

Interactive web tour of cBioPortal + Updated documentation for cBioPortal deployment

Final edit : April 2, 2017

GitHub repository: <https://github.com/pratikabhinav/GSoC-cBioPortal>

Abhinav Pratik

pratikabhinav@gmail.com

Overview

cBioPortal might seem like an extremely complicated website to someone visiting it for the first time. I had a similar experience, although I claim to have some technological background and understanding. And it was not until I went through the paper titled *"Integrative Analysis of Complex Cancer Genomics and Clinical Profiles Using the cBioPortal"*, that I was able to make out how things work on cBioportal.

But it would be asking too much of our visitors, if they have to read the paper first before being able to make use of the cancer genomics data.

The website as it stands is definitely not intuitive to a person browsing it for the first time, especially to someone who does not belong to a technical background. And to some extent, the complex nature of the website is such that it can not be completely intuitive. So what can be done so that people are able to use the analysis and visualization features of cBioPortal for Cancer Genomics, without having to go through the burden of reading a multi-page paper?

The solution lies in implementing an **Interactive web tour** across cBioPortal wherever needed, that guides users through the different features of the website and explains with an easy step-by-step procedure, how to use them. These are very useful to new visitors who are just learning the ropes of a website.

Goals

1. To make the cBioPortal more intuitive to use for the end-user.
2. To add interactive web tours to features of the website that need to be explained to the user in order to be made use of effectively.
3. The web tour should guide users through the different features of the website and explain with an easy step-by-step procedure how to use them, adding necessary details wherever needed.
4. A good starting point is to implement interactive web tours to the examples, *one at a time*, mentioned in the paper titled *"Integrative Analysis of Complex Cancer Genomics and Clinical Profiles Using the cBioPortal"* that provides a practical guide to the analysis and visualization features of the cBioPortal for Cancer Genomics.

This proposal includes an implementation of **"Performing a query of a single cancer study"** (*example 1 in the above mentioned paper*) as a proof of concept for this project.

How to implement the web tours?

There are a number of already existing Javascript based libraries that help achieve the goal of adding an interactive web tour to any website. And practically, we can work with any one of those, or use different scripts across the website depending upon the requirements.

However, based on the research I did, I found **Intro.js** to be really flexible and easy to implement. It should be the right fit for the functions we hope to achieve from our interactive tours.

Why Intro.js?

- No dependencies. Fast and small, 10KB JavaScript and 2.5KB CSS, that's all.
- User friendly navigation either through mouse or a keyboard.
- Works on Google Chrome, Mozilla Firefox, Opera, Safari and Internet Explorer.
- Open source (*GNU AGPLv3*).
- Unique features include adding *Hints*, *Multi-page tours*, *Floating guides that do not focus on elements*, and easy customization options.
- Well documented with code examples and demos.
- Will easily support new tours.
- Looks really elegant in the demos on their website introjs.com. It provides a nice page dimming effect. Should (*and does*) blend right in with cBioPortal. *Refer to the implemented example of "Performing a query of a single cancer study" in this document.*
- *While working with the portal's codebase, I noticed Intro.js's files (CSS and JSS) were already present in their respective folders, as per cBioPortal's structure. Someone in the cBioPortal team also thought using Intro.js was the right idea. This motivated me even more to move forward with Intro.js.*

How does Intro.js work?

Intro.js is an open source vanilla Javascript / CSS library to add step-by-step introduction or hints. It's basic implementation requires the following steps :

1. Add the JS and CSS file to the source code
2. Use `data-intro` in the HTML code of the element, to add a hint/instruction that should pop up pointing to it on the web page preview.
3. Use `data-step` in the HTML code to add a sequence to the hints/instructions.

Example:

data-step="3" data-intro="This third instruction comes after the second one"

data-step="2" data-intro="Check out this second instruction"

Anticipated challenges

- Adding Hints/Instructions to more complex pages such as the **Network tab** would be a challenging task. Since the result shown on these tabs, depends on a previous search query, it would be difficult to pick out exact HTML elements to add Intro.js code to, in order to facilitate the interactive tour.

Example :

