CE 552 Final Project

Project Name: Helixpace – DNA Data Storage

Team Members: Hanyu Wang, Ji Young Chung, Jinan Qin, Sherinx Li

Project Description

As the demand for data storage increases, there is a need to find more storage space and more stable storage methods, and DNA storage is a popular area that is being developed now because of its dense data volume and longer duration time. In this project, we tried to convert information in DICOM (Digital Imaging and Communications in Medicine) files, a commonly used image format to store medical images and patient information, to a DNA sequence and to encode the DNA sequence back to images and text information such as patient information.

This project contains two major functions: encoding and decoding. In the encode part, information will be encoded to a DNA sequence. Users can set up their username and password, input extra information if they want, select the dcm files they would like to store. Our program will create a private primer based on the username and password, encode the patient information contained in the dcm files, encode the images of the dcm files, encode extra information, get the public primer, and connect everything together to output a DNA sequence in a .txt file. In the decode part, the users need to select the DNA sequence file that they want to decode and input their username and password to check whether they are allowed to decode the DNA sequence. The program will decode the DNA sequence and convert the information back into images and text including the patient information and extra information.

Detailed Functions

1. Encode (class DNAEncoder)

a. Load .dcm files (load_dcm)

Users can select the .dcm files they would like to load and encode into DNA files. The .dcm files will be read and saved in the system.

b. Private primer

primerencode(ID, PW)
primercheck_structure(seq,DNA_image,information_DNA,DNA_hos_primer)
primercheck_Tm(seq,DNA_hos_primer)

primercheck(seq)

Private primer is generated as the first 100bps of the compiled sequence. In order to provide uniqueness, it is generated based on the maximum 24 character string given from the user. Generated sequence has a maximum 96bps so the rest of the pairs until 100bps is filled with ‘A’s. Meanwhile, in order to confirm the generated sequence can function as a forward primer, melting point, G/C content percentage, and an identical duplication test is applied to the generated primer sequence.

c. Patient Information (encode_patient_info())

For the encode part, we store the patient information that we have gotten form .dcm file, such as patient ID, name, birthdate, age and sex. Then, we change the sting of patient information into ASCII code. After finishing transferring to ASCII code, we change them into quaternary code, which could combine with ATCG in DNA. Therefore, we could construct a guide that the string of information could be stored as ATCG in DNA.

d. Image (encode_image())

The images stored in .dcm files are a type of bitmaps that store the image pixels as numbers indicating grayscale at the pixel. First, the numerical pixel array of the loaded dcm files are extracted. The size of the 3D image is encoded into a 24-digit long DNA sequence. For each number in the array, it is converted into a 8-digit nucleotide sequence in order. You can consider this process as you convert a decimal number to a quaternary number, but the numbers 0, 1, 2, and 3 are replaced by A, T, C, and G, respectively. All the generated short DNA sequences are connected together to form the DNA sequence for image information.

e. Extra information (encode_extra_info())

The extra information entered in the window is first converted to ASCII information and the ASCII numbers are converted to DNA sequences following the rules above.

f. Get public primer and Encode everything (load_hospital_primer(), encode())

The program will read the public primer in the system (“hospital_primer.txt” in the example). The encode function will encode the private primer, patient information, image, and extra information, and connect everything with the public primer and output a DNA sequence file.

2. Decode

a. Divide the DNA sequence the decode each part (decode())

The first 100 nucleotides represent the private primer and the last 100 nucleotides represent the public primer. The first 8 nucleotides of the patient information part store the length of the patient information and the first 24 nucleotides of the image part store the size of the image. The rest sequence before the public primer is the extra information. The DNA sequence can be divided into smaller sequences and decoded into their corresponding information.

b. Private primer

decode_check(self,username, password, dna_file)

For security check, given username and password is encoded and compared to the first 100bps of the input DNA sequence. If match, further information decode is allowed.

c. Patient Information

This is similar to the encoding process. Firstly, we need to build a connection between the ATCG with the string of patient information. We need to let the DNA sequence which stores the patient information be changed into quaternary. Therefore, when we get the quaternary code, it could be easy for us to restore the patient information.

d. Image (encode_image())

The first three 8-nt long DNA sequences store the x, y, and z length of the 3D image. In the following DNA sequence, each 8 bits represents one data in one position of the array. The DNA can be decoded back into a number as a reverse process of the encoding process. You can think you convert quadrature numbers represented by A, T, C, and G to decimal numbers. The image can be reconstructed based on the image array that contains the grayscale of each pixel. The image array is stored and the images will be output into the output folder as .png files.

e. Extra information (encode_extra_info())

It is the reverse process of encoding. Each 4 bit in the DNA sequence containing the extra information can be converted back to a character. The 4 nucleotides are converted to a number first and converted to the character represented by this ASCII code. The decoded characters are connected as exported as a .txt file in the output folder.

3. GUI

The user can choose to encode, decode, or exit the program.

a. Encode

From the window, the users can set up the username and password, input extra information if they want, and select the dcm file or files they want to encode. The system will output a DNA sequence file.

b. Decode

The users need to input their username and password to check their identity. They can select the DNA sequence file to decode. The system will output a .txt file of patient information, a .txt file of extra information, and .png files of images in the output folder.