
AnnoTree Documentation

Kerrin Mendler, Han Chen, Andrew Doxey

Mar 27, 2019

1	Features	3
2	Example Workflows	5
3	AnnoTree with Custom Data	7
4	AnnoTree Production Version	9
5	AnnoTree Database Scripts	15
6	AnnoTree Backend	17
7	AnnoTree Frontend	23
8	AnnoTree Landing Page	25
9	Citing and Citations	27
10	Contributing	29
11	License	31

Caution: This documentation server is in an early stage of development.

CHAPTER 1

Features

CHAPTER 2

Example Workflows

AnnoTree with Custom Data

The use of custom taxonomic, phylogenetic, proteomic, or annotation data only affects the database preparation step. All other application components operate the same way as the production version of AnnoTree.

3.1 Prepare Database

Documentation for database preparation is located within the *AnnoTree Database Scripts* section. Keep track of the name of the new database for the backend configuration.

3.2 Configure Backend

Documentation for backend configuration is located within the *AnnoTree Backend* section. Keep track of the URL that the backend is being served to for the frontend configuration.

3.3 Configure Frontend

Documentation for the frontend configuration is located within the *AnnoTree Frontend* section.

.. setup-annotree_data:

4.1 Linux Server

- You need at least 150GB of disk space to properly load databases. We recommend 250GB.
- You should always use a **locally attached SSD** with $\geq 40\text{MB/s}$ IO speed for decent query time. Hard drives, network attached SSDs are proven to be too slow for some larger queries (can take 2 minutes). Some Cloud VMs by default will use network attached SSD and will not suit this task.
- It is recommended to create another user `annotree_user` and change user and group to that user: `chown -R annotree_user:annotree_user <this repository>`
- Downloading and loading database can take a long time, you might find it easier to run scripts related to them first
- It can take 30 minutes to 1 hour to follow all instructions (excluding loading time)

The following is a log of setting up a server on Feb 25, 2019, using (Ubuntu 16.04, with 2 core CPU, 8GB of RAM and 250GB of disk space):

Setting up MySQL: (If you are using **Google Cloud**, or other cloud services see bottom before proceeding to this section)

```
# download SQL dump, please refer to https://bitbucket.org/doxeylabcrew/annotree-
↪database/src/master/ for a list of URLs
# e.g. wget <my-dump-url>
sudo apt-get update && sudo apt-get install -y mysql-server mysql-client
# <enter root password>
# The following would take a while, we recommend you use "screen" cmd to avoid
↪terminal interruption
tar -vzxOf <path to .sql.tar.gz file> | mysql -u root -p --default-character-set=utf8
# give full permission to gtldb_user
mysql_username=annotree
mysql_password=<CHOOSE YOUR OWN PASSWORD>
echo "CREATE USER '$mysql_username'@'%' IDENTIFIED BY '$mysql_password';GRANT ALL
↪PRIVILEGES ON *.* TO '$mysql_username'@'%'; FLUSH PRIVILEGES;" | mysql -u root -p --
↪default-character-set=utf8
```

(continues on next page)

(continued from previous page)

```
# in case you want to save it
sudo echo $mysql_password > /root/mysql_annotree_password
```

Now log in to the database, for sanity check

```
mysql -u annotree -p
# <enter your password>
show databases;
## should show everything that's loaded, keep note of the database names
use <any of the database name, e.g. gtdb_bacteria>
show tables;
## should see a list of tables
SELECT COUNT(*) FROM node;
## should show the size of node table
SHOW INDEX FROM pfam_top_hits;
## should list pfam_id and gtdb_id indices, we encountered an issue in the past when
↳ disk space ran out in /tmp and indices were not loaded
```

Setting up server:

```
sudo apt-get update && apt-get install -y git python-pip libmysqlclient-dev python-
↳ dev build-essential
sudo mkdir -p /app
sudo useradd annotree_user
sudo passwd annotree_user
<enter password>
sudo mkhomedir_helper annotree_user
sudo chown -R annotree_user:annotree_user /app
sudo su - annotree_user
cd /app
git clone --branch latest-release --depth=1 https://bitbucket.org/doxeylabcrew/
↳ annotree-backend.git
cd annotree-backend
pip install -r requirements.txt
```

Now you can update config.py in backend

```
sudo su - annotree_user # make sure you are annotree_user, skip if you already are
cd /app/annotree-backend
cp sample-config.py config.py
vi config.py
## change mysql username and password
## change bacterial and archaeal database names as shown when you checked database
## normally they should be gtdb_bacteria and gtdb_archaea
## you may want to make sure config.py has secure permissions
chmod 440 config.py # this ensures only annotree_user and group annotree_user can
↳ read config.py
```

We will also show how to set up landing page and frontend here.

