Teaching Genomics and Bioinformatics to Undergraduates using Java

Andreas Spanias, Niranjan Chakravarthy, Yu Song, Leon D. Iasemidis[+]
Department of Electrical Engineering, [+]Department of Bioengineering,
Fulton School of Engineering
Arizona State University, Tempe, AZ 85287

Abstract

In this paper, we describe Java software that can be used to expose undergraduates to genomics. The content and software developed provide multidisciplinary knowledge to undergraduates in signal processing, genetics, and statistics. DNA, which is the fundamental storehouse of genetic information, is a linear polymer formed from four sub-units called nucleotides. The Java software is embedded in the ASU J-DSP visual programming environment. We have developed Java-DSP modules to present basic concepts about nucleotide character sequences. Specific J-DSP functions to analyze DNA sequences include: Numerical Mapping, FFT Power Spectrum, and Amino Acid Sequencing. Using the Numerical Mapping function, nucleotide sequences can be transformed into the numerical domain through binary, complex or integer number mappings. The FFT Power Spectrum block is used to compute the power spectrum of mapped sequences, and further classify them as belonging to either protein coding or non-coding regions. The above J-DSP blocks can also be integrated to form an internet visualization tool to identify genes in unannotated DNA sequences. A typical scenario is for a student to read a high-level DNA tutorial and execute our Java simulations. A series of learning modules and accompanying web-based exercises have also been developed. Exercises include: DNA fundamentals, numerical mapping of nucleotides, DNA power spectrum computation, and amino acid sequencing. The software and hands-on exercises have been assigned and assessed in the undergraduate DSP class at ASU.

1. Introduction

Research in genomics is expected to provide information that will lead to the prevention and cure of many diseases. Recent findings on DNA (Deoxyribonucleic Acid) sequences and microarrays provide great promise in this direction. Bioinformatics research involves contribution from a number of allied fields such as genetics, statistics, signal processing etc. Consequently, it is necessary to develop educational tools to introduce these concepts to undergraduates. ASU researchers developed an exemplary laboratory tool for use in undergraduate courses such as Digital Signal Processing (DSP) and Bioinformatics, to introduce students to recent research trends in genomic signal processing. In conjunction with the previously developed J-DSP* signal processing suite[7-10], the bioinformatics modules help provide multidisciplinary knowledge to undergraduates in signal processing, genetics and bioinformatics.

In this paper, we describe the bioinformatics Java modules we have developed to introduce basic concepts of DNA sequence analysis. These include a DNA sequence input module, numerical mapping module, FFT-based power spectral analysis tool and an amino-acid sequencing module. This paper is organized as follows. The biological basics of DNA sequences are presented in section 2 along with details about the DNA sequence input block. In section 3, the numerical mapping module is presented, followed by the discrete Fourier transform-based DNA sequence power spectral analysis module in section 4. The amino acid sequencing module is explained in section 5, and proposed future work is presented in section 6.

2. Biological Basics of the DNA and the DNA sequence input block

In living organisms, various cell activities are carried out by proteins. The information to produce proteins is stored in a certain specialized biochemical known as the DNA. The DNA is a sequence of four sub-units, known as *nucleotides*. The nucleotides, which are represented by the characters A, C, G and T, are bound together by strong chemical bonds to form DNA strands. Typically, two DNA strands known as *complementary* strands are bound to each other by weak hydrogen bonds. Such binding takes place only between an A in one strand with T in the other, and C in on strand with G in the other. Figure 1 shows an example of complementary DNA sequences. Due to the chemical structure of the nucleotides, DNA sequences have an inherent

* Work on J-DSP has been funded in part by NSF-CCLI-DUE-0089075 project

directionality. The convention is to 'read' a DNA sequence from the 5' end to the 3' end (see Figure 1). In living cells, DNA double strands typically exist as double helical structures.

