

IMPLEMENTATION OF THE “BIOSYNTHESIS OF NUCLEIC ACIDS AND PROTEINS: GENETIC CODE” TOPIC FOR SECONDARY SCHOOLS FOCUSED ON INFORMATICS

Mária Kožurková^{1*}, Mária Ganajová²

¹Department of Biochemistry, Institute of Chemistry, Faculty of Science, Pavol Jozef Šafárik University in Kosice, 041 80 Košice, Slovakia

²Department of Didaktik Chemistry, Institute of Chemistry, Faculty of Science, Pavol Jozef Šafárik University in Kosice, 041 80 Košice, Slovakia

Abstract

The strategic aim of the “IT Academy – Education for the 21st Century” project was to create an educational model which would prepare young people for the current and future needs of society and the job market, focusing on Informatics and also Information and Communication Technologies. The new subject of Informatics in Natural Sciences and Mathematics consists of five modules that broaden and deepen students' knowledge of Informatics. The Chemistry module focuses on creating models, working with databases, and using selected software programs to teach Chemistry. One of the topics of the module examines the biosynthesis of nucleic acids and proteins, and the use of programming languages such as Python to create genetic code-making models. Students are given the opportunity to work with programming languages, identify differences in the structure and properties of DNA and RNA, and explain the transition of genetic information, transcription and the synthesis of proteins. The content of this task has been selected to present examples of the practical use of Informatics in order to deepen students' knowledge of Chemistry.

Keywords: *information and communication technology, chemistry teaching, Python language*

1. INTRODUCTION

The Faculty of Natural Sciences at Pavol Jozef Šafárik University in Košice is involved in the development of the “IT Academy – education for the 21st century” project which aims to create a model of Informatics classes intended primarily for use in grammar schools. As part of the program, the new subject of Informatics in Natural Sciences (INS) and Mathematics has been created for grammar school students in their 3rd or the 4th years of study.

INS has been designed to show students how Informatics can be applied in other fields. Students begin their studies with a mandatory Informatics section of the module before moving on to modules oriented towards Natural Sciences such as Chemistry. The thematic units of these modules connect their content with Informatics and with the knowledge students have gained in corresponding subjects [1].

The Chemistry module is focused on making models and databases in Chemistry. The tasks presented and solved in the module deepen students' understanding of the structure of chemical compounds, dependencies, and processes and also allow students to see the growing importance of Informatics in research and practice. The strategic aim of the IT (Information Technology) Academy project is to create a model of education which can prepare young people for the current and future needs of society and the job market, focusing on Informatics and Information and Communication Technologies [2].

2. METHODOLOGY

2.1 Materials

Python programming language

The Python programming language was chosen to solve the tasks in the project.

The development tools used (Pycharm – IDE - Integrated development environment - integrated development environment) operate via JRE (Java Runtime Environment), and therefore it is not necessary to install Java to run the Pycharm program.

Pycharm Professional Edition is also available as a paid professional software package which supports several programming languages and includes web development and database management tools. An opensource Community Edition is also available for free and enables the use of Python, XML and JSON for scientific purposes.

Several versions of the Python programming have been developed:

Version 2 - 2.6 and 2.7

Version 3 - from 3.4 to 3.7

Pycharm 2018.1 (Community Edition) and Python 3.6.5 were used in the project (Fig. 1).