Landing page

```
sudo su - annotree_user # make sure you are annotree_user, skip if you already are
cd /app
git clone --depth=1 https://bitbucket.org/doxeylabcrew/annotree-landing-page.git
cd annotree-landing-page
# Check annotree-landing-page for newest set up instructions, what's recorded here
↳ may be outdated
```

(continues on next page)

(continued from previous page)

```

sudo easy_install nodeenv # switch to your own user if necessary
sudo su - annotree_user && cd /app/annotree-landing-page # do NOT use root from now on
nodeenv node6 --node=6.14.4 # this will be stuck if you use root
. node6/bin/activate
# you have node 6 now
node --version
# should say 6.14.4
npm install
node node_modules/gulp/bin/gulp.js
# the default job compiles all SCSS files and minifies javascript, you should be good
↪to go
sudo chown -R annotree_user:annotree_user /app/annotree-landing-page # fix permission
↪if you accidentally installed by root

```

Frontend

```

# use your own user account with sudo access
curl -sL https://deb.nodesource.com/setup_11.x | sudo -E bash -
sudo apt-get install -y nodejs
sudo su - annotree_user # make sure you are annotree_user, skip if you already are
cd /app
git clone --branch latest-release --depth=1 https://bitbucket.org/doxeylabcrew/
↪annotree-frontend.git
cd annotree-frontend
# Check frontend repo for specific instructions, the following instructions may be
↪outdated
npm install
cp app/Config.js.sample app/Config.js
vi app/Config.js
# change SERVER_BASE_URL to "http://<MY IP OR DOMAIN ADDRESS>/annotree-api"
npm run build
# you should see a public/ folder, this is where all html files are
# we will symlink this to inside the landing page
ln -s /app/annotree-frontend/public /app/annotree-landing-page/app

```

Then, to serve frontend and backend using apache WSGI module:

```

sudo apt-get install -y apache2 libapache2-mod-wsgi
sudo a2enmod wsgi

```

We will symlink annotree-landing-page

```

sudo ln -s /app/annotree-landing-page /var/www/html/annotree

```

Change apache config file using the following as an example:

```

sudo vi /etc/apache2/sites-available/000-default.conf
# replace with the following

```

```

Define annotree_backend_dir /app/annotree-backend

<VirtualHost *:80>
    ServerAdmin annotree_backend
    ServerName annotree_backend
    WSGIDaemonProcess dummy.com user=annotree_user group=annotree_user processes=2
    ↪threads=25

```

(continues on next page)

(continued from previous page)

```

WSGIScriptAlias /annotree-api ${annotree_backend_dir}/app.wsgi
DocumentRoot /var/www/html/annotree
<Directory ${annotree_backend_dir}>
    Options Indexes FollowSymLinks
    WSGIProcessGroup dummy.com
    WSGIApplicationGroup %{GLOBAL}
    Order deny,allow
    Allow from all
    Require all granted
</Directory>
ErrorLog ${APACHE_LOG_DIR}/annotree_error.log
CustomLog ${APACHE_LOG_DIR}/annotree_access.log combined
</VirtualHost>

```

Request to `http://example.com/annotree-api/gtdb_bacteria/tree` will be converted to `/gtdb_bacteria/tree` and sent to `app.wsgi`. You can modify API prefix by changing `WSGIScriptAlias` directive.

You will need to check frontend `app/Config.js` to make sure **api URL prefix matches**.

Enable changes and restart:

```
sudo service apache2 restart
```

Now check if everything is working:

```

curl localhost # you should see the landing page html
curl localhost/app # the main app
curl localhost/annotree-api/gtdb_bacteria/tree # check database and backend

```

4.2 Google Cloud

Google Cloud by default, uses network attached SSDs and will not satisfy our database access needs. We need to allocate a *local SSD*, which will be **removed** after instance is stopped (but not restarted). Extra caution is suggested. Database will reside on this drive instead of the default.