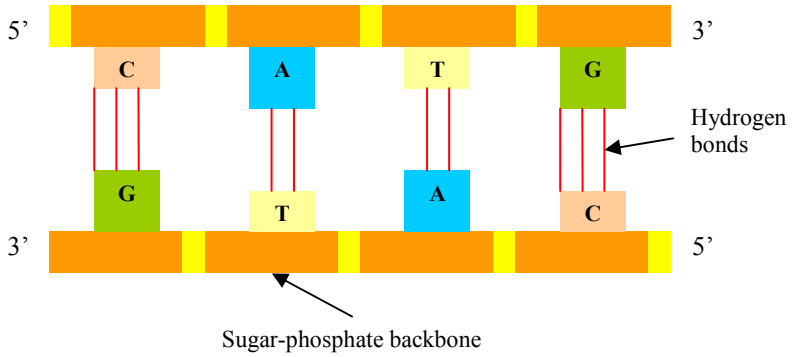


Figure 1: Double Stranded DNA

The actual information for the formation of proteins is contained in certain regions of the DNA sequence known as *protein-coding regions* or *genes*. The primary functionality of coding regions is to store information for protein formation. This information is stored in the sequence of nucleotide triplets are known as *codons*. For example, in the gene DNA sequence 5'-ATGCGATAC-3', the codons are ATG, CGA and TAC. Protein production based on the information in coding regions takes place in two steps known as *transcription* and *translation*. In the transcription stage, the information in the gene is copied onto another biochemical known as the Ribonucleic Acid (RNA), which is made up of 4 subunits indicated by A, C, G and U. During transcription, the RNA sequence is formed as the complement of the gene. For example, the DNA sequence 5'-ATGC-3' is copied as 5'-GCAU-3' in the RNA. The next step in the protein formation process is known as translation. During translation, codons from RNA are 'read' one-by-one, and molecules known as amino acids corresponding to each codon are formed. The amino acid molecules bind together to form proteins. Thus, information from coding DNA regions is used to generate proteins. The actual process involves chemical reactions among a number of other specialized biochemicals such as enzymes[1].

The DNA sequence module is used to input nucleotide sequences. This block can be opened by first starting the J-DSP editor, then placing it in the J-DSP simulation area, and finally double-clicking the block. Figure 2 shows a view of the DNA sequence block and J-DSP

environment. Next, the required DNA sequence can be typed in directly, or copied from another source. The output of this block is the sequence of 4 characters, i.e., a symbolic sequence. The next functionality developed to analyze DNA sequences involves mapping or 'quantizing' the 4 nucleotide characters to obtain an equivalent numerical sequence. Details regarding the numerical mapping block are described next.

