decomposition fragment of each node in the DT combined by the \leq -operator always gives the 1-constant function, than f is said to be monotonic.

Application: Monotonic functions are used in game theory, computational learning theory, harmonic analysis, and signal processing [7,18,27,33]. This is one of the characteristic classes [9].

3.1.2 Linear Boolean function

Definition 8 The Boolean function on k inputs is said to be linear, if it can be represented as $f(x_1, x_2, \dots, x_k) = a_0 \oplus a_1 x_1 \oplus \dots \oplus a_k x_k$, where $a_i \in \{0, 1\}$.

Detection: If the decomposition fragments of the nodes in the first level of the DT combined by the \oplus -operator always give constant functions, then f is said to be a linear function: If the \oplus -combination of D_i is the 0-constant function, then $f(x_1, \dots, 0, \dots, x_k) = f(x_1, \dots, 1, \dots, x_k)$ (tautology in x_i) and therefore coefficient $a_i = 0$. If the \oplus -combination of D_i is the 1-constant function, then $f(x_1, \dots, 0, \dots, x_k) = \bar{f}(x_1, \dots, 1, \dots, x_k)$ and can be written as $f(x_1, \dots, x_k) = x_i \oplus g(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_k)$, because $\bar{x}_i f(x_1, \dots, 1, \dots, x_k) \wedge x_i \bar{f}(x_1, \dots, 1, \dots, x_k) = x_i \oplus f(x_1, \dots, 1, \dots, x_k)$.

Application: Linear functions are needed in cryptography for nonlinearity criteria for cryptographic transformations [12,23]. This is one of the characteristic classes [9].

3.1.3 Positive (negative) unate

Definition 9 A positive unate in x_i function is a monotonic increasing function of x_i . A negative unate in x_i function is a monotonic decreasing function of x_i . Unate (positive or negative) is a function which is unate in all variables.

Detection: If the \leq (\geq)- combination of D_i is 1-constant, then x_i is positive (negative) unate. If this holds for all i , then f is unate. The function of example 6 is unate in all variables.

Application: If f is positive unate in x_i , then $f(x) = x_i f(x_1, \dots, 1, \dots, x_k) \vee f(x_1, \dots, 0, \dots, x_k)$. This is used by the Unate Recursive Paradigm and it is important for the design of conventional cryptosystems [5,6,20,25].

3.1.4 Symmetry

Definition 10 *If a Boolean function does not change when any possible pair of variables is exchanged, it is said to be a totally symmetric function. If a function does not change when any possible pair of a subset of inputs is exchanged, it is said to be a partially symmetric function.*

Detection: A function is totally symmetric if all decompositions of the first level are the same. For example, the totally symmetric function 01101000 ($x_1 \bar{x}_2 \bar{x}_3 \wedge \bar{x}_1 \bar{x}_2 x_3 \wedge \bar{x}_1 x_2 \bar{x}_3$) gives the same decompositions:

$$D_1 : \begin{array}{c} 0110 \\ 1000 \end{array}, D_2 : \begin{array}{c} 0110 \\ 1000 \end{array}, D_3 : \begin{array}{c} 0110 \\ 1000 \end{array}.$$

If not all but just some of the first level decompositions are the same, then it is a partially symmetric function. Other types of symmetry can be detected by a comparison of decompositions corresponding to nodes of higher levels.

Application: Symmetric functions can be synthesized with fewer logic elements, thus they play an important role in logic synthesis and functional verification. They are used for efficient circuit restructuring [3,8,13]. Detection of symmetries is the topic of some recent papers [2,21].

3.1.5 Canalyzing Boolean function

Definition 11 *A Boolean function f on k variables is said to be a canalyzing function if $\exists a, b \in \{0, 1\}$ and $\exists i \in \{1, \dots, k\}$ so that $\forall x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_k$*

$$f(x_1, x_2, \dots, x_{i-1}, a, x_{i+1}, \dots, x_k) = b.$$

Detection: If it exists i , such that the fragment f_a^i of a node in the first level of the DT is a b -constant function, then f is canalyzing. For instance, the function of example 6 is canalyzing. The structure of the DT proposes to extend the canalyzing concept to combinations of inputs. For instance, in example 6 the combination $\bar{x}_2 x_3$ canalyzes from 1 to 1.

Application: Canalyzing functions have simplified logic expressions of the form $f(x) = \tilde{x}_i \odot g(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_k)$, where $g \in \{0, 1\}^{k-1}$, $\odot \in \{\vee, \wedge\}$. There

are many applications in genetic network modeling [15], artificial life [16], nonlinear digital filter design [27], and sociology [17].

3.1.6 Quadratic Boolean function

Definition 12 *Function f is quadratic if the degree of the highest order term in the algebraic normal form is ≤ 2 [23].*

Detection: Linear terms in the algebraic normal form can be detected if the corresponding \oplus -combination in D_i is a constant function (cf. detection of linear functions) and quadratic terms can be obtained if the \oplus -combination in $D_i D_j$ is a constant function: if the \oplus -combination in $D_i D_j$ is 1-constant, then $f(x_1, \dots, x_k) = \tilde{x}_i \tilde{x}_j \oplus g(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_{j-1}, x_{j+1}, \dots, x_k)$, where g does not depend on x_i and x_j .

Application: Cryptography [23].

3.2 Computation of Prime Implicants

An important step for logic minimization of a given Boolean function is the detection of the corresponding prime implicants. All implicants and prime implicants can be detected with the help of the \wedge -combination for each node in the DT.

Definition 13 *An implicant of a function f is a product term p with $1 \leq l \leq k$ literals $\tilde{x}_{i_1} \tilde{x}_{i_2} \dots \tilde{x}_{i_l}$, where $\tilde{x}_i \in \{\tilde{x}_i, x_i\}$ and $i_1 < \dots < i_l \in \{1, \dots, k\}$, which is fully covered by f , so that $p \leq f$. An implicant p of f is called prime, if it is not fully covered by any other implicant of f , i.e. $p \not\leq q$, for any other implicant q of f .*

Each implicant of a given function can be derived from true points of the \wedge -combination, which we call \wedge -patterns.

Example 14 \wedge -combinations for $f(x_1, x_2, x_3) = 11011100$

$$\begin{array}{rcl}
 f_{00}^{12} & = & 11 \qquad \qquad \qquad 10 \\
 \wedge f_{01}^{12} & = & 01 \qquad \qquad \qquad \wedge \quad 11 \\
 D_1 D_2 \wedge f_{10}^{12} & = & 11 \quad D_1 D_3 : \quad \wedge \quad 10 \\
 \wedge f_{11}^{12} & = & 00 \qquad \qquad \qquad \wedge \quad 10 \\
 \hline
 g(x_3) & = & 00 \qquad \qquad \qquad g(x_2) = 10
 \end{array} \tag{5}$$

$D_1 D_2$ has no \wedge - pattern because its \wedge -combination $g(x_3)$ is 0- constant. $D_2 D_3$ and $D_1 D_2 D_3$ have no \wedge - pattern as well. $D_1 D_3$ has the \wedge - pattern in $x_2 = 0$.

We will show that this true point of the \wedge - combination leads to the implicant \bar{x}_2 .