We will also need to make sure cloud service has the correct network configuration, this can be done by checking “Allow HTTP/Allow HTTPS” traffic in the VM instance EDIT tab (not included in the following instructions)

The following has been tested on Ubuntu 16.04 LTS:

Steps to set up database on Google Cloud

First go to https://cloud.google.com/compute/docs/disks/local-ssd#creating_a_local_ssd to allocate a SSD for your machine. Then run the following

```

# the following mounts up SSD drive, to /mnt/disks/ssd
lsblk # lists all attached drives, usually "sdb" is the local SSD
sudo mkfs.ext4 -F /dev/sdb
sudo mkdir -p /mnt/disks/ssd
sudo mount /dev/sdb /mnt/disks/ssd
sudo chmod a+w /mnt/disks/ssd

```

We will install MYSQL server then move to SSD


```

sudo apt-get update && sudo apt-get install -y mysql-server mysql-client
# ENTER "root" as password
sudo service mysql stop
sudo mv /var/lib/mysql /mnt/disks/ssd/mysql
sudo ln -s /mnt/disks/ssd/mysql /var/lib/mysql
# Edit config, change "tmpdir" from /tmp to /tmp/mysql
sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf
mkdir -p /mnt/disks/ssd/tmp-mysql
ln -s /mnt/disks/ssd/tmp-mysql /tmp/mysql
chmod a+rw /mnt/disks/ssd/tmp-mysql

```

Google Cloud uses apparmor to manage application read/write permissions

```

sudo echo "alias /var/lib/mysql/ -> /mnt/disks/ssd/mysql/, " >> /etc/apparmor.d/
↳tunables/alias
sudo vi /etc/apparmor.d/usr.sbin.mysqld

```

Add the following to /etc/apparmor.d/usr.sbin.mysqld; source from: <https://support.plesk.com/hc/en-us/articles/360004185293-Unable-to-start-MySQL-on-Ubuntu-AVC-apparmor-DENIED-operation-open->

```

/proc/*/status r,
/sys/devices/system/node/ r,
/sys/devices/system/node/node*/meminfo r,
/sys/devices/system/node/*/* r,
/sys/devices/system/node/* r,

/tmp/mysql/ r,
/tmp/mysql/** rwk,
/mnt/disks/ssd/tmp-mysql/ r,
/mnt/disks/ssd/tmp-mysql/** rwk,

/mnt/disks/ssd/mysql/ r,
/mnt/disks/ssd/mysql/** rwk,

```

```

sudo apparmor_parser -r /etc/apparmor.d/usr.sbin.mysqld
sudo service mysql start
echo "Sanity check to make sure everything is working"
echo "CREATE DATABASE temp; DROP DATABASE temp;" | mysql -u root -proot
# should say ok

```

Now MySQL is good to go, you can continue in previous session

If you are interested in testing speed, use the following command. (It runs a large PFAM query that would normally take ~40s to >1minute on other machines):

```

echo "SELECT node_id
FROM gtdb_node gn
JOIN
(
SELECT
gtdb_id,
COUNT(DISTINCT pfam_id) AS num_hit_per_genome
FROM pfam_top_hits
WHERE pfam_id IN ('PF00252') AND eval <= 1.0
GROUP BY gtdb_id
HAVING num_hit_per_genome >= 1
) g

```

(continues on next page)

(continued from previous page)

```
ON gn.gtdb_id = g.gtdb_id;" | mysql -u root -proot -D gtdb_bacteria_RS86
```

Updating

- Please check each repository (annotree-landing-page, annotree-frontend, annotree-backend) for possible instructions, what's listed here may not be up to date
- First pull newest code to each repository by running `git pull origin latest-release:latest-release`; or for landing page `git pull origin master`
- For database, run `mysql -u annotree -p` and `SHOW DATABASES;` in MySQL to check if all data are loaded properly, do not forget to update backend `config.py` in case of DB name change
- In case of server domain change, you need to change that in frontend: `vi /app/annotree-frontend/app/Config.js` to make sure new domain name matches.
- For backend, you may need to run `pip install -r requirements.txt` again, and `npm install` && `npm run build` for frontend; any change to frontend code must be followed by `npm run build` for it to compile.
- Finally check firewall settings, both on your local machine and on network (You need to allow incoming traffic on any cloud services)
- `run service apache2 restart service mysql restart` to bring up databases

