Gene Info App/Visual Ideas

Back End Development

- Reads through gene list
- Uses site to search gene name
- Return site results
  - Instead of doing a ton of api calls, each ncbi gene has a "gene id". could get a list of gene ids and iterate over it to generate the ncbi gene link.
- Extracts direct link (ncbi) of definition or information
- Uses the direct link to get definition
- Extract the definition
- Add gene + definition + ncbi link to excel file

Front End Development

- Import excel file into widget (bokeh or R)
- Drop down menu for tiers of genes
- Make each gene clickable for popup information
- Popup information consists of link to ncbi, gene id, sequence information (as long as species is specified - homo sapiens would be best I suppose).

Sample Working Script

I created a sample script with some of the elements and examples. I’ve also been working with bokeh to create some sample interactive charts for different subsets of our data.

```python
# -*- coding: utf-8 -*-
"""
Created on Wed Dec 28 15:03:16 2016

@author: shutchins2
"""
# Modules Used
from googleapiclient.discovery import build
from pprint import pprint
from bs4 import BeautifulSoup as BS
from urllib.request import urlopen

# Use your google api key and project id
my_api_key = "AIzaSyDCmwhWbKpEcSKl7eVCAQ-6X0uV-Q48qLE"
my_cse_id = "011398459973079660664:noklcc-uegu"

def google_search(search_term, api_key, cse_id, **kwargs):
    service = build("customsearch", "v1", developerKey=api_key)
    res = service.cse().list(q=search_term, cx=cse_id, **kwargs).execute()
    return res['items']


# Search for the gene using a site and specify # of results using 'num='
results = google_search(
    'HTR1A site:https://www.ncbi.nlm.nih.gov/gene', my_api_key, my_cse_id, num=1)

# Write results to a file
with open('results.txt', 'w') as file1:
        for result in results:
            pprint(result, stream=file1)
#            pprint(result)

# Extract 'formattedUrl' from the results

# Use urllib2 to get the link and create an html file
ncbigene = 'https://www.ncbi.nlm.nih.gov/gene/3350'
html = urlopen(ncbigene)
soup = BS(html.read(), 'lxml')

# Print and save information
with open('html.txt', 'w', encoding='utf-8-sig') as file2:
    file2.write(soup.prettify())

# Extract the specified summary info or other information
#a = soup.find_all('dd')  # Find specific tags in the html file
#print(a)
#print(soup.get_text())  # Extract all text from the html page
#print(soup.find_all(attrs={"class": "section"})) # Gene Summary text
```