Lemma 15 (\wedge - patterns imply implicants) *If the \wedge -combination of $D_{i_1}D_{i_2}\dots D_{i_l}$ -decomposition has the \wedge - pattern in point (a_{i_1+1}, \dots, a_k) , then function f contains the implicant $p = x_{i_1+1}^{a_{i_1+1}} \dots x_{i_k}^{a_k}$.*

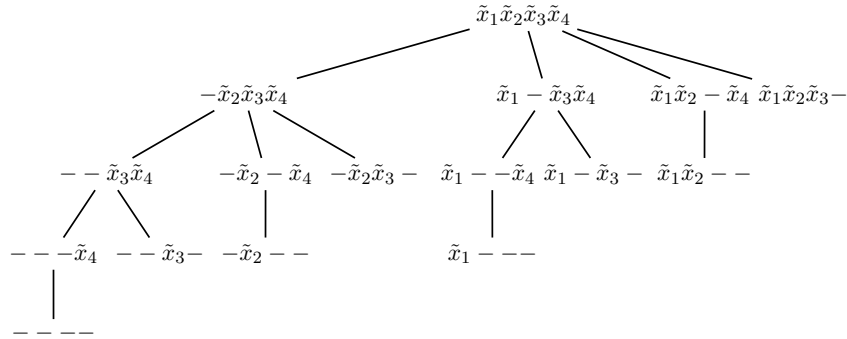
PROOF. Without loss of generality, given the D_1 -decomposition of function f with a \wedge -pattern in point (a_2, \dots, a_k) . We show that the product term $p(x_2, \dots, x_k) = \bigwedge_{i=2}^k x_i^{a_i}$ is an implicant of f . From the \wedge - combination $f(1, a_2, \dots, a_k) = f(0, a_2, \dots, a_k) = 1$ or $1 = p(a_2, \dots, a_k) \leq f(x_1, a_2, \dots, a_k) = x_1 f(1, a_2, \dots, a_k) \vee \bar{x}_1 f(0, a_2, \dots, a_k) = x_1 \vee \bar{x}_1 = 1$, it follows $p \leq f$ for all points, that means by definition 13 p is an implicant of function f . For any decomposition in $1 \leq l \leq k$ inputs, using Shannon Expansion 2^l times, it can be shown that p is an implicant of f .

Lemma 16 (False points of the \wedge -combination) *Given a decomposition $D_{i_1}D_{i_2}\dots D_{i_l}$ and function g on $k-l$ inputs as corresponding \wedge -combination. If $g(a_{i_1+1}, \dots, a_k) = 0$ then product term $p = x_{i_1+1}^{a_{i_1+1}} \dots x_{i_k}^{a_k}$ is no implicant of f .*

Lemma 17 (Termination condition) *If there is no \wedge - pattern in a given decomposition $D_{i_1}D_{i_2}\dots D_{i_l}$, then no product term of a subset of $\{\tilde{x}_{i_1+1}, \tilde{x}_{i_1+2}, \dots, \tilde{x}_{i_k}\}$ is an implicant of f .*

Each decomposition $D_{i_1}D_{i_2}\dots D_{i_l} \in D$, $i_1 < \dots < i_l \in \{1 \dots k\}$ corresponds to a class of implicants: $\tilde{x}_{i_1+1}\tilde{x}_{i_1+2} \dots \tilde{x}_{i_k}$, where $i_1+1 < \dots < i_k \in \{1 \dots k\} \setminus \{i_1, \dots, i_l\}$. It follows that the Decomposition Tree implies an Implicant Tree, which is shown for $k = 4$ in the next example.

Example 18 General Implicant Tree for $k = 4$



For the prime implicants the following special checking condition is needed.

Lemma 19 (Prime implicant checking condition)

If $\exists(a_1, \dots, a_k) \in \{0, 1\}^k$ so that $f(a_1, a_2, \dots, a_k) = 1$ and the k equations are fulfilled:

$$\begin{aligned} f(\bar{a}_1, a_2, \dots, a_k) &= 0 \\ f(a_1, \bar{a}_2, \dots, a_k) &= 0 \\ &\dots \\ f(a_1, a_2, \dots, \bar{a}_k) &= 0, \end{aligned} \tag{6}$$

then a product term $p = x_1^{a_1} x_2^{a_2} \dots x_k^{a_k}$ is a prime implicant of f .

3.2.1 Prime Implicants Computation Algorithm

Generally, in level l of the DT one has to do the operator-combination of 2^l fragments of length 2^{k-l} for each node (cf. Example 2). Considering \wedge -combinations, we can reduce this to a comparison of two fragments of length 2^{k-l} by decomposing the bit string of the \wedge -combination instead that of the input fragments (of level $l-1$). This leads to a significant reduction in time complexity, which is based on the following lemma:

Lemma 20 *Without loss of generality, given a Boolean function $f \in \{0, 1\}^k$ and its \wedge -combination $g \in \{0, 1\}^{k-1}$. If p is an implicant / prime implicant of function g , then p is also an implicant / prime implicant of f .*

PROOF. $p \leq g = f(0, x_2, \dots, x_k) \wedge f(1, x_2, \dots, x_k) \leq f$.

Lemma 20 is used in each node of the DT, so we do not need all 2^l comparisons, but only 2 in each node. The following short recursive algorithm computes the prime implicants of a given Boolean function f :

Algorithm A1. PRIME IMPLICANT COMPUTATION

INPUT : BF $f(x_1, \dots, x_k)$ as a bit string S of length 2^k

OUTPUT : the set of all PIs for f

```
{
  PROCEDURE GetPrimeImplicants( $D_c$ )
  {
     $ChildD_c := \emptyset$ ;
    for all  $s \in D_c$  do
    {
      if  $s \neq \{0\dots 0\}$  then
      {
         $ChildD_c := ChildD_c \cup \wedge$ -combinations(Decompositions( $s$ ));
         $PI := PI \cup TranslateToPI(s, ChildD_c)$ ;
      }
    }
  }
   $D_c := ChildD_c$ ;
```

```

    if  $D_c \neq \emptyset$  then GetPrimeImplicants( $D_c$ );
  }

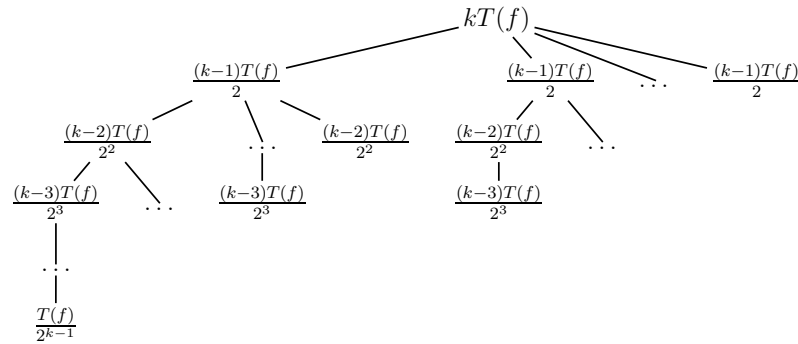
   $PI := \emptyset$ ;
   $D_c := \wedge$ - combinations(Decompositions( $S$ ));
  GetPrimeImplicants( $D_c$ );
  return  $PI$ ;
}

```

The function *Decompositions(s)* builds and returns a set of all possible decompositions of a given function *s* according to the considered level of the DT. Set D_c contains all \wedge - combinations. Function *TranslateToPI(s, ChildD_c)* proofs each \wedge - pattern of *s* (true point of \wedge -combination) according to lemma 19 with help of *ChildD_c*. *ChildD_c* contains all child \wedge - combinations for the considered level. If the \wedge -pattern corresponds to a prime implicant, then it is translated into the corresponding logical expression (lemma 15). Our algorithms are implemented in C++ and are available from the authors upon request.

3.2.2 Time complexity

We make a conservative runtime estimation depending on the number of true points $T(f) = \sum_{i=0}^{2^k-1} f_i$ of a given function *f*. Because of the \wedge -combination, the number of true points that have to be analyzed in each node can be at most half of the number of true points of its parent node. For each of the true points (at most $\frac{T(f)}{2^i}$ for a node in level *i*) one has to do $k - i$ prime implicant checking operations (with the true points of the corresponding \wedge -combinations of the child level). The maximal number of operations per node can be figured out as follows:



The time complexity

$$\sum_{i=0}^{k-1} (k-i) \frac{T(f)}{2^i} \binom{k}{k-i} = T(f) \sum_{i=0}^{k-1} \binom{k}{i} \frac{k-i}{2^i} \leq 2^k (1 + \frac{1}{2})^k k = O(3^k k).$$

Therefore the time complexity for the whole prime implicant algorithm is $O(3^k k)$. This is less than the time complexity needed by the well-known algorithm of Quine and McCluskey for the true table input, which has a runtime of $O(N^{\log^3} \log^2 N) = O(3^k k^2)$ ($N = 2^k$) [30,32]. Of course, heuristics, such as ESPRESSO [25] are faster, but our approach is the fastest exact algorithm for prime implicant calculation which we are aware of.