AnnoTree Database Scripts

This repository contains all the scripts for creating a MySQL database that can be queried by the front-end AnnoTree application for the visualization of functional data on a phylogenetic tree. The scripts were written for implementation with data provided by the [GTDB](#) but any data files that use the same format may be used.

5.1 Installation Requirements

- You must be running a Unix-based operating system (ie. Linux or Mac)
- Install MySQL 5.7+
- Install Python 3.+
- Install required Python modules. To do this, download the `requirements.txt` file in the root of this repo and run the following:

```
pip install -r requirements.txt
```

5.2 Configuration YAML and Data Files:

Use `gtdb_database/test_data/db_config_example.yml` as a template and change the fields accordingly. It is suggested that this file be password-protected in order to hide the secure database information it contains. Consult the `gtdb_database/test_data` directory for example data files.

NOTE: All file paths must be full paths or be relative to the directory that you are running `make_db.py`.

5.3 Running Instructions

Once you have satisfied all of the installation requirements, all data is formatted correctly, and you have generated a configuration file, run the wrapper script: `python make_db.py --config path/to/config.yml` It will generate and populate a new MySQL database and output progress to the screen and `make_db.log`.

This is the backend part for [AnnoTree](#).

6.1 Install

Clone this repo, then `pip install -r requirements.txt`

Connection to database is in `config.py`. Create `config.py`, using `sample-config.py` as an example, for security purpose please turn off read access for other group: `chmod o-r config.py`

Development server and debug

`python app.py` to start development server. App is served on `localhost:5001`, you can change that in `app.run` line. Note, normally port 5001 is blocked by firewall, to allow remote access, do: `sudo iptables -I input -p tcp --dport 5001 -j ACCEPT`, BE CAREFUL THIS MAY OPEN SECURITY VULNERABILITIES

6.2 Production

- You need at least 150GB of disk space to properly load databases. We recommend 250GB.
- You should always use a **locally attached SSD** with $\geq 40\text{MB/s}$ IO speed for decent query time. Hard drives, network attached SSDs are proven to be too slow for some larger queries (can take 2 minutes). Some Cloud VMs by default will use network attached SSD and will not suit this task.
- It is recommended to create another user `annotree_user` and change user and group to that user: `chown -R annotree_user:annotree_user <this repository>`
- Downloading and loading database can take a long time, you might find it easier to run scripts related to them first
- It can take 30 minutes to 1 hour to follow all instructions (excluding loading time)

The following is a log of setting up a server on Feb 25, 2019, using (Ubuntu 16.04, with 2 core CPU, 8GB of RAM and 250GB of disk space):

Setting up MySQL: (If you are using **Google Cloud**, or other cloud services see bottom before proceeding to this section)

```
# download SQL dump, please refer to https://bitbucket.org/doxeylabcrew/annotree-
↳database/src/master/ for a list of URLs
# e.g. wget <my-dump-url>
sudo apt-get update && sudo apt-get install -y mysql-server mysql-client
# <enter root password>
# The following would take a while, we recommend you use "screen" cmd to avoid
↳terminal interruption
tar -vzxOf <path to .sql.tar.gz file> | mysql -u root -p --default-character-set=utf8
# give full permission to gtdb_user
mysql_username=annotree
mysql_password=<CHOOSE YOUR OWN PASSWORD>
echo "CREATE USER '$mysql_username'@'%' IDENTIFIED BY '$mysql_password';GRANT ALL
↳PRIVILEGES ON *.* TO '$mysql_username'@'%' ; FLUSH PRIVILEGES;" | mysql -u root -p --
↳default-character-set=utf8
# in case you want to save it
sudo echo $mysql_password > /root/mysql_annotree_password
```

