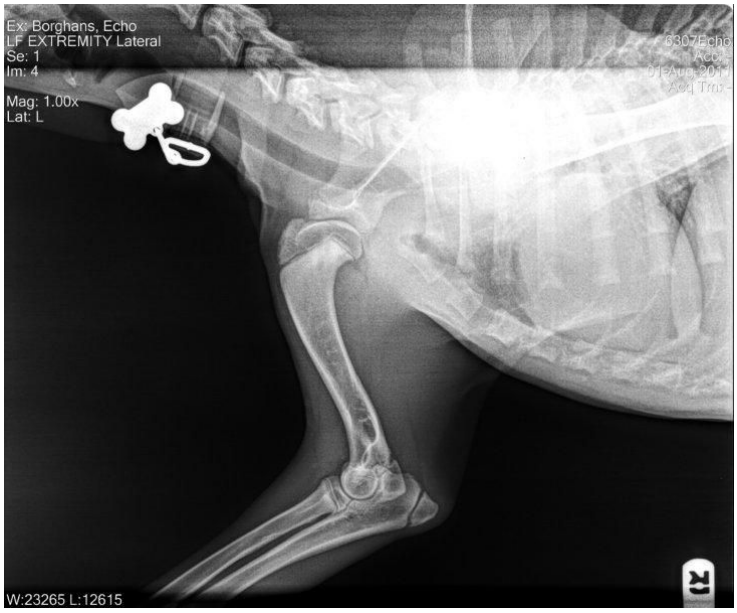
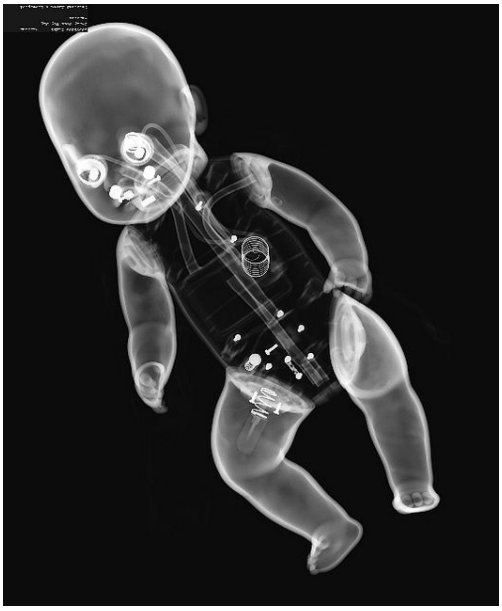
Homework 10

Due 4:00 PM, Wednesday, April 24, 2019





Introduction

After all this work, we now know that the true difference between puppies and babies lies within! You will develop a set of imaging tools to determine, once and for all, who is the best. Part I of this homework looks at genetic sequencing data and uses Regular Expressions to search through large DNA databases. Part II requires you to demonstrate your new skills in 2D and 3D data manipulation in MATLAB and uses the Image Processing Toolbox.

Please feel free to post to Piazza for additional support.

Installation and Handin

Homework Setup. To copy support files to your to your home directory for this homework type the following in a Brown CS terminal window:

```
cs4_install hw10
```

There should now be a `hw10` folder within your homeworks directory. Using Terminal, you can move into the `hw07` folder with the `cd` command:

```
cd ~/course/cs004/homeworks/hw10
```

Homework hand-in. Be sure to turn in all the files requested and that they are named exactly as specified, including spelling and case. When you're ready to submit the files, run:

```
cs4_handin hw10
```

from a Brown CS Terminal window from your `~/course/cs004/homeworks/hw10` directory. The entire contents of `~/course/cs004/homeworks/hw10` will be handed in. Check for a confirmation email to ensure that your assignment was correctly submitted using the `cs4_handin` command. You can resubmit this assignment using the `cs4_handin` command at any time, but be careful, as only your most recent submission will be graded.

