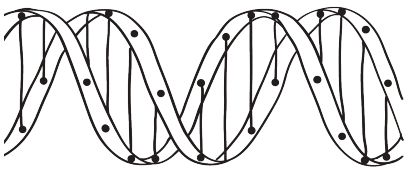


# A1: *The MEAN Central Dogma*

Julian Mazzitelli, *iGEM UofT*

May 14, 2015

Due: May 20, 2015 or May 22, 2015, or *anytime thereafter*

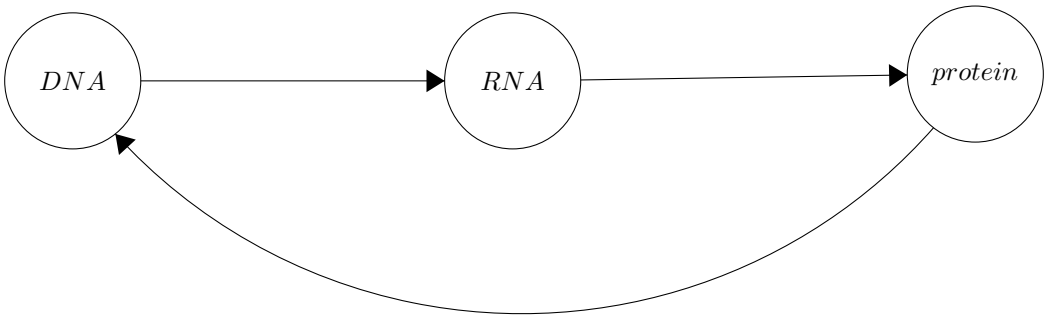
## 1 Introduction

This assignment will be an introduction to the modern full-stack web application while also promoting an improved understanding of the transfer of information within biological systems. You will build a single page application (SPA) with AngularJS which communicates with a RESTful Node.js (Express) *application programming interface* (API) and stores data on a MongoDB database. This is the MEAN stack:

$$\text{MongoDB} \leftrightarrow \text{Node.js (Express)} \leftrightarrow \text{AngularJS}$$

Moreover, you will write code in more than one language, interact with a number of frameworks/APIs, make a new best friend (documentation!), and consider the viability of your chosen data structures and algorithms for a given problem.

Your API will be capable of manipulating data forwards and backwards between each stage of biological information flow. You will not know what format the incoming data is, and will have to decide which conversion to use. That is, given DNA, RNA, or a polypeptide sequence, you must convert to each of the other two. Ultimately, you will describe the central dogma of molecular biology:




*Note* We will not consider DNA/DNA, RNA/RNA, or **protein/DNA** interactions in this assignment. Though you should realize the **crucial** implications of positive/negative regulation of DNA expression through protein/DNA interactions.

*Prerequisites* You **must** complete our Python DNA! assignment before starting this. You will call your python script from the back-end. You should also have completed javascripting, learnyounode, and expressworks. If you have trouble completing the DNA! assignment, consider trying out codecademy's Python tutorial.

## 2 Environment

It is necessary you have a proper development environment set up before beginning this assignment. You will need Python 3, Node.js and MongoDB. Your npm version should be up to date (Node.js comes with npm 2.7.4 but the latest is 2.9.something). Upgrading npm can be achieved with

```
$ npm install -g npm
```

If the previous does not upgrade your npm, it is most likely that you are on Windows and the path to Node.js occurs before the path to node_modules between your PATH and path environment variables (PATH loaded first). Once npm is up to date, you can install a few other global modules which you will be using:

```
$ npm install -g yo bower grunt-cli gulp
```

If you have issues starting a local MongoDB service, again, you are probably on Windows and what you need to do is

```
$ mkdir C:\data\db
```

and run (while you start `mongo` in another terminal)

```
$ mongodb -dbpath C:\data\db
```

If you have any issues getting set up, Google is your friend.

## 3 Requirements

Your app must:

- use a Python script to convert DNA → protein
- use JavaScript to convert protein → DNA
- accept data at endpoints of your choice
- provide endpoints to retreive data from database
- use unit testing with Mocha
- store converted data in the database

## 4 Boilerplates

You may find the following boilerplate generators useful.
The generator made by the Express team:
`$ npm install -g express-generator`
A port of the above to yeoman:
`$ npm insall -g generator-express`
A popular MEAN boilerplate:
`$ npm install -g generator-angular-fullstack`
mean.io's generator
`$ npm install -g mean-cli`
A front-end only AngularJS generator
`$ npm install -g generator-gulp-angular`

# 5 The MEAN Stack

The MEAN stack is the acronym given to a full-stack web server using Node.js as it's serverside engine, with Express as a framework for Node, using MongoDB for a database, and using the frontend framework AngularJS which eases the difficulty of making single page applications. We will go through a quick intro to each of these members, but first an understanding of how a web application works must be acquired.

A simple, bare bones website is just an `html` file, or series of `html` files. HTML stands for *hypertext markup language* and is very simple to understand. Here is a basic `helloworld.html` webpage:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Hello world!</title>
    </head>
    <body>
        <h1>HELLO world?</h1>
    </body>
</html>
```