3.2.3 Space complexity

For traversing the DT it is not necessary to store all nodes. Since our prime implicant algorithm has to store only two successive levels of the tree, the algorithm needs $\max_i \{2^{k-i} \binom{k}{i}\}$ bits for D_c , what can be improved to $O(2^{k \log 3} / k^{1/2})$ (cf. [30], page 102).

3.2.4 Example

Given the function 010011101101001 ($k = 4$). The DT with the corresponding \wedge -combinations is shown in fig. 1. In the first level one obtains the \wedge -combination set $D_c = \{g_1, g_2, g_3, g_4\}$. During calculation of the first level we can mark all \wedge -patterns of function f which are covered by \wedge -patterns of the functions in level one. This results in just one \wedge -pattern at position 10 of f that is not covered by any pattern of level 1. Therefore, we obtain the prime implicant $x_1 \bar{x}_2 x_3 \bar{x}_4$. The \wedge -pattern of g_1 contains all possible implicants of the type $\bar{x}_2 \bar{x}_3 \bar{x}_4$. Since we have three \wedge -patterns in this function (positions 1, 4 and 7) we get the following three different implicants: $x_2 x_3 x_4$, $x_2 \bar{x}_3 \bar{x}_4$ and $\bar{x}_2 \bar{x}_3 x_4$. Combination g_2 leads to the implicant $\bar{x}_1 \bar{x}_3 x_4$. From g_3 we get the implicants $\bar{x}_1 x_2 \bar{x}_4$ and $\bar{x}_1 x_2 x_4$, and from g_4 : $\bar{x}_1 x_2 \bar{x}_3$ and $\bar{x}_1 x_2 x_3$. In the next step we delete level 0 and calculate level 2 from the DT. By decomposing function g_1 we get the following \wedge -combinations: g_5, g_6 and g_7 (0-constant). We do not need to decompose function g_2 , because it only contains one \wedge -pattern which cannot lead to any new \wedge -pattern. The \wedge -combinations g_1 and g_2 produce no \wedge -pattern sets and therefore do not give any additional implicants. The decomposition of function g_3 , which represents implicants of class $\bar{x}_1 \bar{x}_2$, contains one \wedge -pattern, that is the implicant $\bar{x}_1 x_2$.

Again we can mark all \wedge -patterns of the parent level that are covered by \wedge -patterns of the child level. It follows that all implicants found in g_1 and g_2 are prime implicants. The implicants of the functions g_3 and g_4 are fully covered by the implicant $\bar{x}_1 x_2$ and are therefore not prime.

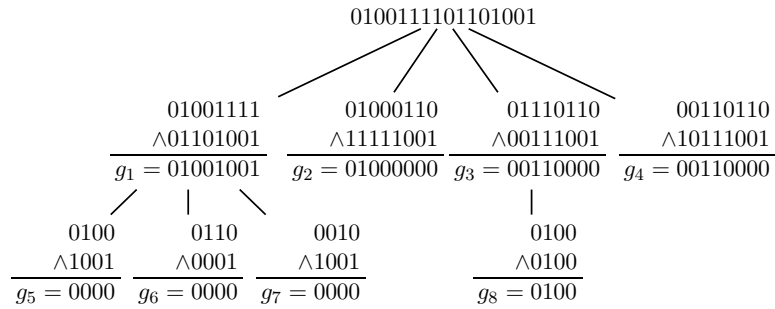


Fig. 1. Decomposition Tree of function 0100111101101001

Summarizing, function f yields the prime implicants $x_1\bar{x}_2x_3\bar{x}_4$, $x_2x_3x_4$, $x_2\bar{x}_3\bar{x}_4$, $\bar{x}_2\bar{x}_3x_4$, $\bar{x}_1\bar{x}_3x_4$, \bar{x}_1x_2 .

3.3 ROBDD construction

The DT can be used to construct the corresponding ROBDD of a given BF: At each decomposition node (starting at the highest level) one has to compare the corresponding decomposition fragments for equivalence. If an equivalence is found the corresponding edge in the ROBDD is constructed, and the DT can be bound appropriately.

3.4 Summary

Table 3.4 summarizes all discussed applications of the Decomposition Tree.

Type of pattern	Detection by the DT
Implicant	\wedge - combination
Prime implicant	\wedge - combination (Algorithm A1)
Clause	\vee -combination
Prime clauses	\vee -combination (analog to Algorithm A1)
Monotonic function	\leq -combinations
Linear function	\oplus -combinations of the D_i
Positive (negative) unate	\leq (\geq)- combination of the D_i
Quadratic function	\oplus -combination of the $D_i, D_i D_j$
Symmetry	All decompositions of the first level are the same.
Canalyzing function	fragment $f_a^i \equiv b$
ROBDD	equivalence of decomposition fragments

4 Conclusion

We presented a new data structure called the Decomposition Tree that provides a unified approach for different analyses of Boolean functions. By decomposing bit strings and combining them by different operators one can classify each given function, construct the corresponding ROBDD, and efficiently compute its prime implicants. The necessary decomposition and operator-combinations can be done in a highly parallel manner, because each node can be computed independently from all other nodes. The DT may also be used for logic minimization: after the fast classification of the function one can apply the appropriate special minimization algorithm. Because the DT represents the most general decomposition of a given Boolean function, we also expect different other possible applications.

References

- [1] H. R. Andersen, *An introduction to binary decision diagrams*, Course Notes on the WWW, 1997.
- [2] L. Benini and G. de Micheli, "A Survey of Boolean Matching Techniques for Library Binding," *ACM Transactions on Design Automation of Electronic Systems*, vol. 2, no. 3, pp. 193-226, 1997.
- [3] J. Boyar, R. Peralta and D. Pochuev, "On the multiplicative complexity of Boolean functions over the basis $(\wedge, \oplus, 1)$," *Theoretical Computer Science*, vol. 235, pp. 43-57, 2000.

- [4] R. E. Bryant, *Symbolic Boolean manipulation with ordered binary decision diagrams*, Pittsburgh, Pa.: School of Computer Science, Carnegie Mellon, 1992.
- [5] R. Brayton, G. Hachtel, C. McMullen and A. Sangiovanni-Vincentelli, *Logic Minimization Algorithms for VLSI Synthesis*, Boston: Kluwer, 1984.
- [6] R. Brayton, "Symbolic Boolean manipulation with ordered binary-decision diagrams," *ACM Computing surveys*, vol. 24, no. 3, pp. 293-318, 1992.
- [7] N. Bshouty and C. Tamon, "On the Fourier spectrum of monotone functions," *J. ACM*, vol. 43, no. 4, pp. 747-770, 1996.
- [8] K. S. Chung and C. L. Liu, "Local transformation techniques for multi-level logic circuits utilizing circuit symmetries for power reduction," *Proceedings of International Symposium on Low Power Electronics and Design*, pp. 215-220, 1998.
- [9] P. Clote and E. Kranakis, *Boolean Functions and Computation Models*, Springer Verlag, 2002.
- [10] O. Coudert, "Two-level logic minimization: an overview, Integration," *VLSI Journal*, vol. 17, no. 2, pp. 97-140, 1994.
- [11] Y. Crama and P. L. Hammer. *Boolean Functions - Theory, Algorithms, and Applications*, In preparation: <http://www.sig.egss.ulg.ac.be/rogp/Crama/>.
- [12] S. Hirose and K. Ikeda, *Nonlinearity criteria of Boolean functions*, KUIS Technical Report KUIS-94-0002, Kyoto, Japan: Kyoto University, 1994.
- [13] S.W. Jeong, T.-S. Kim and F. Somenzi, *An efficient method for optimal BDD ordering computation*, In Proc. International Conference on VLSI and CAD, Taejon, Korea, 1993.
- [14] K. Forbus and J. de Kleer, *Building problem solvers*, MIT Press, 1992.
- [15] S. Kauffman, *Investigation*, Oxford University Press, New York, 2000.
- [16] J. De Kleer , A. Mackworth and R. Reiter, "Characterizing diagnoses and systems," *Artificial Intelligence*, vol. 59, pp. 63-67, 1993.
- [17] J. Klüver and J. Schmidt, "Topology, Metric and Dynamics of Social Systems," *Journal of Artificial Societies and Social Simulation*, vol. 2, 1999.
- [18] K. Makino, and T. Ibaraki, "The maximum latency and identification of positive Boolean functions," *SIAM J. Comput.*, vol. 26, no. 5, pp. 1363-1383, 1997.
- [19] E. J. McCluskey, "Minimization of Boolean functions," *Bell System Technical Journal*, vol. 35, pp. 1417-1444, 1956.
- [20] P. C. McGeer, J. V. Sanghavi, R. K. Brayton, A. L. Sangiovanni-Vincentelli, "Espresso-Signature: A New Exact Minimizer for Logic Functions, *DAC*, pp. 618-624, 1993.
- [21] J. Mohnke and S. Malik, "Permutation and phase independent Boolean comparison," *Integration*, vol. 16, pp. 102-129, 1993.