Now log in to the database, for sanity check

```
mysql -u annotree -p
# <enter your password>
show databases;
## should show everything that's loaded, keep note of the database names
use <any of the database name, e.g. gtdb_bacteria>
show tables;
## should see a list of tables
SELECT COUNT(*) FROM node;
## should show the size of node table
SHOW INDEX FROM pfam_top_hits;
## should list pfam_id and gtdb_id indices, we encountered an issue in the past when
↳disk space ran out in /tmp and indices were not loaded
```

Setting up server:

```
sudo apt-get update && apt-get install -y git python-pip libmysqlclient-dev python-
↳dev build-essential
sudo mkdir -p /app
sudo useradd annotree_user
sudo passwd annotree_user
<enter password>
sudo mkhomedir_helper annotree_user
sudo chown -R annotree_user:annotree_user /app
sudo su - annotree_user
cd /app
git clone --branch latest-release --depth=1 https://bitbucket.org/doxeylabcrew/
↳annotree-backend.git
cd annotree-backend
pip install -r requirements.txt
```

Now you can update config.py in backend

```

sudo su - annotree_user # make sure you are annotree_user, skip if you already are
cd /app/annotree-backend
cp sample-config.py config.py
vi config.py
## change mysql username and password
## change bacterial and archaeal database names as shown when you checked database
## normally they should be gtdb_bacteria and gtdb_archaea
## you may want to make sure config.py has secure permissions
chmod 440 config.py # this ensures only annotree_user and group annotree_user can_
↳read config.py

```

We will also show how to set up landing page and frontend here.

Landing page

```

sudo su - annotree_user # make sure you are annotree_user, skip if you already are
cd /app
git clone --depth=1 https://bitbucket.org/doxeylabcrew/annotree-landing-page.git
cd annotree-landing-page
# Check annotree-landing-page for newest set up instructions, what's recorded here_
↳may be outdated
sudo easy_install nodeenv # switch to your own user if necessary
sudo su - annotree_user && cd /app/annotree-landing-page # do NOT use root from now on
nodeenv node6 --node=6.14.4 # this will be stuck if you use root
. node6/bin/activate
# you have node 6 now
node --version
# should say 6.14.4
npm install
node node_modules/gulp/bin/gulp.js
# the default job compiles all SCSS files and minifies javascript, you should be good_
↳to go
sudo chown -R annotree_user:annotree_user /app/annotree-landing-page # fix permission_
↳if you accidentally installed by root

```

Frontend

```

# use your own user account with sudo access
curl -sL https://deb.nodesource.com/setup_11.x | sudo -E bash -
sudo apt-get install -y nodejs
sudo su - annotree_user # make sure you are annotree_user, skip if you already are
cd /app
git clone --branch latest-release --depth=1 https://bitbucket.org/doxeylabcrew/
↳annotree-frontend.git
cd annotree-frontend
# Check frontend repo for specific instructions, the following instructions may be_
↳outdated
npm install
cp app/Config.js.sample app/Config.js
vi app/Config.js
# change SERVER_BASE_URL to "http://<MY IP OR DOMAIN ADRESS>/annotree-api"
npm run build
# you should see a public/ folder, this is where all html files are
# we will symlink this to inside the landing page
ln -s /app/annotree-frontend/public /app/annotree-landing-page/app

```

Then, to serve frontend and backend using apache WSGI module:

```
sudo apt-get install -y apache2 libapache2-mod-wsgi
sudo a2enmod wsgi
```

We will symlink annotree-landing-page

```
sudo ln -s /app/annotree-landing-page /var/www/html/annotree
```

Change apache config file using the following as an example:

```
sudo vi /etc/apache2/sites-available/000-default.conf
# replace with the following
```

```
Define annotree_backend_dir /app/annotree-backend

<VirtualHost *:80>
    ServerAdmin annotree_backend
    ServerName annotree_backend
    WSGIDaemonProcess dummy.com user=annotree_user group=annotree_user processes=2_
    threads=25proper permissions
    WSGIScriptAlias /annotree-api ${annotree_backend_dir}/app.wsgi
    DocumentRoot /var/www/html/annotree
    <Directory ${annotree_backend_dir}>
        Options Indexes FollowSymLinks
        WSGIProcessGroup dummy.com
        WSGIApplicationGroup %{GLOBAL}
        Order deny,allow
        Allow from all
        Require all granted
    </Directory>
    ErrorLog ${APACHE_LOG_DIR}/annotree_error.log
    CustomLog ${APACHE_LOG_DIR}/annotree_access.log combined
</VirtualHost>
```

Request to `http://example.com/annotree-api/gtdb_bacteria/tree` will be converted to `/gtdb_bacteria/tree` and sent to `app.wsgi`. You can modify API prefix by changing `WSGIScriptAlias` directive.