Part I: DNA sequence patterns

Genome sequencing of biological deoxyribonucleic acid (DNA) forms the basis for modern technology in criminal forensics, anthropology, pathogenic bacteria evolution, cancer screening and treatment, genetic engineering, and biological cloning. The process of genome sequencing is conducted by extracting the order of nucleic acid molecules that make up the strands of DNA. These molecules are organized into long chains of four basic nucleotides: adenine, cytosine,

guanine, and thymine. In short, these nucleotides are represented in a string sequence using the letters A, C, G, and T. When analyzing DNA, proteins called restriction enzymes cut the DNA into shorter sequences. Restriction enzymes only cut at certain sites defined by a certain DNA sequence.

Problem 10.1a) dnaPattern (20)

For this question, you will be writing a function that takes in a filename and a regex pattern and provides the indices of all instances of that pattern. Place your solution in a file called `dnaPattern.m`

The function should have the following signature:

```
 [index]= dnaPattern(filename, pattern)
```

where `filename` and `pattern` are strings, and `index` is a 1xN row vector with the index of the start of each instance of the pattern until the Nth instance. To do this problem, you must use **`regexpi`.** Here is an example of what a run would look like, where sample text contains the sequence **CATATTTATTTACCATC**:

```
>>> dnaPattern('sample.txt', 'ATTTAC{2,3}')
ans =
       8
```

You will be running your code on the following 2 restriction enzymes with restriction pattern listed out. Elements in parentheses indicate optional elements that can be included in the restriction enzyme site. Character separated by commas in parentheses indicate it can be any of the characters within the parentheses. An N indicates that any nucleotide is allowed at that position:

ECORAI: NATCTAC(CC)

SALY: AT(A,G)NGTC(G)A

- Run the above program on the four samples (`sample1.txt, sample2.txt, sample3.txt, sample4.txt`*)* for *each* restriction enzyme.
- Report the total number of sites in a separate file titled `report.txt`.
- Find the length of the longest DNA fragment for each sample for each restriction enzyme. This length is equal to the largest distance from the start of a site to the beginning of the next site. Report this length as well.

You do not have to turn in any code for this part.

While we are **not** asking you to write test cases for your code, we will be comparing both the lengths of your reported indices as well as the actual value, so you should still test your code.

Hint: Consider using matlab's `fscanf` function. The linked website describes how to open and read text files.

Part 2: Medical Imaging

For Part 2, place all your solutions in the `medicalImagingVolume.m` file under the corresponding sections.

Problem 10.2a) Reading DICOM Formatted Images (5)

In this section, you'll be reading in all the DICOM formatted images we provide you with. First, use the `dir()` command to obtain a list of the files. For example, the below returns all dcm files in the folder within the cs0040 course directory, the path to which is provided for you in the stencil.

Then, for each file, use the `dicomread()` function, which takes in a file name, to read in the file data. Store the data as the third dimension in a 3 dimensional matrix named `volumeData`. Your code might look something like this for a single file:

```
filename = fullFile(filedir,files(n).name)
volumeData(:,:,1) = dicomread(filename);
```

Note: The data is located in the cs0040 course directory, so if you are working on a department machine, you won't need to change `filedir`. *If, however, you choose to work locally, you can unzip the datafile provided* (`data.tar.gz`) *using the following command from your* `hw10/data` *folder*:

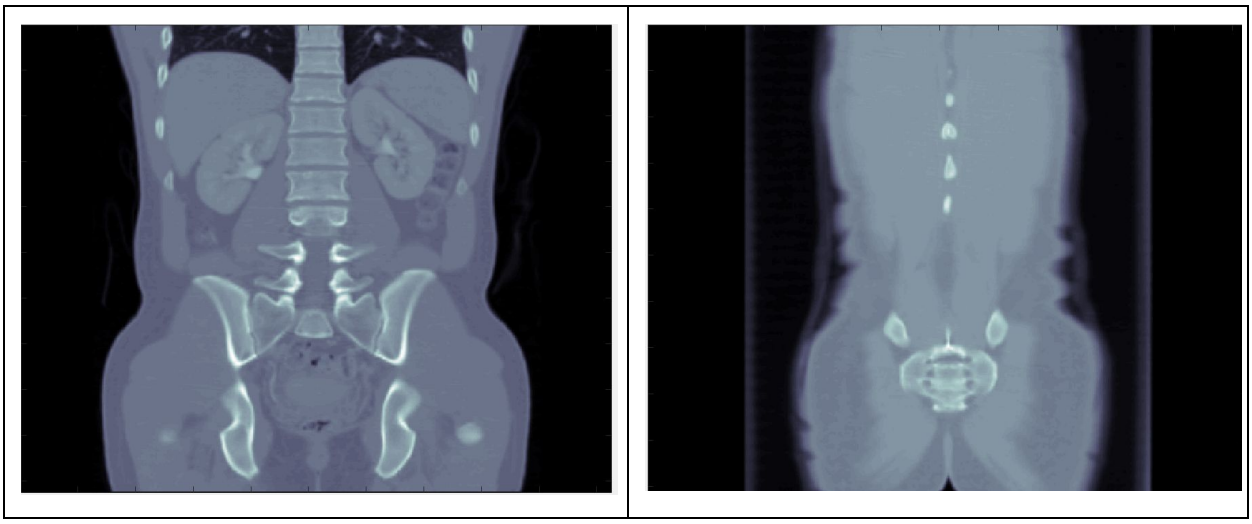
```
tar -xzvf data.tar.gz
```

Problem 10.2b) Plotting Slices (10)

To plot each data slice as an image, first remove all empty data volume.

Hint: You can use `find()` to do so!

Then, for each element in the third dimension of `volumeData`, plot the slice using `squeeze()` and `imagesc()`. Calling `colormap(bone)` should give you the proper color scheme. Title each with an appropriate label. When you run this, it should look like a short flipbook-like video. Each slice should look something like this:



Problem 10.2c) Plotting 3D Volumetric Data (15)

To plot the 3D volumetric data, you should first create `X`, `Y`, and `Z` coordinate vectors. You can use `dicominfo()` on a single file to get a struct of necessary measurements from that file. For example:

```
d = dicominfo(file.name)
```

This will store, in d, information on pixel spacing, height and width, and even some information on the patient and physician related to the CT scan.

`Z` should be a list of the same length as your files list where each element is `(index - 1)*d.SliceThickness`, meaning the first element should be 0.

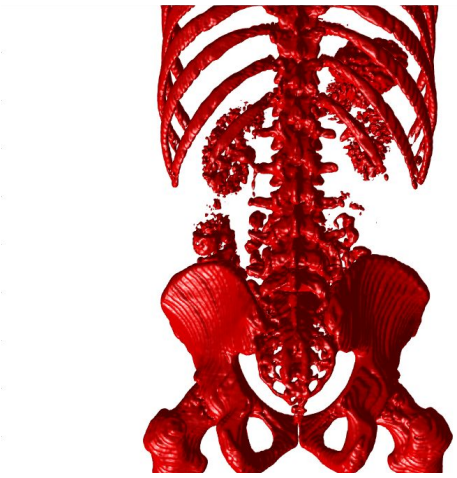
`X` should be a list of the same length as d.Height where each element is `(index - 1)*d.PixelSpacing(1).`

`Y` should be a list of the same length as d.Width where each element is `(index - 1)*d.PixelSpacing(2).`

This will allow you to plot each pixel along the X and Y direction of a given image segment. To actually plot this data, you will need `isosurface()`. 1200 should suffice for the isovalue. Afterwards, you must compute the isonormals to enhance your plot.

Take a look at the `patch()` and `isonormals()` functions discussed in lecture to figure out how to do this! Be sure to play around with lighting effects to get your final product to look like ours:

Here's a look at our finished product, though your colors could be slightly different:



Make sure all the m-files you want to submit are in your Brown CS `~/course/cs004/homeworks/hw10` directory. Be sure to turn in ALL the files requested and that they are named exactly as specified, including spelling and case. When you're ready to submit the files, run:

```
cs4_handin hw10
```

using a CIT computer terminal window. The entire contents of your `~/course/cs004/homeworks/hw10` directory will be handed in.

Please check that you receive an email confirming that your submission was successful. If you do not receive an email and have continued issues with the handin script, please contact the HTAs.

Please let us know if you find any mistakes, inconsistencies, or confusing language in this document or have any concerns about this and any other CS4 document by posting on Piazza or filling out our anonymous feedback form.