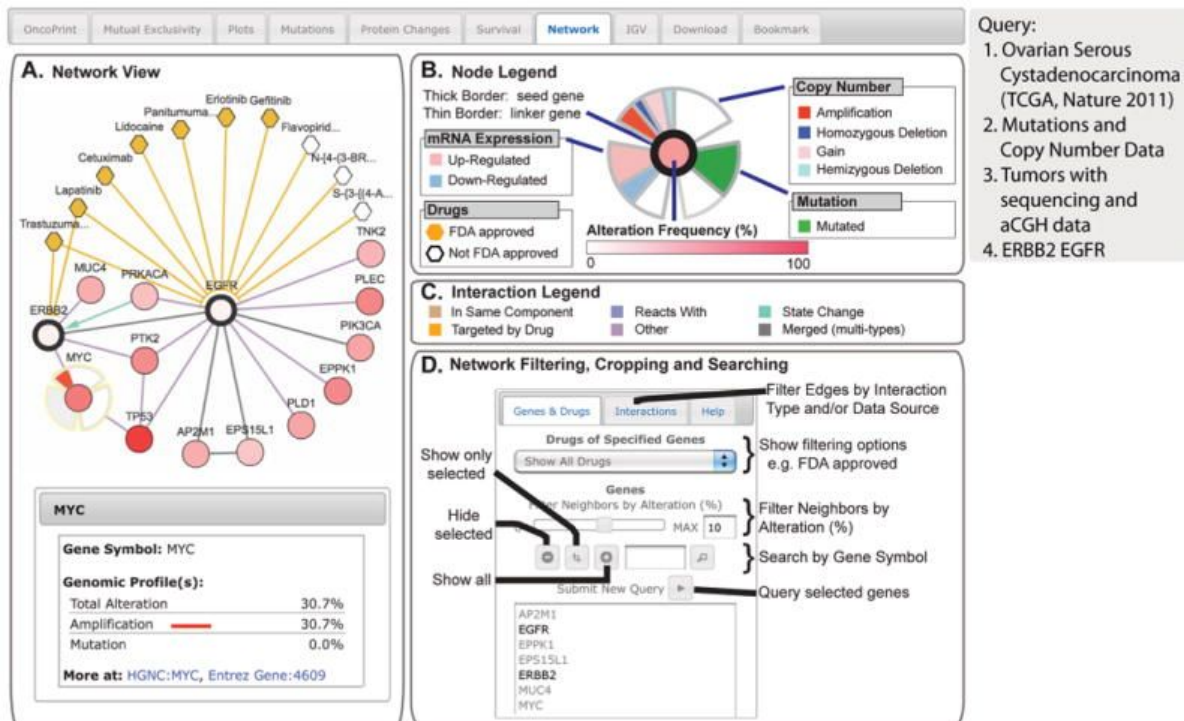
The Network tab has the following functions, as mentioned in the paper "*Integrative Analysis of Complex Cancer Genomics and Clinical Profiles Using the cBioPortal*"

Quoting the paper ;

The example shows network analysis of EGFR networks in serous ovarian cancer. (A) Network view of the EGFR and ERBB2 neighborhood in serous ovarian cancer (TCGA data set) rendered with Cytoscape Web (34). The query genes, EGFR and ERBB2, are outlined with a thick border, and nearest neighbor genes are color-coded by their alteration frequency in ovarian cancer. One can display drugs that target EGFR or ERBB2 (hexagons; orange indicates FDA-approved), as well as details about genomic alterations and links to external resources for any gene in the network (bottom left, example MYC). (B) The "Gene Legend" accessed from the "Legend" button. Mousing over any gene in the network or single-clicking the gene displays multidimensional genomic data (copy number, mutation, and mRNA expression) onto all nodes in the network. (C) The "Interaction Legend" accessed from the "Legend" button. Double-clicking the edge displays additional details about the interaction between the two nodes. Edges can represent different interaction types (color-coded, such as "reacts with"). (D) Options for filtering, cropping, and searching the network are shown.

A lot of precise hints/instructions will go into the page to make sure all the points from (A) to (D) are covered for the end-user.

But no task is impossible. With proper guidance and hard work, i think we will be able to *write* the *right* code to solve this problem.



- There is a lot of *dynamic behavior* on the cBioPortal website. While adding the web tours, this should be especially taken care of.

For example :

Performing a query of a single cancer study” form on the Homepage by default, has only three sections in it. These are enabled in case the user wants to query across cancer studies (*by selecting more than one study*). The following screenshot shows this case :

The screenshot displays the 'Query' form on the cBioPortal website. At the top, there are two tabs: 'Query' (active) and 'Download Data'. The form is divided into three main sections:

- Select Cancer Study:** This section includes a search bar with the text 'Search...' and a dropdown arrow. Below the search bar, it says 'No studies selected.' There is a tree view of cancer studies:
 - ☐ All (1)
 - ☐ Breast (1)
 - ☐ Breast Invasive Carcinoma (1)
 - ☐ Test study es 0 826 samples
- Select Data Type Priority:** This section has three radio buttons: 'Mutation and CNA' (selected), 'Only Mutation', and 'Only CNA'.
- Enter Gene Set:** This section has two options: 'Advanced: Onco Query Language (OQL)' and 'User-defined List' (selected). Below the 'User-defined List' option is a text input field with the placeholder text 'Enter HUGO Gene Symbols or Gene Aliases'.

At the bottom of the form is a 'Submit' button.

However, in the case of querying a single cancer study, the form changes to a four section form, and then the web tour should show a different set of instructions. We deal with this dynamic behavior in the *Proof of Concept* implementation of this project, which follows later in this document.