You will need to check frontend `app/Config.js` to make sure **api URL prefix matches**.

Enable changes and restart:

```
sudo service apache2 restart
```

Now check if everything is working:

```
curl localhost # you should see the landing page html
curl localhost/app # the main app
curl localhost/annotree-api/gtdb_bacteria/tree # check database and backend
```

6.3 Google Cloud

Google Cloud by default, uses network attached SSDs and will not satisfy our database access needs. We need to allocate a *local SSD*, which will be **removed** after instance is stopped (but not restarted). Extra caution is suggested. Database will reside on this drive instead of the default.

We will also need to make sure cloud service has the correct network configuration, this can be done by checking “Allow HTTP/Allow HTTPS” traffic in the VM instance EDIT tab (not included in the following instructions)

The following has been tested on Ubuntu 16.04 LTS:

Steps to set up database on Google Cloud

First go to https://cloud.google.com/compute/docs/disks/local-ssd#creating_a_local_ssd to allocate a SSD for your machine. Then run the following

```
# the following mounts up SSD drive, to /mnt/disks/ssd
lsblk # lists all attached drives, usually "sdb" is the local SSD
sudo mkfs.ext4 -F /dev/sdb
sudo mkdir -p /mnt/disks/ssd
sudo mount /dev/sdb /mnt/disks/ssd
sudo chmod a+w /mnt/disks/ssd
```

We will install MYSQL server then move to SSD

```
sudo apt-get update && sudo apt-get install -y mysql-server mysql-client
# ENTER "root" as password
sudo service mysql stop
sudo mv /var/lib/mysql /mnt/disks/ssd/mysql
sudo ln -s /mnt/disks/ssd/mysql /var/lib/mysql
# Edit config, change "tmpdir" from /tmp to /tmp/mysql
sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf
mkdir -p /mnt/disks/ssd/tmp-mysql
ln -s /mnt/disks/ssd/tmp-mysql /tmp/mysql
chmod a+rw /mnt/disks/ssd/tmp-mysql
```

Google Cloud uses app armor to manage application read/write permissions

```
sudo echo "alias /var/lib/mysql/ -> /mnt/disks/ssd/mysql/, " >> /etc/apparmor.d/
↳tunables/alias
sudo vi /etc/apparmor.d/usr.sbin.mysqld
```

Add the following to /etc/apparmor.d/usr.sbin.mysqld; source from: <https://support.plesk.com/hc/en-us/articles/360004185293-Unable-to-start-MySQL-on-Ubuntu-AVC-apparmor-DENIED-operation-open->

```
/proc/*/status r,
/sys/devices/system/node/ r,
/sys/devices/system/node/node*/meminfo r,
/sys/devices/system/node/*/* r,
/sys/devices/system/node/* r,

/tmp/mysql/ r,
/tmp/mysql/** rwk,
/mnt/disks/ssd/tmp-mysql/ r,
/mnt/disks/ssd/tmp-mysql/** rwk,

/mnt/disks/ssd/mysql/ r,
/mnt/disks/ssd/mysql/** rwk,
```

```
sudo apparmor_parser -r /etc/apparmor.d/usr.sbin.mysqld
sudo service mysql start
echo "Sanity check to make sure everything is working"
echo "CREATE DATABASE temp; DROP DATABASE temp;" | mysql -u root -proot
# should say ok
```

Now MySQL is good to go, you can continue in previous session

If you are interested in testing speed, use the following command. (It runs a large PFAM query that would normally take ~40s to >1minute on other machines):

```
echo "SELECT node_id
FROM gtdb_node gn
JOIN
(
SELECT
gtdb_id,
COUNT(DISTINCT pfam_id) AS num_hit_per_genome
FROM pfam_top_hits
WHERE pfam_id IN ('PF00252') AND eval <= 1.0
GROUP BY gtdb_id
HAVING num_hit_per_genome >= 1
) g
ON gn.gtdb_id = g.gtdb_id;" | mysql -u root -proot -D gtdb_bacteria_RS86
```