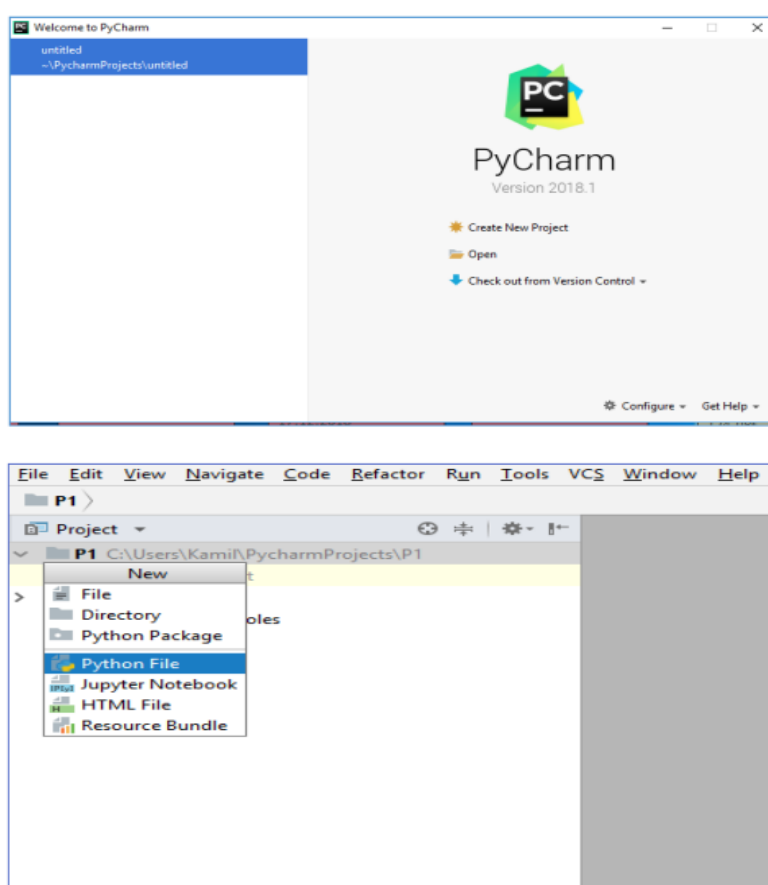


Fig. 1. Pycharm 2018.1 and Python 3.6.5

2.2 Methods

Genetic code is a method of transcribing genetic information through the order of the nitrogenous bases in the DNA structure (A, T, C, G). The essence of genetic information is the coding of the order of amino acids in a peptide chain which produces a protein. Around twenty amino acids are involved in the formation of proteins and, given that RNA uses four types of nucleotides (bases), an amino acid must therefore be determined by at least three bases. These sets of nitrogenous bases are called triplets. A triplet that codes for an amino acid is called a codon [3-5].

In the tasks created for the project, students learn how to use the Python programming language to solve examples of translation, transcription and genetic code. Genetic information is a message written in the structure of the DNA molecule which allows an organism to create a certain characteristic in its specific form. Genetic information is stored in linear DNA molecules and is transcribed using genetic code (Fig. 2).

Students' requirements for entry knowledge and skills:

DNA, RNA, DNA synthesis - replication, RNA synthesis - transcription, protein synthesis (proteosynthesis - translation), genetic information, adenine, guanine, cytosine, uracil, thymine, nucleotide, amino acid, messenger, transfer, ribosomal RNA, complementarity, codon, anticodon, polynucleotide chain, polypeptide chain. Genetic code universal and degenerate.

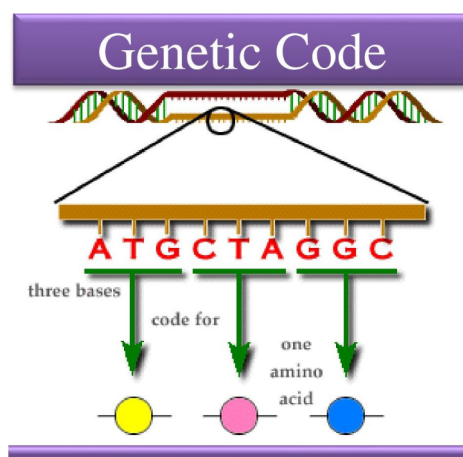


Fig. 2. Genetic code [5].

Knowledge acquired by students through the Biosynthesis of Nucleic Acids and Proteins: Genetic Code topic

- an understanding of the building blocks and structure of nucleic acids and proteins.
- an awareness of the importance of proteins and nucleic acids in organisms.
- the ability to explain the differences in the structure and properties of DNA and RNA.
- an understanding of how individual types of RNA function.
- an understanding of the method of transmission of genetic information - transcription and proteosynthesis.