- [22] J. O. W. Quine, "The Problem of Simplifying Truth Functions," *American Math. Monthly*, vol. 59, pp. 521-531, 1952.
- [23] B. Preneel, R. Govaerts and J. Vandewalle, *Cryptographic properties of quadratic Boolean functions*, In Abstracts of the 1st International Conference on Finite Fields and Applications, 1991.
- [24] R. Reiter, J. De Kleer. *Foundations of Assumption-based Truth Maintenance Systems: Preliminary Report*, AAAI-87, Seattle, Washington, pp. 183-189, 1987.
- [25] R. Rudell, *Espresso software program*. Computer Science Dept., University of California, Berkeley, 1985.
- [26] Z. Szallasi and S. Liang, "Modeling the Normal and Neoplastic Cell Cycle With Realistic Boolean Genetic Networks: Their Application for Understanding Carcinogenesis and Assessing Therapeutic Strategies," *In Pacific Symp. on Biocomputing*, vol. 3, pp. 66-76 1998.
- [27] I. Shmulevich, H. Lähdesmäki and K. Egiazarian, "Spectral Methods for Testing Membership in Certain Post Classes and the Class of Forcing Functions," *IEEE Signal Processing Letters*, vol. 11, no. 2, pp. 289-292, 2004.
- [28] T. Strzemecki, "Polynomial-time algorithms for generation of prime implicants," *Journal of Complexity*, vol. 8, pp. 37-63, 1992.
- [29] Y. Wang, C. McCrosky and X. Song, "Single-faced Boolean Functions and their Minimization," *Computer Journal*, vol. 44, no. 4, pp. 280-291, 2001.
- [30] I. Wegener, *Branching Programms and Binary Decision Diagrams - Theory and Application*, in: SIAM Monographs on Discrete Mathematics and Applications, 2000.
- [31] I. Wegener, *The Complexity of Boolean Functions*, New York: Wiley, 1987.
- [32] I. Wegener, *Effiziente Algorithmen für grundlegende Funktionen*, B.G. Teubner, 1989.
- [33] P. Wendt, E. Coyle and N. Gallagher, "Stack Filters," *IEEE Trans. Acoustics Speech Signal Process*, vol. 34, no. 4, pp. 898-911, 1986.

Discussion

In this PhD thesis we have presented a collection of six articles divided into three sections. All three sections deal with binary patterns on different microbiological level.

In the first two sections we used one single bit to encode all four nucleotide bases. In contrast to more complicated models, proposed by [Bashford and Jarvis \(2000\)](#); [Karasev and Stefanov \(2001\)](#); [Morimoto \(2002\)](#); [He *et al.* \(2004\)](#), and [Sánchez *et al.* \(2004\)](#), which are based on different two bit codings, our new scheme of the genetic code shows the same information content using only one bit. Applying the purine (1) - pyrimidine (0) coding for codons we found a new form of the genetic code shown in Table 3. This new classification scheme consists of 8 rows numbered from 000 up to 111, due to the $2^3 = 8$ binary representations for all possible codons. The column order is defined by the number of hydrogen bonds in the first two codon bases. With help of our new scheme we can explain the small number of different tRNA genes in mitochondria. We have shown that deviations from the standard genetic code are confined to specific regions. Although we reduced the number of fields from 64 to 32, our new classification scheme still highlights known regularities in amino acid - codon assignments more clearly than the common scheme and it even reveals new patterns. Thus, it becomes clear that most amino acid properties are strongly correlated to the corresponding codon-anticodon binding strength ([Wilhelm and Nikolajewa, 2004a](#)). Additionally, all five possible codon symmetries can easily be seen in one table (Tab. 3): [Halitsky \(2003\)](#) point symmetry, codon - anticodon, purine - pyrimidine, sense - antisense

and codon - reverse codon symmetry.

Codon	Strong 6 hydrogen bonds	Mixed 5 hydrogen bonds	Mixed 5 hydrogen bonds	Weak 4 hydrogen bonds
000	Pro CC (C/U)	Ser UC (C/U)	Leu CU (C/U)	Phe UU (C/U)
001	Pro CC (A/G)	Ser UC (A/G)	Leu CU (A/G)	Leu UU (A/G)
100	Ala GC (C/U)	Thr AC (C/U)	Val GU (C/U)	Ile AU (C/U)
101	Ala GC (A/G)	Thr AC (A/G)	Val GU (A/G)	Ile/Met AU (A/G)
010	Arg CG (C/U)	Cys UG (C/U)	His CA (C/U)	Tyr UA (C/U)
011	Arg CG (A/G)	Stop/Trp UG (A/G)	Gln CA (A/G)	Stop UA (A/G)
110	Gly GG (C/U)	Ser AG (C/U)	Asp GA (C/U)	Asn AA (C/U)
111	Gly GG (A/G)	Arg AG (A/G)	Glu GA (A/G)	Lys AA (A/G)

Table 3. The new classification scheme of the genetic code. Each field stands for two codons, where the third bases are given in parentheses. For instance **CC(C/U)** means that the two codons **CCC** and **CCU** encode for *Proline*. The red arrows indicate pairs of codon - reverse codons, where a reverse codon of any codon **XYZ** is defined as **ZYX**.

As shown in our paper (Nikolajewa *et al.*, 2006) the codon - reverse codon symmetry plays an important role and provides new insights into the evolution of the genetic code. The codon - reverse codon patterns are indicated by red arrows in Table 3 and divide our new scheme into four blocks of equal arrow patterns. All strongly evolutionary conserved groups of amino acids (Thompson *et al.*, 1994) are subsets of exactly one codon - reverse codon block. Interestingly all 16 self reverse codons of the genetic code (out of 64) correspond to 15 different amino acids (out of 20). This means that those codons themselves could nearly cover all of the 20 amino acids. Based on the codon - reverse codon symmetry we examined the number of tRNA genes for each anticodon and its reverse anticodon. It is known that STOP codons do not have their own tRNA. We observed, that also the reverse STOP codons do not have own tRNA.

The new scheme of the genetic code and the tRNA usage pattern allow for

speculations about the origin and evolution of the genetic code.

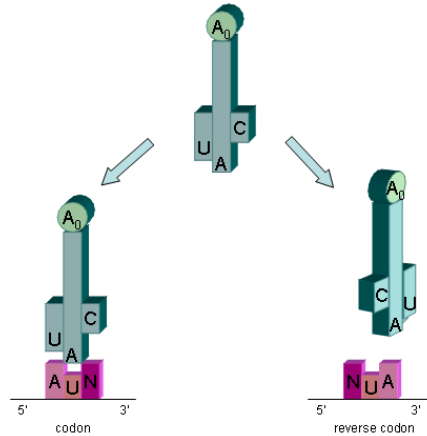


Figure 1. Pre-tRNA with anticodon UAC could recognize codon AUN and its reverse codon NUA.

Is the genetic code really the result of pure chance or a "frozen accident" (Crick, 1968)? In a huge amount of literature (Patel, 2005; Copley *et al.*, 2005; Wu *et al.*, 2005) it is suggested and also following from our reduced scheme that the genetic code had a doublet precursor. If there was a doublet code, how was the genetic information translated? As already Crick (1968) noted, it is very unlikely that there was a doublet reading frame in doublet coding times, because during transition to a triplet reading frame all encoded protein information would be lost. But if there was always a triplet reading frame (Landweber, 2002) also in doublet coding times the information of each third codon base would be wasted. Could nature allow a wasting of 33 % of the RNA information? We think no, and provide a new hypothesis that would solve this problem. We assume that a pre-tRNA had no direction and allowed binding of pre-RNA in both directions (Fig. 1,2).