Updating

- Please check each repository (annotree-landing-page, annotree-frontend, annotree-backend) for possible instructions, what's listed here may not be up to date
- First pull newest code to each repository by running `git pull origin latest-release:latest-release;` or for landing page `git pull origin master`
- For database, run `mysql -u annotree -p` and `SHOW DATABASES;` in MySQL to check if all data are loaded properly, do not forget to update backend `config.py` in case of DB name change
- In case of server domain change, you need to change that in frontend: `vi /app/annotree-frontend/app/Config.js` to make sure new domain name matches.
- For backend, you may need to run `pip install -r requirements.txt` again, and `npm install` && `npm run build` for frontend; any change to frontend code must be followed by `npm run build` for it to compile.
- Finally check firewall settings, both on your local machine and on network (You need to allow incoming traffic on any cloud services)
- `run service apache2 restart service mysql restart` to bring up databases

6.4 Issues and contributing

Please feel free to open an issue on Bitbucket page for developers to review.

This is the frontend of AnnoTree, that is used to browse and explore GTDB microbial genome data. Full website is at: <http://annotree.uwaterloo.ca>

7.1 Installation

To create a standalone instance of AnnoTree that include frontend, backend, and database, it is recommended to use our [docker installer](#).

The following is for developer use.

Install

```
npm install
```

Set up app/Config.js

```
cp app/Config.js.sample app/Config.js
# You need to change parameters in Config.js
# if you have the backend in debug mode running in 10.123.45.78:5001, then change_
↪ SERVER_BASE_URL to "http://10.123.45.78:5001"
# NO TRAILING SLASHES
```

Start the application in development mode

```
npm run start
```

If `npm run start` failed, try switching node version to 6.x.

Open <http://localhost:8080> in your browser.

Build for production

```
npm run build
```

This will generate many `html.js` files in the `public` folder. You can point `Apache DocumentRoot` to `public` folder to serve web pages, or symlink `/var/www/html/annotree` to `public` folder

Here is a sample apache config:

```
<VirtualHost example.com:80>
    ServerAdmin webmaster@localhost
    # here we created a symlink /var/www/html/annotree that points to public folder
    DocumentRoot /var/www/html/
    # alternatively use DocumentRoot /var/www/html/annotree
    ErrorLog ${APACHE_LOG_DIR}/gtdb_frontend_error.log
    CustomLog ${APACHE_LOG_DIR}/gtdb_frontend_access.log combined
</VirtualHost>
```

Now you can visit `example.com/annotree` to see served pages. Please make sure that `app/Config.js` is pointed to the correct backend URL.

7.1.1 Issues and contributing

Please feel free to open an issue on Bitbucket page for developers to review.

CHAPTER 8

AnnoTree Landing Page

This project is based on [Freelancer landing page](#)

#Installing and Updating

Making sure you have node6

Gulp-sass, part of the build tool here works well with node 6.x

use `node --version` to check your version, if different, proceed to <https://github.com/ekalinin/nodeenv> to install nodeenv that customizes your node environment. Specifically:

```
sudo easy_install nodeenv
# do NOT use root user now, otherwise script will be stuck
nodeenv node6 --node=6.14.4
. node6/bin/activate
# you have node 6 now
node --version
# should say 6.14.4
npm install
node node_modules/gulp/bin/gulp.js
# the default job compiles all SCSS files and minifies javascript, you should be good.
↪to go
```

Running in dev mode, real time update in browser

```
node node_modules/gulp/bin/gulp.js dev
```

Making production files

```
node node_modules/gulp/bin/gulp.js dev
```

8.1 Linking with main app

Symlink in this folder `app/` to the `public/` folder in `annotree-frontend`, so that all requests to `<base url>/app/` gets redirected

8.2 Copyright and License

Copyright 2013-2018 Blackrock Digital LLC. Code released under the [MIT](#) license.

CHAPTER 9

Citing and Citations

CHAPTER 10

Contributing

CHAPTER 11

License