Skills developed by students

- to apply the principle of complementarity to examples,
- to use the genetic code when solving examples,
- to apply knowledge about the direction of transcription and translation,
- to program solutions to problems using the Python language,
- to apply modelling procedures to new problems,
- to develop skills associated with the development of thinking and learning.

Solution of educational issue

In Chemistry classes, the teaching of nucleic acids and proteins is typically limited to an explanation of their construction and structure, and within the framework of the basics of biochemistry, teachers lack sufficient time to address their biosynthesis. This shortfall is often reflected in the fact that students have difficulty understanding and applying some principles of the transfer of genetic information in biology.

As a means of strengthening interdisciplinary relationships and interactions with biology, it is also appropriate to practice examples of replication, transcription and translation in Chemistry, and this is made possible in an interesting and unusual way through the use of the Python programming language.

3. RESULTS

3.1 Modelling in the Python program for the Biosynthesis of Nucleic Acids and Proteins: Genetic Code topic

The Python programming language is interactive and dynamically typed and is therefore a more suitable tool for teaching programming than statically typed languages where all types of data must be input in advance. The Python language also integrates many advanced features of modern programming languages, such as support for working with data structures and object-oriented software development. Python is also a universal programming language that provides means for creating modern applications, such as data analysis, media processing and network applications [6,7].

Genetic code is a method of transcribing genetic information based on the order of nitrogenous bases in the DNA structure [3-5]. As means of strengthening interdisciplinary relationships and interactions with biology, it is also an appropriate tool for practising examples of replication, transcription and translation in chemistry, and this is made possible in an interesting and unusual way through the Python programming language. The goal of this methodology is to teach students how to use the Python programming language to solve amino acid sequence notations and perform translation and transcription notations.

Methodology sheets for teachers and worksheets for students have been created for the topics of the mentioned modules.

3.2 Course of teaching

The teacher should check students' knowledge about the structure and properties of nucleic acids and their biosynthesis and must know the solutions to the worksheet. This methodology builds on knowledge obtained from the foregoing Substances in Living Organisms topic [8].

3.3 Knowledge that should be repeated with students - nucleic acids

Content standard:

genetic information, adenine, guanine, cytosine, uracil, thymine, nucleotide, DNA, RNA, messenger, transfer, ribosomal RNA, complementarity, codon, anticodon, polynucleotide chain.

Performance standard:

- ability to characterize nucleic acids in terms of occurrence and meaning,
- ability to characterize nucleic acids in terms of their classification and composition,
- ability to compare the structure of DNA and RNA,
- ability to characterize messenger, transfer and ribosomal RNA in terms of their function and occurrence in the cell,
- ability to explain the meaning of the concept of complementarity in the case of DNA,
- ability to explain the difference between universal and degenerate genetic code.

Several tasks with this topic were selected in the project [9]. Two of these tasks are presented below.

TASK 1

You have a given section of the DNA strand:

3'GTCAAGTCAAAC 5'.

Add a complementary strand to the given strand to restore the double helix.

Write the example in the Python program.

Solution:

According to Chargaf's rule, there must be an equal representation of purine (adenine A, guanine G) and pyrimidine (cytosine C, thymine T) bases in the DNA molecule. Adenine was found to pair with thymine A-T, and cytosine with guanine G. The strands of the double helix are oriented in an antiparallel arrangement. DNA strand has two ends 3' and 5'. When transcribing the complementary fibre, we must take into account the fact that the second fibre is oriented in an antiparallel manner.

Results:

3' GTCAAGTCAAAC 5'
5' CAGTTCAGTTTG 3'

Solution to Task 1 in Python:

```
character=["3","5","A","T","G","C"]
#working strand
def create working strand(input):
    a = ""
    for i in range(len(input)):
        if input[i] == "5":
            a = a + "3"
        elif input[i] == "3":
            a = a + "5"
        elif input[i] == "A":
            a = a + "T"
        elif input[i] == "T":
            a = a + "A"
        elif input[i] == "G":
            a = a + "C"
        elif input[i] == "C":
            a = a + "G"
        else: a = a + input[i]
    return a
coding strand = input("enter the coding strand:")
```

```
i = len(coding strand)
if i < 5: 240 print("strand too short")
exit(0)
if (coding strand[0] != '3') and (coding strand[0] != '5'):
print("Incorrect first character")
exit(0)
if (coding strand[i-1] != '3') and (coding strand[i-1] != '5'):
print("Incorrect last character")
exit(0)
if coding strand[0] == coding strand[i-1]:
print("Illegal match between first and last character")
exit(0)
for j in range (0,i):
if coding strand[j] not in character: print("Incorrect character in strand: "+ coding strand[j]) exit(0)
working strand = create_ working strand(coding strand)
print("Working strand:", working strand)
```