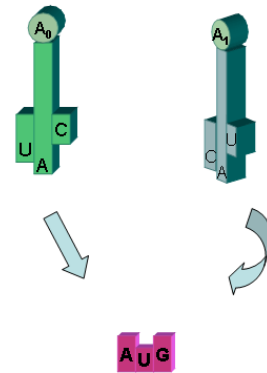


Figure 2. Two different pre-tRNAs with anticodon **UAC** and its reverse anticodon **CAU** could recognize the same codon **AUG**.

One problem with this hypothesis is the given direction in contemporary RNA. But in the paper of Knight and Landweber (2000) PNA⁵ is suggested to be the ancestor of RNA, which had no direction. The concept of the rotating pre-tRNA would also provide a possible solution for the reading frame "problem". In this hypothesis one codon would code for two amino acids (Fig. 1,2), therefore pre-mRNA could lead to many different proteins. To ensure that the proteins have similar properties, we have to assume that a codon and its reverse codon must code either for the same or similar amino acids. Interestingly, this is still the case in the contemporary code (Nikolajewa *et al.*, 2006) and it is also known that the middle base defines important biochemical properties of the amino acid (Knight, 1999). The novel hypothesis could completely change our today's understanding about the origin and evolution of life. The possibility to use two similar amino acids for each codon (Fig. 2) and thus having a large number of similar

⁵PNA is peptide nucleic acid, "...in which the backbone is polymeric N-(2-aminoethyl)glycine (AEG) and the N-acetic acids of the bases are linked via amide bonds" (Knight and Landweber, 2000).

proteins would increase the evolutionary variability enormously. One could still go on and ask whether a single letter code exists before an ancient doublet code. This code could have the contemporary four bases but also their ancestors, which were maybe purine and pyrimidine.

The significant patterns within the genetic code lead us to assume that there are also conserved binary patterns on higher DNA level. Specific Protein DNA binding motifs are often short DNA sequences which are in most cases longer than one single codon. The best studied DNA binding motifs are recognition sequences of restriction enzymes (RE). In our studies we divided them into two subsets of asymmetrical and symmetrical binding sites. Then all dinucleotides, threenucleotides and tetranucleotides were binary translated, according to the three binary coding schemes. Making a comprehensive statistic over all one-bit codings we identified a significant overrepresentation of strings of purines(**R**) (or pyrimidines(**Y**)). In the symmetrical set the most significant dinucleotides are **RR** (or 11) and **YY** (or 00), and in the asymmetric set **RRR**, **YYY** and **YYYY** are even more significant, but **RR** and **YY** also stand out.

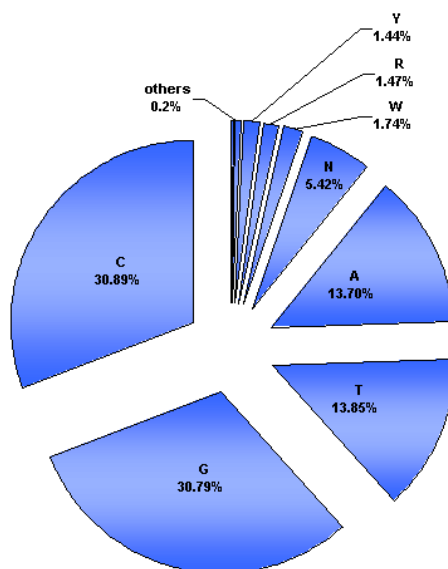


Figure 3. The nucleotide composition of type II recognition sequences.

Moreover, we detected a significant predominance of **G** and **C** over **A** and **T** (Fig.

3). For each of the observations we discussed three possible explanations in our paper (Nikolajewa *et al.*, 2006). The high **G+C** content can be explained by (i) a higher number of methylations (Tab. 4), (ii) the stability of the **GC** base pairing, and (iii) the higher number of bifurcated H-bonds between **GC** base pairs and residues of the recognizing amino acids.

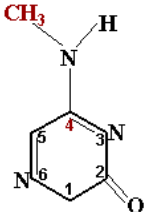
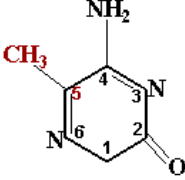
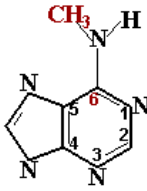
N4-methylcytosine	5-methylcytosine	6-methyladenosine
		
146	1352	524

Table 4. The number of methylations, that reliably prevent DNA cutting of type II restriction enzymes, taken from REBASE (Roberts *et al.*, 2005).

The three possible reasons for the significant nucleotide patterns of the two adjacent purines (in complementary strand pyrimidines) are (i) stronger H-bond donor and acceptor clusters, (ii) a special geometrical arrangement in the DNA structure, and (iii) most important, a lower stacking energy, that allows for conformational changes during the specific protein-DNA binding. All of the explanations refer to changes in DNA geometry and flexibility which probably helps the protein to find its target site and facilitates the DNA binding.

In the last years a large number of articles about DNA - protein interactions has been appeared. But the process of how DNA binding proteins find their recognition sequences is still a mystery. From the last *in vivo* measurements of reaction characteristics it is clear that simple diffusion can not explain the extra rapid association rate constant (Halford and Marko, 2004). The rapid mechanism of protein-DNA association can only mathematically be explained by facilitated

diffusion. The model of facilitated diffusion includes protein sliding and hopping with an optimal sliding range of 100 base pairs (Halford and Marko, 2004). Single molecule experiments on plasmid rings (Gowers and Halford, 2003) confirmed the fact that proteins can jump, to find their target. If this model is true, then an interesting question that could be investigated in further work would be: Is there a significant global DNA pattern that is responsible for the hopping and sliding? The genome-wide pattern would probably lead to a new understanding of "non-coding" DNA sequences. First steps to find global DNA patterns have already been done. Recently, Allen *et al.* (2006) found a long-range pattern in microbial genomes and Yagil (2006) has shown that long DNA tracts, as well as promoter regions are composed of only two of the four bases (what he named "binary DNA").

The genetic information not only consists of statical patterns stored in DNA. It also comprises dynamical patterns of interacting genes. To understand the logic behind gene regulatory networks, we investigated binary patterns of the gene interaction rules. Recent publication have shown that there are special classes of gene regulatory rules. Analyzing the patterns of naturally observed rules (Harris *et al.*, 2002), Kauffman found out that canalyzing functions are biological relevant. In a canalyzing function an input x_i exists, so that a value a ($x_i = a$) can determine the output b of function $f(x)$, independent of the other inputs. It is simple to show that a canalyzing input x_i can be factored out (Tab. 5).

$a \rightarrow b$	logical formula of canalyzing function $f(x)$
$0 \rightarrow 0$	$f(x) = x_i \wedge \hat{f}(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$
$0 \rightarrow 1$	$f(x) = \bar{x}_i \wedge \hat{f}(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$
$1 \rightarrow 0$	$f(x) = \bar{x}_i \vee \hat{f}(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$
$1 \rightarrow 1$	$f(x) = x_i \vee \hat{f}(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$

Table 5. Pattern in the formula representation of a canalyzing function.

In general, the formula for any canalyzing input x_i can be written as

$$f(x_1, x_2, \dots, x_n) = x_i^\sigma \odot \hat{f}(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n),$$

where \odot can either be the binary operation AND or OR, and $\sigma = 0$ stands for the unary negation of x_i ($x_i^{\sigma=1} = \bar{x}_i$).

