FlaGs-Viz minor code review

1. In order to ignore system files or folders (like *_pycashe_*) by git use .gitignore file. For ex. in order to remove *_pycashe_* include in this file a line: *_pycashe_*, also include '.*' line (to ignore all "hidden" files).
   So, your .gitignore file would be:

```
__pycashe__
.*
```

2. In the first script *app.py* a lot of spaces and empty lines. There are fixed rules about skipped lines
   For example:

```
def say_hi() -> str:
    """Say hi
    #1
    Returns:
        str: Hi
    #1
    """
    return("Hi")
#1
#2
def say_bye() -> str:
.....
```

3. There is also a problem with line breaks, there shouldn't be any code after 121 "column" (symbol), often in an IDE it's visualised as a vertical line. Almost in all IDEs there is a function to make auto-refactoring to fix such minor problems. The only problem it can't solve long str objects, for that there is syntax rule to make a line break, for ex:

```
text_var = f""It's a looooooooooooooooong liiiineeee"
            "neeeeee and that's how you move it to"
            "the next lines"
```

4. Sometimes there are unnecessary line breaks (for ex. this which could be only one line:). Another thing: use double quotes ("), not single (')

```
app.layout = html.Div([dcc.Location(id='url', refresh=False),
    html.Div(id='page-content')
])
```

5. So, i did small refactoring with your scripts and i'll send it back. It's really important to keep an eye on that since 1. it makes it more readable for you 2. it makes it more readable for reviewers (me) and (which is much more important) your article reviewers if they open it. 3. easier to update it later.

6. For me it's hard to comment this dash' way of buildsing an html. I would say that embedding a plotly output into the normal html flask template page could be more intuitive, because this syntax looks scary for me and Dash is not really for such things... Let's discuss it separately

7. ***home.py line 268 (and many others)***. Since your app located at server don't search for files through http/https, just parse the local folder.

8. ***home.py lines 357-363*** it's a bad thing to put smth before commenting function. For proper docstring see this manual (https://sphinxcontrib-napoleon.readthedocs.io/en/latest/example_google.html) (i recommend to use google's style)

9. ***home.py*** to be honest, it's quite impossible to review the part with plottly. The only thing i can say here: even if you're sure that it was written properly, try to test all cases manually whether everything is correctly plotted. This part can be done only by author.

Main part

1. *find your submission* somehow doesn't work..

2. *Upload your local files* works, but why the output figure aligned to the right?

3. Minor things about plots we can discuss a bit later, after we set up the app, but I see few problems, for example with overlapped genes (probably problem with stroke and fill values of objects).

4. **Key thing at this moment**, the way Dash operate with html is quite funky...and actually it wasn't created for such web apps.
   It's important to separate visualisation and calculations. Firstly, because it's more readable, but, more important because in a current version the app can stop responding after 5 requests sent within short time.
   My Q: how long would it take for you to simply write a function to get a plotly output? Actually it's written, just to separate. There is nothing special in html part, it seems your code mostly covers plotly. I can help with html template.

Setting up the app

Folder: /home/webapp/FlaGs-Viz

- `python3 -m venv .venv`
  `source .venv/bin/activate`
  `python3 -m pip install -r requirements.txt`
  `python3 -m pip install uwsgi flask`

- *wsgi.py* file:

```
from app import app

if __name__ == "__main__":
    app.run()
```

- *app.ini* file (server run on 5 threads):

```
    [uwsgi]
    module = wsgi:app

    master = true
    processes = 5

    socket = app.sock
    chmod-socket = 660
    vacuum = true

    die-on-term = true
    stats = /tmp/uwsgi.stats
```

- */etc/systemd/system/FlaGs-Viz.service*

```
    [Unit]
    Description=uWSGI instance to serve FlaGs-Viz
    After=network.target

    [Service]
    User=ar4622eg
    Group=www-data
    WorkingDirectory=/home/webapp/FlaGs-Viz
    Environment="PATH=/home/webapp/FlaGs-Viz/.venv/bin"
    ExecStart=/home/webapp/FlaGs-Viz/.venv/bin/uwsgi --ini app.ini

    [Install]
    WantedBy=multi-user.target
```

- `sudo chgrp www-data /home/webapp/FlaGs-Viz/`

- `sudo systemctl start FlaGs-Viz`
  `sudo systemctl enable FlaGs-Viz`

- `sudo systemctl status FlaGs-Viz`

```
● FlaGs-Viz.service - uWSGI instance to serve myproject
     Loaded: loaded (/etc/systemd/system/FlaGs-Viz.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2022-12-22 00:14:09 CET; 24min ago
   Main PID: 1511353 (uwsgi)
      Tasks: 6 (limit: 77043)
     Memory: 251.0M
     CGroup: /system.slice/FlaGs-Viz.service
             ├─1511353 /home/webapp/FlaGs-Viz/.venv/bin/uwsgi --ini app.ini
             ├─1511396 /home/webapp/FlaGs-Viz/.venv/bin/uwsgi --ini app.ini
             ├─1511397 /home/webapp/FlaGs-Viz/.venv/bin/uwsgi --ini app.ini
             ├─1511398 /home/webapp/FlaGs-Viz/.venv/bin/uwsgi --ini app.ini
             ├─1511399 /home/webapp/FlaGs-Viz/.venv/bin/uwsgi --ini app.ini
             └─1511400 /home/webapp/FlaGs-Viz/.venv/bin/uwsgi --ini app.ini
```

- */etc/nginx/sites-available/FlaGs-Viz.conf*

```
server {
        server_name go.atkinson-lab.com www.go.atkinson-lab.com;

        location / {
                include uwsgi_params;
                uwsgi_pass unix:/home/webapp/FlaGs-Viz/app.sock;
         }

    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/go.atkinson-lab.com/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/go.atkinson-lab.com/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

}
server {
    if ($host = go.atkinson-lab.com) {
        return 301 https://$host$request_uri;
    } # managed by Certbot


        listen 80;
        server_name go.atkinson-lab.com www.go.atkinson-lab.com;
    return 404; # managed by Certbot

}
```

- `sudo ln -s /etc/nginx/sites-available/FlaGs-Viz.conf /etc/nginx/sites-enabled/`
- `sudo systemctl restart nginx`