TASK 2

- You have the indicated section of the mRNA strand: 5 AUGCAAUGA 3'.
- Make a translation and write down the resulting amino acid sequence.
- Write the example in the Python program.

Solution:

Translation takes place on ribosomes where the matrix (mRNA strand) is read in the 5'→3' direction. The genetic code is a triplet, and each specific example can be recognized as an amino acid due to the fact that they are always determined by three neighbouring nucleotides. These triplets are labelled on the mRNA as codons. Amino acids are brought to the site of photosynthesis bound to tRNA molecules.

tRNA molecules have a triplet of bases included in their sequence which is complementary to a specific triplet - codon on mRNA. When bound to tRNA, this triplet is called an anticodon. The codon - anticodon pairing ensures the correct placement of amino acids depending on the mRNA matrix. This task will be solved according to the genetic code table (Fig. 3), where the triplets are listed with the corresponding amino acid.

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

Fig. 3. Genetic code table [7]

Translation is initiated by the initiation codon AUG (also coding for methionine) and terminated by codon terminations UAA, UAG and UGA.

Results:

met-glu

Solution to Task 1 in Python:

```
aminoacids = { "UUU": "phe", "UUC": "phe", 242 "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", "CUU": "leu", "CUC": "leu",
"CUA": "leu", "CUG": "leu", "CCU": "pro", "CCC": "pro", "CCA": "pro", "CCG": "pro", "CAU":
"his", "CAC": "his", "CAA": "glu", "CAG": "glu", "CGU": "arg", "CGC": "arg", "CGA": "arg",
"CGG": "arg", "AUU": "ile", "AUC": "ile", "AUA": "ile", "AUG": "met", "ACU": "thr", "ACC":
"thr", "ACA": "thr", "ACG": "thr", "AAU": "asn", "AAC": "asn", "AAA": "lys", "AAG": "lys",
"AGU": "ser", "AGC": "ser", "AGA": "arg", "AGG": "arg", "GUU": "val", "GUC": "val", "GUA":
"val", "GUG": "val", "GCU": "ala", "GCC": "ala", "GCA": "ala", "GCG": "ala", 243 "GAU": "asp",
"GAC": "asp", "GAA": "glu", "GAG": "glu", "GGU": "gly", "GGC": "gly", "GGA": "gly", "GGG":
"gly"
}
znaky=["3","5","A","U","G","C"]
inic_aacid = "met"
```

working strand

def create working strand(input):

a = "" for i in range(len(input)):

if input[i] == "5":

a = a + "3"