But the number of all possible canalyzing functions is still too large to model gene networks with always stable dynamical behavior. The pattern within the logical representation of a canalyzing function led us to restrict this class into subclasses, where all inputs can be factored out one by one:

$$f(x_1, x_2, \dots, x_n) = x_{i_1} \odot (x_{i_2} \odot (\dots \odot x_{i_n} \dots)), \quad i_1 \neq i_2 \neq \dots \neq i_n \in \{1, 2, \dots, n\} \quad (1)$$

Reviewing the literature on Boolean networks, we found a description of this class given by [Szallasi and Liang \(1998\)](#), who named functions of this class *hierarchically canalyzing functions* (HCF). They reported that genetic networks, containing only functions of this class, show always a stable dynamical behavior. [Szallasi and Liang \(1998\)](#) have numerically estimated the number of HCFs for small input degrees ($k = 2, 3, 4$). Analyzing the properties of HCFs, we have calculated the exact number of all possible functions for arbitrary input number k .

To proof the belonging of naturally observed rules to the hierarchically canalyzing class, we contacted Mr. Harris, who promised us to provide his data ([Harris et al., 2002](#)). Waiting for this data we unfortunately explained our idea on the "Finnish Signal Processing Symposium", in Tampere, Finland, on May 2003. Some months later (December 2003), [Kauffman et al. \(2003\)](#) published a paper, introducing the hierarchically canalyzing functions and renaming them into "Nested canalyzing functions". They showed that all rules from [Harris et al. \(2002\)](#) data set belong to nested canalyzing functions (or HCFs). After the publication of [Kauffman et al. \(2003\)](#) we received the data from Mr. Harris. Analyzing this data we observed that the rules are much more simpler than those of HCFs. Based on the representation of a hierarchically canalyzing function and according to the operation pattern in formula (1), we divided the HCFs into 2^k subclasses, which we named S_0^k, S_1^k, \dots . We found out that only two of the 2^k subclasses are biologically relevant for classifying [Harris et al. \(2002\)](#) functions. All rules were contained in the first two subclasses (S_0^k and S_1^k). The first class S_0^k consists of

rules where only AND operations are used in the formula (1), whereas the second class has only AND operations except for the last operation, which is OR. Investigating the stability of networks, only made up of these rules, we demonstrated that "Boolean Networks with biologically relevant rules show ordered behavior". Moreover, the larger the number of genes in a regulatory system that are controlled by rules from the S_0^k and S_1^k classes, the more stable the dynamical behavior of the whole system.

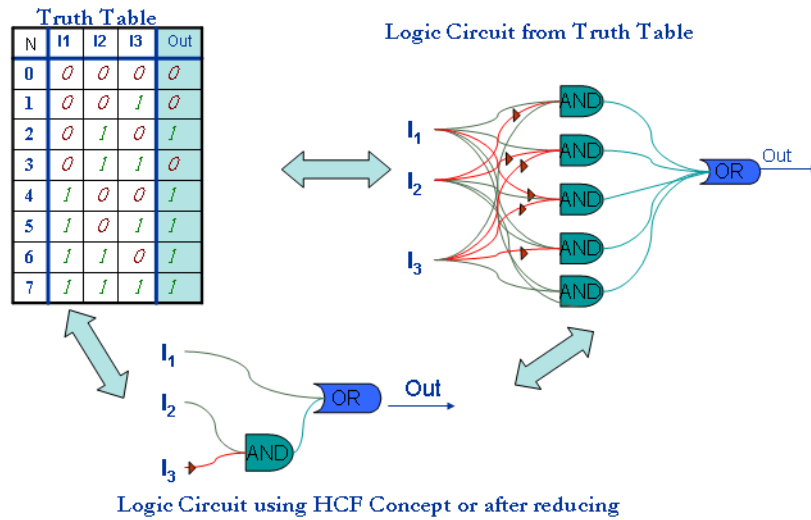


Figure 4. To find a minimal logical formula of a Boolean function is known to be NP-complete (Umans *et al.*, 2005). Interestingly, for a hierarchically analyzing function the minimal logical representation and also its recognition can be done in polynomial time.

To efficiently identify binary patterns in a bit-string representation of a Boolean function, we have developed a binary tree structure, called Decomposition Tree (DT). The structure is based on a bit-string decomposition and substring comparison for each input combination. The Decomposition Tree can be applied to any Boolean function to classify it into important Boolean classes (i.e. linear, monotone, ...). We used this tree structure to recognize canalyzing and hierarchically canalyzing functions. It also calculates the minimal formula for HCFs in polynomial time. Moreover this structure allows us to find other Boolean patterns i.e. unates, prime implicants and clauses.

We have demonstrated that the two simple classes S_0^k and S_1^k support the stable dynamical behavior of cellular processes. One can still wonder, if the simple logical organization (Fig. 4) of these rules really reflect the reality of gene interaction or if it is just an artifact. Maybe the simple rules only derived from the way of thinking during the experiments (Ioannidis, 2005). In contrast to the genetic code and the restriction enzyme binding sites, there is a lack of data for the analysis of gene regulatory rules. To confirm our results about the simple structure of naturally occurring gene regulatory rules, it is necessary to analyze additional datasets. Our restriction of biological relevant rules into the defined subclasses and our "Decomposition Tree" provide powerful tools for further analysis of binary patterns within the gene regulation.

In this work I presented basic patterns of genetic information which lead to a deeper understanding of genetic organization in living things. I was able to show that simple binary patterns are very important and widespread in biological structures. In the genetic code I found the most interesting pattern of my PhD thesis that allows the interpretation of many facts and observations, which are collected since the discovery of the genetic code.

Bibliography

- Allen, T.E., Price, N.D., Joyce, A.R., and Palsson, B.O., 2006. Long-Range Periodic Patterns in Microbial Genomes Indicate Significant Multi-Scale Chromosomal Organization. *PLoS Comput Biol*, vol. 2(1) pp. e2.
- Arber, W., and Linn, S., 1969. DNA Modification and Restriction. *Annual Review of Biochemistry*, vol. 38, pp. 467-500.
- Bashford, J.D., and Jarvis, P.D., 2000. The genetic code as a periodic table: algebraic aspects. *BioSystems*, vol. 57(3), pp. 147-161.
- Calladine, C.R., Drew, H.R., Luisi, B.F., and Travers, A.A., 2006. DNA. Das Molekül und seine Funktionsweise. 3th ed. München: Elsevier GmbH.
- Copley, S.D., Smith, E., and Morowitz, H.J., 2005. A mechanism for the association of amino acids with their codons and the origin of the genetic code, *Proc Natl Acad Sci USA*, vol. 102, pp. 4442-4447.
- Crick, F.H., 1968. The origin of the genetic code. *J Mol Biol*, vol. 38(3), pp. 367-379.
- Di Giulio, M., 2005. The origin of the genetic code: theories and their relationships, a review. *BioSystems*, vol. 80(2), pp. 175-184.
- Gowers, D.M., and Halford, S.E., 2003. Protein motion from non-specific to specific DNA by three-dimensional routes aided by supercoiling. *EMBO J*, vol. 22, pp. 1410-1418.

- Halitsky,D., 2003. Extending the (hexa-)rhombic dodecahedral model of the genetic code: the codes 6-fold degeneracies and the orthogonal projections of the 5-cube as 3-cube. Contributed paper (983-92-151), American Mathematical Society; and personal communication.
- Halford,S.E., and Marko,J.F., 2004. How do site-specific DNA-binding proteins find their targets? *Nucl Acids Res*, vol. 32, pp. 3040-3052.
- Harris,S., Sawhill,B., Wuensche,A., and Kauffman,S., 2002. A Model of Transcriptional Regulatory Networks Based on Biases in the Observed Regulation Rules. *Complexity*, vol. 7(4), pp. 23-40.
- Hasty,J., McMillen,D., and Collins,J.J., 2002. Engineered gene circuits. *Nature*, vol. 420, pp. 224-230.
- He,M., Petoukhov,S.V., and Ricci,P.E., 2004. Genetic Code, Hamming Distance and Stochastic Matrices. *Bull Math Biol*, vol. 00, pp. 1-17, doi:10.1016/j.bulm.2004.01.002.
- Hiller, M., Huse,K., Szafranski,K., Jahn, N., Hampe,J., Schreiber,S., Backofen,R., and Platzer,M., 2004. Widespread occurrence of alternative splicing at NAG-NAG acceptors contributes to proteome plasticity. *Nature Genetics*, vol. 36(12), pp. 1255-1257.
- Hoag,H., 2006. Drug hunt. *Nature*, vol. 439, pp. 886-887.
- Ioannidis,J.P., 2005. Why most published research findings are false. *PLoS Med* vol. 2(8), e124.
- Jiménez-Montano,M.A., de la Mora-Basanez,C.R., and Poschel,T., 1996. The hypercube structure of the genetic code explains conservative and non-conservative amino acid substitutions in vivo and in vitro. *BioSystems*, vol. 39, pp. 117-125.
- Karasev,V.A., and Stefanov,V.E., 2001. Topological Nature of the Genetic Code. *J Theor Biol*, vol. 209, pp. 303-317.