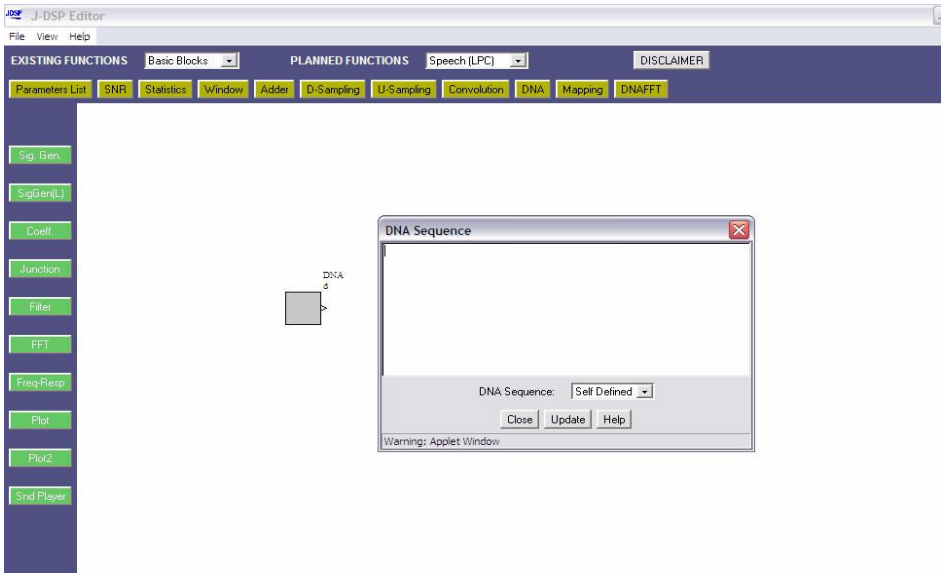


Figure 2: DNA sequence block

3. Numerical Mapping of DNA sequences

DNA sequences can be considered as discrete symbolic or categorical signals. Digital signal processing (DSP) methods are typically applicable to numerical signals. In order to apply DSP-based techniques for DNA sequence analysis, the 4 nucleotide characters are assigned numbers. In other words, the nucleotide character sequence is mapped into the number domain. Various DSP-based DNA sequence analysis are based on numerical mappings such as binary[2], integer[3] and complex number mapping[4,6]. Binary mappings have been typically used for discrete Fourier transform (DFT)[5] and correlation function-based DNA sequence analysis[2]. Specifically for DFT-based analysis, *indicator sequences* corresponding to A, C, G and T are used. An indicator sequence corresponding to a particular nucleotide is obtained by assigning 1 wherever that nucleotide occurs, and 0 elsewhere. For instance, for the DNA sequence 5'-ATCGGCTA-3' the binary indicator sequence corresponding to A is: 5'-10000001-3', the indicator sequence

corresponding to G is 5'-00011000-3' and so on. The DFT power spectrum of a DNA sequence is computed by the summing the power spectra of the four indicator sequences corresponding to A, C, G and T. Other DSP-based analyses such as amino acid sequencing have been carried out using complex number mapping rules[4]. The DNA sequence block can be connected to the numerical mapping block, to perform numerical mapping of a given nucleotide sequence. Currently, the type of numerical mapping can be chosen from Binary Indicator, Integer (A:1,C:2,G:3,T:4) and Complex number (A:1+j,C:-1-j,G:-1+j,T:1-j). The output of the numerical mapping block is a numerical sequence, which can be further analyzed. Figure 3 shows the numerical mapping block while using the binary indicator mapping. Note that the equivalent numerical sequences can also be viewed. The numerical mapping block thus acts as the second module for DSP-based DNA sequence analysis. Next, details about the DFT based analysis are presented.

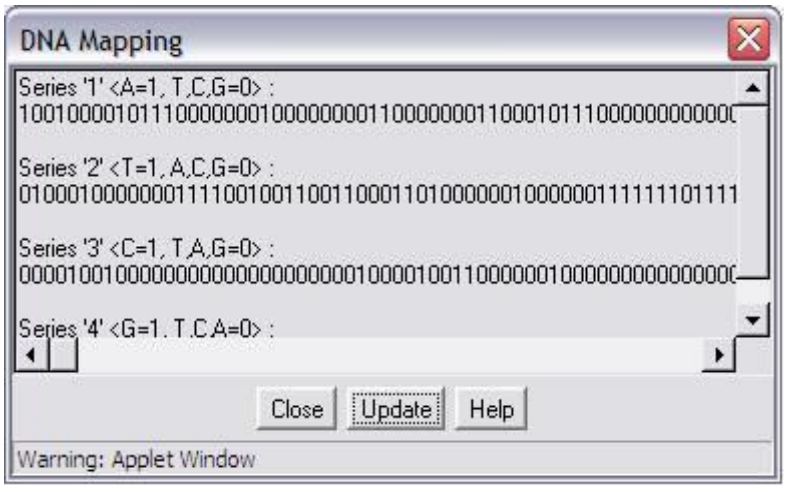
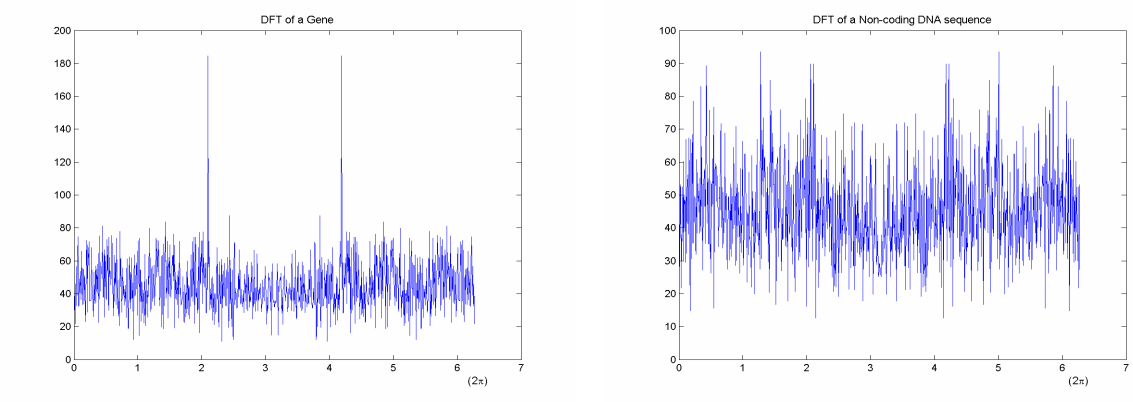