elif input[i] == "3":

```
a = a + "5"
elif input[i] == "A":
a = a + "T"
elif input[i] == "T":
a = a + "A"
elif input[i] == "G":
a = a + "C"
elif input[i] == "C":
a = a + "G"
else: a = a + input[i] return a
```

```
def create_mrna(vstup):
b = ""
for i in range(len(vstup)):
    if input[i] == "5":
b = b + "3"
    elif input[i] == "3":
b = b + "5"
    elif input[i] == "A":
b = b + "U"
    elif input[i] == "T":
b = b + "A"
    elif input[i] == "G":
b = b + "C"
    elif input[i] == "C":
b = b + "G"
else: b = b + input[i]
return b
```

```
def create_tripeptid(input):
if vstup[0] == "3":
x = otocenie(vstup)
print("otoceny:", x)
else:
# x = input[2:-2]
x = input[1]+ input[2]+ input[3]
```

```
triplet = ""
aminoacid = ""
i = 0
while i < len(x):
    triplet = x[i] + x[i + 1] + x[i + 2]
    codon = aminoacids[triplet]
    if codon == "stop":
        break
    else:
        aminoacid = aminoacid + "-" + codon
    i = i + 3
aminoacid = aminoacid[1:]
return aminoacid

.....
coding strand = input("Input coding strand:")
i = len(coding strand)
if i < 5:
    print("strand too short")
    exit(0)
if (coding strand [0] != '3') and (coding strand [0] != '5'):
    print("Incorrect first character")
    exit(0)
if (coding strand [i-1] != '3') and (coding strand[i-1] != '5'):
    print("Incorrect last character")
    exit(0)
if coding strand [0] == coding strand[i-1]:
    print("Illegal match between first and last character")
    exit(0)
for j in range (0,i):
    if coding strand [j] not in znaky:
        print("Nesprávny znak v retazci: "+ coding strand[j])
        exit(0)
working strand = create_ working strand (coding strand)
print("working strand DNA:", working strand)
mrna = create_mrna(working strand)
print("mRNA:", mrna)
otocene = otocenie(mrna)
```

```
print("correctly oriented strand", otocene) v  
formed_peptid= create_peptid(mrna)  
print("formed aminoacid / peptid",formed_peptid)
```

4. CONCLUSIONS

The subsequent analysis of the feedback provided by teachers concerning the methodologies focused on the creation of models in Chemistry, more specifically the Biosynthesis of Nucleic Acids and Proteins: Genetic Code topic revealed that the following features were seen as advantageous: the possibility of using the methodologies to establish knowledge and the interconnection between Chemistry and IT studies. The teachers noted that the content complemented the thematic unit in an engaging manner and reported that students greatly appreciated the use of the program in their studies.

The methodologies use the method of active learning which results in the development of competencies needed for the job market of the 21st century and the future employment prospects of students such as digital competencies, scientific competencies and competencies connected with thinking and learning. The topic was also found to promote students' interest in Chemistry as a scientific subject.

ACKNOWLEDGEMENTS

This research was carried out within the framework of the "IT Academy – Education for the 21st Century" national project and was supported by the European Social Fund and the European Regional Development Fund in the framework of the Operational Program Human Resources (ITMS code: 312011F057), VEGA Project No. 1/0037/22, 1/0265/17 and KEGA Project No. 004UPJŠ-4-2020.

REFERENCES

1. Ganajová, M & Kozurkova M 2021, "The implementation of a new subject "Informatics in Chemistry" for Slovak secondary schools aimed at Informatics", *INTED 2021 : 15th International Technology, Education and Development Conference* : conference proceedings. - Valencia : IATED, pp. 5276-5282.
2. Ganajová, M et al. 2021, "Zbierka inovatívnych metodík z chémie pre stredné školy". 2nd ed., Bratislava, Centrum vedecko-technických informácií SR, pp. 941, <<https://vzdelavanie.itakademia.sk/vystupy/zim-che-ss.pdf>>.
3. Lesson Plan Coaches, Built with GeneratePress. <<https://lessonplancoaches.com/genetic-code-form-sc-botony/>>
4. Koonin, EV & Novozhilov AS 2012, "Origin and evolution of the genetic code: the universal enigma". *IUBMB Life*, vol. 61, no 2, pp. 99-111.
5. Saier, MH Jr. 2019, "Understanding of genetic code", *Journal of Bacteriology*, vol. 201, no. 15.
6. Python, Computer programming for everybody. <<https://www.python.org/doc/essays/cp4e/>>
7. Ayer, VM 2014, "Why scientist should learn to program in Python". *Powder distraction*, vol. 39, S48-D64.
8. Kmeťová at al. 2011, "Chémia pre 3. ročník gymnázia so štvorročným štúdiom a 7. ročník gymnázia s osemročným štúdiom", Vydavateľstvo Matice slovenskej s.r.o.
9. Genetika – Biologie "Příklady z molekulární genetiky". Viewed 07 May 2018, <<http://www.genetikabiologie.cz/priklady-z-genetiky/molekularni-genetika/>>.