- Kauffman,S., 1969. Metabolic Stability and Epigenesis in Randomly Constructed Genetic Nets, *J Theor Biol* vol. 22, pp. 437-467.
- Kauffman,S., 1993. The Origins of Order: Self-Organization and Selection in Evolution. Oxford University Press, New York.
- Kauffman,S., Peterson,C., Samuelsson,B., and Troein,C., 2003. Random Boolean network models and the yeast transcriptional network. *Proc Natl Acad Sci U S A*, vol. 100(25), pp. 14796-14799. Epub 2003 Dec 1.
- Knight,R.D., Freeland,S.J. and Landweber,L.F., 1999. The 3 Faces of the Genetic Code. *Trends in the Biochemical Sciences*, vol. 24(6), pp. 241-247.
- Knight,R.D., and Landweber,L.F., 2000. The early evolution of the genetic code. *Cell*, vol. 101, pp. 569-572.
- Lagerkvist,U., 1978. "Two out of three": An alternative method for codon reading. *Proc Natl Acad Sci USA*, vol. 75, pp. 1759-1762.
- Landweber, L.F., 2002. Custom codons come in threes, fours, and fives. *Chem Biol*, vol. 9, pp. 143.
- Luscombe,N.M., Greenbaum,D., and Gerstein, M., 2001. What is Bioinformatics? A Proposed Definition and Overview of the Field. *Methods of information in medicine*, vol. 40(4), pp. 346-358.
- Monod,J., and Jacob,F., 1961. General conclusions: telenomic mechanisms in cellular methabolism, growth, and differentiation. *Cold Spring Harb. Symp. Quant. Biol.*, vol. 26, 389-401.
- Morimoto,S., 2002. A periodic table for genetic codes, *Journal of Math Chem*, vol. 32(2), pp. 159-200.
- Nikolajewa,S., Friedel,M., Beyer,A. and Wilhelm,T., 2006. The new classification scheme of the genetic code, its early evolution, and tRNA usage, *J Bioinf and Comp Biol*, in press.

- Nikolajewa,S., Beyer,A., Friedel,M., Hollunder,J.,and Wilhelm,T., 2005. Common patterns in type II restriction enzyme binding sites. *Nucleic Acids Research*, vol. 33(8), pp. 2726-2733.
- Noramly,S., Zimmerman,L., Cox,A., Aloise,R., Fisher,M., and Grainger,R.M., 2005. A gynogenetic screen to isolate naturally occurring recessive mutations in *Xenopus tropicalis*. *Mechanisms of Development*, vol. 122, pp. 273-287.
- Palmer,A.R., 2004. Symmetry breaking and the evolution of development. *Science*, vol. 306(5697), pp. 828-833.
- Patel,A., 2005. The triplet genetic code had a doublet predecessor. *J Theor Biol*, vol. 233, pp. 527-532.
- Rigoutsos,I., and Floratos,A., 1998. Combinatorial pattern discovery in biological sequences: the TEIRESIAS algorithm. *Bioinformatics*, vol. 14, pp. 55-67.
- Roberts, R.J. et al. REBASE - restriction enzymes and DNA methyltransferases. *Nucleic Acids Res*, vol. 33, D230.
- Roberts,R.J., Belfort,M., Bestor,T., Bhagwat,A.S., Bickle,T.A., Bitinaite,J., Blumenthal,R.M., Degtyarev,S.Kh., Dryden,D.T., Dybvig,K., et al. 2003. A nomenclature for restriction enzymes, DNA methyltransferases, homing endonucleases and their genes, *Nucleic Acids Res*, vol. 31, pp. 1805-1812.
- Sánchez,R., Morgado,E., and Grau, R., 2004. The Genetic Code Boolean Lattice. *MATCH Commun Math Comput Chem*, vol. 52, pp. 29-46.
- Shannon,C.E., 1948. A mathematical theory of communication. *Bell System Technical Journal*, vol. 27, pp. 379-423 and 623-656.
- Stambuk,N., 2000. Universal Metric Properties of the Genetic Code. *Croatica Chemica Acta*, vol. 73, pp. 1123-1139.
- Stambuk,N., and Konjevoda,P., 2005. Binary coding of the secondary protein structure, *Periodicun Biologorum*, vol. 107(4), pp. 393-396.

- Stormo,G.D., 2000. DNA binding sites: representation and discovery. *Bioinformatics*, vol. 16, pp. 16-23.
- Szallasi,Z., and Liang,S., 1998. Modeling the normal and neoplastic cell cycle with realistic Boolean Genetic Networks: Their application for understanding carcinogenesis and assessing therapeutic strategies. in: *Pacific Symp. on Bio-computing* vol. 3, pp. 66-76.
- Taylor,F.J., and Coates,D., 1989. The code within the codons, *BioSystems*, vol. 22(3), pp. 177-187.
- Thompson,J.D., Higgins,D.G., and Gibson,T.J., 1994. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res*, vol. 22, pp. 4673-4680.
- Umans,C., Villa,T., and Sangiovanni-Vincentelli, A.L. 2005. How hard is two-level logic minimization: an addendum to garey & Johnson. Technical report, International Workshop on Logic and Synthesis, June 2005.
- Walker,M.G., Volkmuth,W., Sprinzak,E., Hodgson,D., and Klingler,T., 1999. Prediction of gene function by genome-scale expression analysis: Prostate cancer-associated genes. *Genome Res*, vol. 9, pp. 1198-1203.
- Wilhelm,T., and Nikolajewa,S., 2004. A new classification scheme of the genetic code. *J Mol Evol*, 59, pp. 598-605.
- Wilhelm,T., and Nikolajewa,S., 2004. A purine-Pyrimidine Classification Scheme of the Genetic Code. *BIOforum Europe*, vol. 6, pp. 46-49.
- Woese,C.R., Olsen,G.J., Ibba,M., and Söll,D., 2000. Aminoacyl-tRNA Synthetases, the Genetic Code, and the Evolutionary Process, *Microbiology and Molecular Biology Reviews*, vol. 64, pp. 202-236.

Wu,H.L., Bagby,S., and van den Elsen,J.M., 2005. Evolution of the genetic triplet code via two types of doublet codons, *J Mol Evol*, vol. 61, pp. 54-64.

Yagil,G., 2006. DNA tracts composed of only two bases concentrate in gene promoters. *Genomics*, e-pub: Jan 24.

Zhang,M.Q., 2002. Computational prediction of eukaryotic protein-coding genes. *Nature Rev Genetics*, vol. 3, pp. 698-709.