Figure 3: Binary indicator sequence mapping using the DNA numerical mapping block

4. DFT-based DNA sequence analysis

DNA sequences from coding and on-coding regions have different power spectral characteristics[5]. The power spectrum of DNA sequences is typically calculated by first mapping them into the numerical domain to obtain binary indicator sequences. Next, the DFT of each of these indicator sequences are computed and summed to obtain the power spectrum of the analyzed DNA sequence. Coding region sequences typically exhibit a characteristic spectral peak at $2\pi/3$, whereas non-coding region DNA sequences do not exhibit such a spectral signature (see

Figure 4). This difference in spectral characteristics between coding and non-coding regions has also been used to predict the location of genes in unannotated DNA sequences[5].



(a) (b)

Figure 4: DFT of a coding (a) and non-coding (b) DNA sequence

We have developed a Java module to compute the DFT spectra of DNA sequences. The Fast Fourier Transform (FFT) algorithm is used to implement the DFT. The inputs to this DNA-FFT block are the 4 binary indicator sequences which are output from the numerical mapping block. From the power spectra of the 4 indicator sequences, the power spectrum of the DNA sequence is calculated. The magnitude of the power spectrum is output from the DNA-FFT block, which can be plotted using the J-DSP Plot module. The flowgram for the above mentioned DFT-based DNA sequence analysis is shown in Figure 5. The power spectra of two DNA sequences are shown in Figure 6 and Figure 7. In the first case, the DNA sequence of a gene was analyzed. The gene's DNA sequence and its corresponding power spectrum are shown in Figure 6. In the second example, the DNA sequence from a non-coding region was analyzed. The power spectrum is shown in Figure 7. No characteristic spectral peak is observed in this case. Next, details about the stand-alone module for amino acid sequencing are presented.

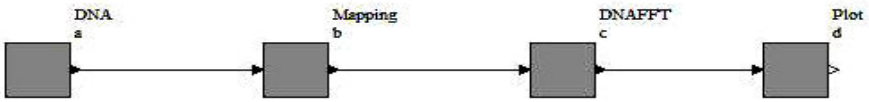


Figure 5: Flowgram for DFT-based DNA sequence analysis in J-DSP

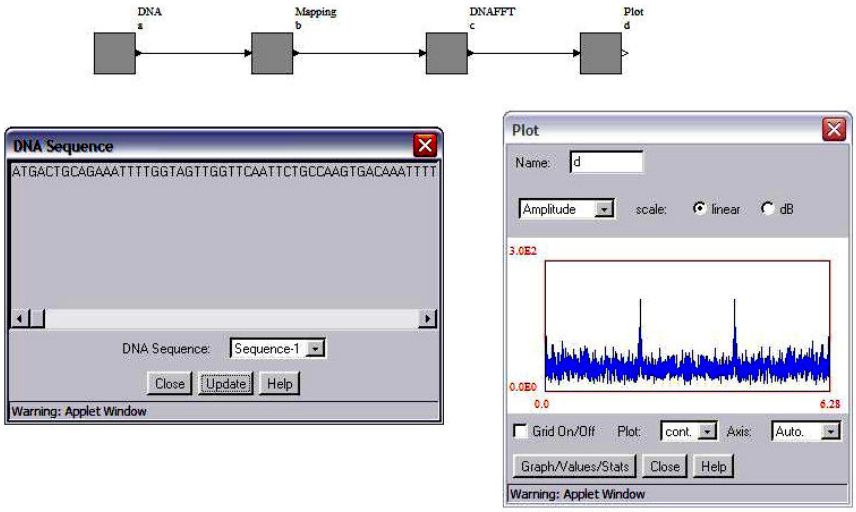


Figure 6: DFT of a gene in J-DSP

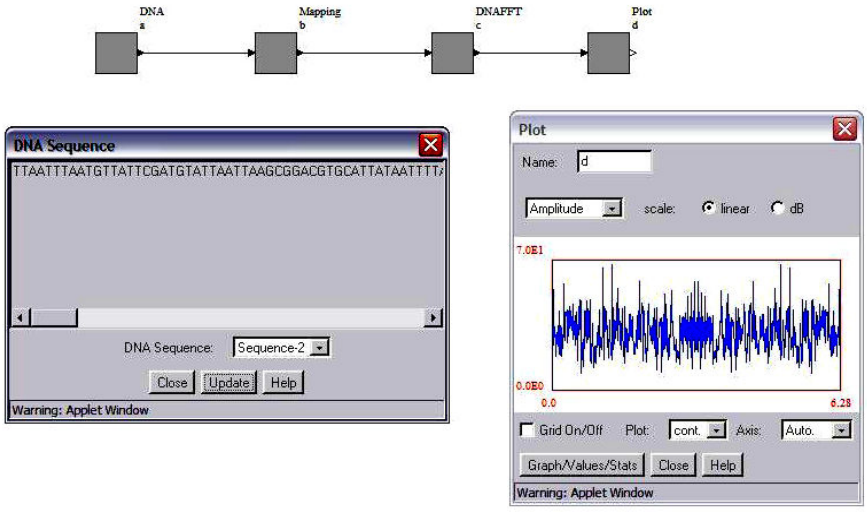


Figure 7: DFT of a non-coding DNA sequence in J-DSP

5. Amino acid sequencing

DSP concepts such as FIR filtering and VQ have also been used in DNA sequence analyses such as amino acid sequencing[4]. As mentioned in section 2, a sequence of amino acid molecules forms a protein. In turn, amino acids are formed based on the information in genes. In other words, codons in a gene are mapped onto corresponding amino acid molecules through the processes of transcription and translation. The mapping between codons and amino acids is

depicted in Table 1. Note that the same amino acid can be formed from different codons in a gene (in other words, a many to one mapping).

Given a gene nucleotide sequence, it is possible to obtain the equivalent amino acid sequence using a third order digital FIR filter and vector quantizer[5]. First, the gene sequence is mapped onto the numerical domain using the complex number mapping rule. This is input to a third order FIR filter which essentially computes a weighted average of a moving window (of length 3) over the gene sequence. The output of the FIR filter is then downsampled by three. This is done to obtain the weighted averages corresponding to the sequence of codons in the gene. The resulting signal is vector quantized to map the codons to their corresponding amino acids[5]. Figure 8 shows the amino acid sequencing module. It is possible to view the corresponding amino acid sequence as well as the DFT spectrum of the analyzed DNA sequence. The amino acid sequencing is performed by first entering the DNA sequence, then choosing the 'Gene FIR' option, and then updating the module. The corresponding amino acid sequence is displayed in the same window.

The aforementioned modules have been used to introduce concepts from DSP and bioinformatics, in the undergraduate DSP class at ASU. The typical scenario would be for the students to read through high-level tutorials explaining the basics of DNA sequences, numerical mapping, DFT, FIR filtering and VQ. Then, they are asked to analyze the spectral characteristics of DNA sequences, and classify the sequences as either coding or non-coding. The next step would be for the student to perform amino acid sequencing, to obtain the protein corresponding to a gene.

6. Future Work

This section provides suggestions for future work on J-DSP's bioinformatics module. The current modules for power spectrum based DNA sequence analysis can be extended to gene-finding[5]. This would involve computing the power spectrum of a moving window over a long unannotated DNA sequence, and classifying various regions as either coding or non-coding based on the power spectral characteristics. Gene-finding algorithms based on HMMs and other computational techniques can also be developed to introduce various gene-finding

methodologies. The amino acid sequencing module can be generalized for other numerical mappings as well. This would involve designing an FIR filter and vector quantizer to map the codons into amino acids, based on the number mapping used. Such modules can also enhance the learning of related topics such as FIR digital filter design and VQ.

Recent DNA sequence analysis methodologies are based upon a number of DSP concepts such as correlations, linear prediction (LP) time-frequency representations (TFR). J-DSP contains a rich suite of signal processing modules for functions such as correlations, LP, TFR etc., which can be used to introduce recent trends in genomic signal processing research. Finally, modules to introduce gene-expression concepts are being developed. Gene-expression analysis using techniques such as microarrays, has been reported to be highly effective for understanding various cell mechanisms. J-DSP modules are being developed to introduce the basics of microarray analysis and related concepts such as reverse transcription and hybridization.