Anhang

Angabe zum Eigenanteil

Titel	Literaturangabe	Autoren	Arbeitsanteil
A new classification scheme of the genetic code.	<i>J. of Mol. Evol.</i> , vol. 59, pp. 598-605, 2004.	Thomas Wilhelm, Swetlana Nikolajewa.	50% 50%
A purine-pyrimidine classification scheme of the genetic code	<i>BIOforum Europe</i> vol. 6, pp. 46-49, 2004.	Thomas Wilhelm, Swetlana Nikolajewa.	50% 50%
The new classification scheme of the genetic code, its early evolution, and tRNA usage.	<i>J Bioinformatics and Computational Biology</i> accepted on December, 22 th 2005, in press.	Swetlana Nikolajewa, Maik Friedel, Andreas Beyer, Thomas Wilhelm.	70% 10% 10% 10%
Common patterns in type II restriction enzyme binding sites.	<i>Nucleic Acid Research</i> , vol. 33(8), pp. 2726-2733, 2005.	Swetlana Nikolajewa, Andreas Beyer, Maik Friedel, Jens Hollunder, Thomas Wilhelm.	70% 7% 7% 7% 9%
Boolean Networks with biologically relevant rules show ordered behavior.	Submitted to the <i>BioSystems</i> on November, 18 th 2005.	Swetlana Nikolajewa, Maik Friedel, Thomas Wilhelm.	70% 15% 15%
The Decomposition Tree for analysis of Boolean Functions.	Submitted to the <i>J. of Complexity</i> on March, 2 nd , 2006	Maik Friedel, Swetlana Nikolajewa, Thomas Wilhelm.	70% 15% 15%

Lebenslauf

Persönliche Daten

Name: Swetlana Nikolaeva

Geburtsdatum: 19. November 1975

Geburtsort: Rostow-am-Don, Russland

Nationalität: russisch

Familienstand: geschieden

Ausbildung

1983-1993: Schule N 87 mathematischer Richtung in Rostow-am-Don,
Abschluss mit "Goldener Medallie"

1993-1997: Abschluss als Bachalaureus der Mathematik (mit Auszeichnung),
Rostower Staatliche Universität

1997-1999: Abschluss als Magister der Mathematik (mit Auszeichnung),
Rostower Staatliche Universität

1999-2002: Aspiranturstudium an der Fakultät für Mechanik und Mathematik,
Rostower Staatliche Universität, bei Prof. Dr. Sergey Zhak

2002: DAAD-Stipendiatin an der Friedrich-Alexander-Universität,
Erlangen-Nürnberg

seit 2003: Promotion am Institut für Molekulare Biotechnologie in Jena
bei Dr. Thomas Wilhelm

Publikationen

1. Thomas Wilhelm and Swetlana Nikolajewa, 2004. A new classification scheme of the genetic code. *Journal of Molecular Evolution*, volume 59, pp. 598-605.
2. Thomas Wilhelm and Swetlana Nikolajewa, 2004. A purine-pyrimidine classification scheme of the genetic code. *BIOforum Europe Journal*, volume 6, pp. 46-49.
3. Swetlana Nikolajewa, Maik Friedel, Andreas Beyer and Thomas Wilhelm, 2006. The new classification scheme of the genetic code, its early evolution, and tRNA usage. *Journal of Bioinformatics and Computational Biology*, accepted to the publication on 22 December 2005, in press.
4. Swetlana Nikolajewa, Andreas Beyer, Maik Friedel, Jens Hollunder and Thomas Wilhelm, 2005. Common Patterns in Type II Restriction Enzyme Binding Sites. *Nucleic Acid Research*, volume 33(8), pp. 2726-2733.
5. Swetlana Nikolajewa, Maik Friedel, and Thomas Wilhelm. Boolean Networks with biologically relevant rules show ordered behavior. Submitted to the *BioSystems* on November, 18th, 2005.
6. Maik Friedel, Swetlana Nikolajewa and Thomas Wilhelm. The Decomposition Tree for analysis of Boolean functions. Submitted to the *Journal of Complexity* on March, 2nd 2006.

7. Swetlana Nikolajewa, Maik Friedel, Andreas Beyer, and Thomas Wilhelm. Purine-pyrimidine patterns in the genetic code and in restriction enzyme recognition sequences, *Proceedings of MCCMB'05*, Moscow, Juli 18-21, 250.
8. Swetlana Nikolajewa, Maik Friedel, Andreas Beyer, and Thomas Wilhelm, New classification scheme of the genetic code, and early evolution of translation, Tagungsband zur VAAM-Jahrestagung 2006, 213, Jena 19.-22.März 2006.

Vorträge

1. Swetlana Nikolajewa, Maik Friedel, Andreas Beyer, and Thomas Wilhelm, Purine-Pyrimidine patterns in the genetic code and in restriction enzyme recognition sequences, MCCMB'05, Moscow, Juli 19, 2005.
2. Swetlana Nikolajewa, and Thomas Wilhelm, A new classification scheme of the genetic code, Jena, Workshop JCB, Dezember, 3, 2004.
3. Swetlana Nikolajewa, Maik Friedel, Andreas Beyer, and Thomas Wilhelm, New classification scheme of the genetic code, its early evolution, and tRNA usage, Jena, Workshop JCB, März, 31, 2006.
4. Thomas Wilhelm, Swetlana Nikolajewa, Maik Friedel, Andreas Beyer, and Jens Hollunder, Common patterns in type II restriction enzyme binding sites, ELSO'05, Dresden, September, 6, 2005.
5. Thomas Wilhelm, Andreas Beyer, Swetlana Nikolajewa, Jens Hollunder, Regina Brockmann, Maik Friedel, Johannes Wollbold, Tommi Aho. Patterns in biological networks. NiSIS 2005. European Symposium on Nature-inspired Smart Information Systems, Albufeira, Portugal Final Programme & Proceedings, October 4 - 5, 2005,

Poster

1. Wilhelm T., Nikolajewa S., Beyer A., Friedel M., and Hollunder J., Common patterns in type II restriction enzyme binding sites, *ELSO Meeting Poster Abstracts*, Dresden, September, 6, 2005, [Abstract](#).
2. Wilhelm T. and Nikolajewa S., A new classification scheme of the genetic code, *5th International conference on Systems Biology*, **Poster**: Heidelberg, October 9-13, 2004.
3. Nikolajewa S. and Wilhelm T., Purine-Pyrimidine patterns in the genetic code and in protein-DNA binding sitesclassification scheme reveals new patterns in the genetic code, *German Conference on Bioinformatics*, **Poster**, Hamburg, October 5-7, 2005.
4. Nikolajewa S. and Wilhelm T., The Purine-Pyrimidine classification scheme reveals new patterns in the genetic code, *Foundations of Systems Biology in Engineering*, **Poster**, Santa Barbara, August 7-10, 2005.
5. Nikolajewa S., Beyer A., Friedel M., Hollunder J., and Wilhelm T., Common patterns in type II restriction enzyme binding sites, *ECCB/JBI Computational Biology*, **Poster**, Madrid, September 28 - Oktober 1, 2005. [Abstract](#)
6. Nikolajewa S., Friedel M., Beyer A., and Wilhelm T., New classification scheme of the genetic code, and early evolution of translation, VAAM-Jahrestagung 2006, Jena 19.-22.März 2006.

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Hilfsmittel angefertigt habe. Mir ist die geltende Promotionsordnung bekannt und ich habe weder die Hilfe eines Promotionsberaters in Anspruch genommen, noch haben Dritte unmittelbare oder mittelbare geldwerte Leistungen für Arbeit erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen.

Die vorgelegte Dissertation wurde noch nicht als Prüfungsarbeit für eine staatliche oder andere wissenschaftliche Prüfung eingereicht. Weiterhin habe ich mich mit der vorliegenden Arbeit an keiner anderen Hochschule um den akademischen Grad doctor rerum naturalium (Dr. rer. nat.) beworben und weder früher noch gegenwärtig die Eröffnung eines Verfahrens zum Erwerb des o.g. akademischen Grades an einer anderen Hochschule beantragt.

Bei der Auswahl und Auswertung des Materials, sowie bei der Herstellung des Manuskripts hat mich meine Arbeitsgruppe Theoretische Systembiologie unter Leitung von Dr. Thomas Wilhelm unterstützt.

Jena, den April 5, 2006

.....
(Swetlana Nikolajewa)

Danksagung

Besonderen Dank möchte ich an die Arbeitsgruppe Theoretische Systembiologie (TSB) des Leibniz-Instituts für Altersforschung e.V. Fritz-Lipmann-Instituts (vormals IMB Jena) richten, ohne die meine Dissertation in dieser Form unmöglich gewesen wäre. Mein Dank geht dabei in erster Linie an meine Mitautoren Dr. Thomas Wilhelm und Maik Friedel. Ich danke meinem Freund Maik für die liebevolle und geduldige Unterstützung, Dr. Wilhelm für die einzigartige und unkomplizierte Betreuung und Prof. Dr. S. Schuster für seine freundliche Betreuung.

Weiterhin möchte ich meinen Eltern und meinem Bruder danken, die all die Jahre an mich geglaubt und mir dieses Studium erst ermöglicht haben.