1st Position (5' end)	2nd Position				3rd Position (3' end)
	U	C	A	G	
U	Phe	Ser	Tyr	Cys	U
	Phe	Ser	Tyr	Cys	C
	Leu	Ser	STOP	STOP	A
	Leu	Ser	STOP	Trp	G
C	Leu	Pro	His	Arg	U
	Leu	Pro	His	Arg	C
	Leu	Pro	Gln	Arg	A
	Leu	Pro	Gln	Arg	G
A	Ile	Thr	Asn	Ser	U
	Ile	Thr	Asn	Ser	C
	Ile	Thr	Lys	Arg	A
	Met	Thr	Lys	Arg	G
G	Val	Ala	Asp	Gly	U
	Val	Ala	Asp	Gly	C
	Val	Ala	Glu	Gly	A
	Val	Ala	Glu	Gly	G

Table 1: The genetic code mapping codons to amino acids

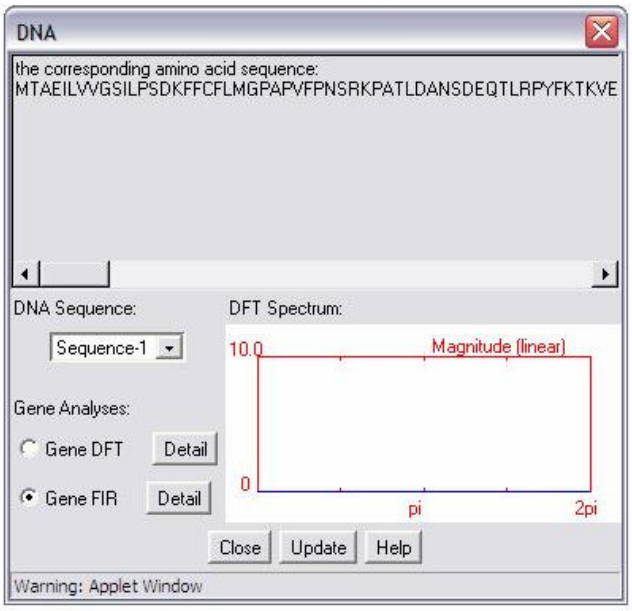


Figure 8: Amino acid sequencing to obtain the amino acid corresponding to a gene DNA sequence

References

1. B. Alberts, D. Bray, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *Essential Cell Biology*. New York: Garland Publishing, 1998.

2. P. Bernaola-Galvan, P.Carpena, R.Roman-Roldan and J.L. Oliver, "Study of statistical correlations in DNA sequences" in *Gene*, vol. 300, pp.105-115, 2002.

3. A.A.Tsonis, , J.B.Elsner, and P.A.Tsonis, "Periodicity in DNA coding sequences: Implications in gene evolution" in *J. Theor. Biol.* vol.151, pp.323-331, 1991.

4. D. Anastassiou, "Genomic Signal Processing" in *IEEE Signal Processing Magazine*, vol. 18, No. 4, pp. 8-10, July 2001.

5. S. Tiwari, S. Ramachandran, A. Bhattacharya, S. Bhattacharya, and R. Ramaswamy, "Prediction of probable genes by Fourier analysis of genomic sequences" in *CABIOS*, vol. 113, pp. 263-270, 1997.

6. Niranjan Chakravarthy, A.Spanias, L.D.Iasemidis, K.Tsakalis, "Autoregressive modeling and parametric analysis of DNA sequences" in *EURASIP Special Issue on Genomic Signal Processing*, To appear in 2004

7. A. Spanias, *et al.*, "On-line laboratories for speech and image processing and for communication Systems Using J-DSP", in *2nd DSP-Education workshop*, Pine Mountain GA, Oct 13-16, 2002.

8. A. Spanias, K. Ahmed, A. Papandreou-Suppappola, and M. Zaman, "Assessment of the Java-DSP (J-DSP) On-Line Laboratory Software," in *33rd ASEE/IEEE FIE-03*, Boulder, Nov. 2003.

9. T. Thrasyvoulou, K. Tsakalis, and A. Spanias, "J-DSP-C, A control systems simulation environment for distance learning: labs and assessment," in *33rd ASEE/IEEE FIE-03*, Boulder, Nov. 2003

10. V. Atti and A. Spanias, "On-line simulation modules for teaching speech and audio compression," in *33rd ASEE/IEEE FIE-03*, Boulder, Nov. 2003