Action plan

1. Get familiar with the cBioPortal website from both a technical and a non-technical viewpoint.
2. With the website in mind, go through the examples listed in the paper titled *"Integrative Analysis of Complex Cancer Genomics and Clinical Profiles Using the cBioPortal"* that provides a practical guide to the analysis and visualization features of the cBioPortal for Cancer Genomics.
3. Get familiar with the cBioPortal website design. This should give an idea about the elements to target when implementing the interactive web tour.
4. Setup a private instance of cBioPortal by referring to the documentation.
5. Get familiar with the portal's codebase, forked off of GitHub.
6. Download and setup Intro.js
7. Read through Intro.js documentation
8. The aforementioned paper and *Getting Started* slideshow on the Tutorials page are decent sources of some tour recipes that cBioPortal needs.
9. Start by implementing a **Proof of Concept** for the project. This would help evaluate how this idea (*of a web tour*) works, looks and feel on the website. A good place to start would be the tour of **"Performing a query of a single cancer study"** (*example 1 in the aforementioned paper*), which can be found on the site's home page.
10. To implement *bullet point 9*, we would need to mark the four instructions in a way that they appear corresponding to their sections like shown in the figure above :

The screenshot shows the cBioPortal Query interface with four numbered annotations pointing to specific sections:

- 1 Select a cancer study**: Points to the "Select Cancer Study:" dropdown menu, which is currently set to "Glioblastoma (TCGA, Nature 2008)".
- 2 Select genomic profiles**: Points to the "Select Genomic Profiles:" section, which includes checkboxes for "Mutations", "Copy Number data", and "mRNA expression".
- 3 Define a patient / case set**: Points to the "Select Patient/Case Set:" dropdown menu, which is currently set to "Tumors with sequencing and aCGH data".
- 4 Enter genes of interest**: Points to the "Enter Gene Set:" section, which includes a text input field containing "CDKN2A CDK4 RB1".

The interface also includes a "Download Data" tab, a "Build Case Set" button, and a "Submit" button at the bottom.

data-step="1" data-intro="Select a cancer study"

data-step="2" data-intro="Select genomic profiles"

data-step="3" data-intro="Define a patient / case set"

data-step="4" data-intro="Enter genes of interest"

Adding the above code to the right elements would complete our task.

11. After the initial proof of concept implementation of the idea, the next step is to figure out the possible tours we could make based on the examples from the mentioned paper.
12. Also, work on a plan to keep the tours working with changes to the code base.
13. When *phase 1 coding* begins, work extensively on **bullet points 11 and 12**.

Timeline

(Till phase 1 of coding. I thought it would only be fair to take it one step at a time. I will update the timeline as and when the goals are met.)

- I. March 14, 2017 - Write a first draft of the project proposal
- II. March 14 - April 1 - Discuss the project proposal with a mentor / mentors from cBioPortal and keep making the required changes to the proposal.
- III. April 3, 2017 - Submit final application
- IV. April 4 - April 25 - Keep working with private cBioPortal codebase. Keep interacting with mentors, seeking help and guidance whenever needed.
- V. April 26 - May 13 -Unavailable due to exams in college.
- VI. May 4, 2017 - Keep fingers crossed as list of accepted student proposals get announced.
- VII. May 13 -May 29 - Community bonding period, as per *Google Summer of Code 2017 Timeline*.
- VIII. **May 30 - June 25** - Phase 1 Coding begins, as per *Google Summer of Code 2017 Timeline*. I can start working full time on my project as I will be done with my exams and having my summer break in college. I can devote as much time as is needed (I won't mind working 40 hours / week if it is required because that is how passionate I am about this project) to deliver on **bullet point 11** (delivering the tours mentioned in the list of Possible web tours, that follows this section) **and bullet point 12** (work on a plan to keep the tours working with changes to the code base) **from my Action Plan**.

Possible web tours

The following list contains a list of web tours I think should exist on cBioPortal to help the visitors. All of these tours should be feasible to implement on the portal. Having said that, the mentors would be more suited to suggest what tours they would actually need on cBioPortal.

As recipes for the tour, I would be using the text mentioned in the paper "*Integrative Analysis of Complex Cancer Genomics and Clinical Profiles Using the cBioPortal*", unless explicitly otherwise provided.

1. Performing a query of a single cancer study (*implemented in the Proof of Concept*).
2. Tours to provide help / instructions while viewing and interpreting the results.
 - We can choose tabs that would be very complex for the visitor to understand completely until a set of hints / instructions is provided to them. To facilitate this, *Intro.js* provides a function specially **designed to add hints** to the page. These hints appear as *floating bubbles* next to the elements. On being clicked, the bubbles display a pop-up with the required hint / instruction.
 - Although these might seem trivial, **IGV, Download, Bookmark** steps can also be explained via a web tour.
3. Again, this might not be required but since we are adding tours all across the portal, we can implement them for **Programmatic Access** (*REST-Based Web Service Interface and R and MATLAB Packages*).

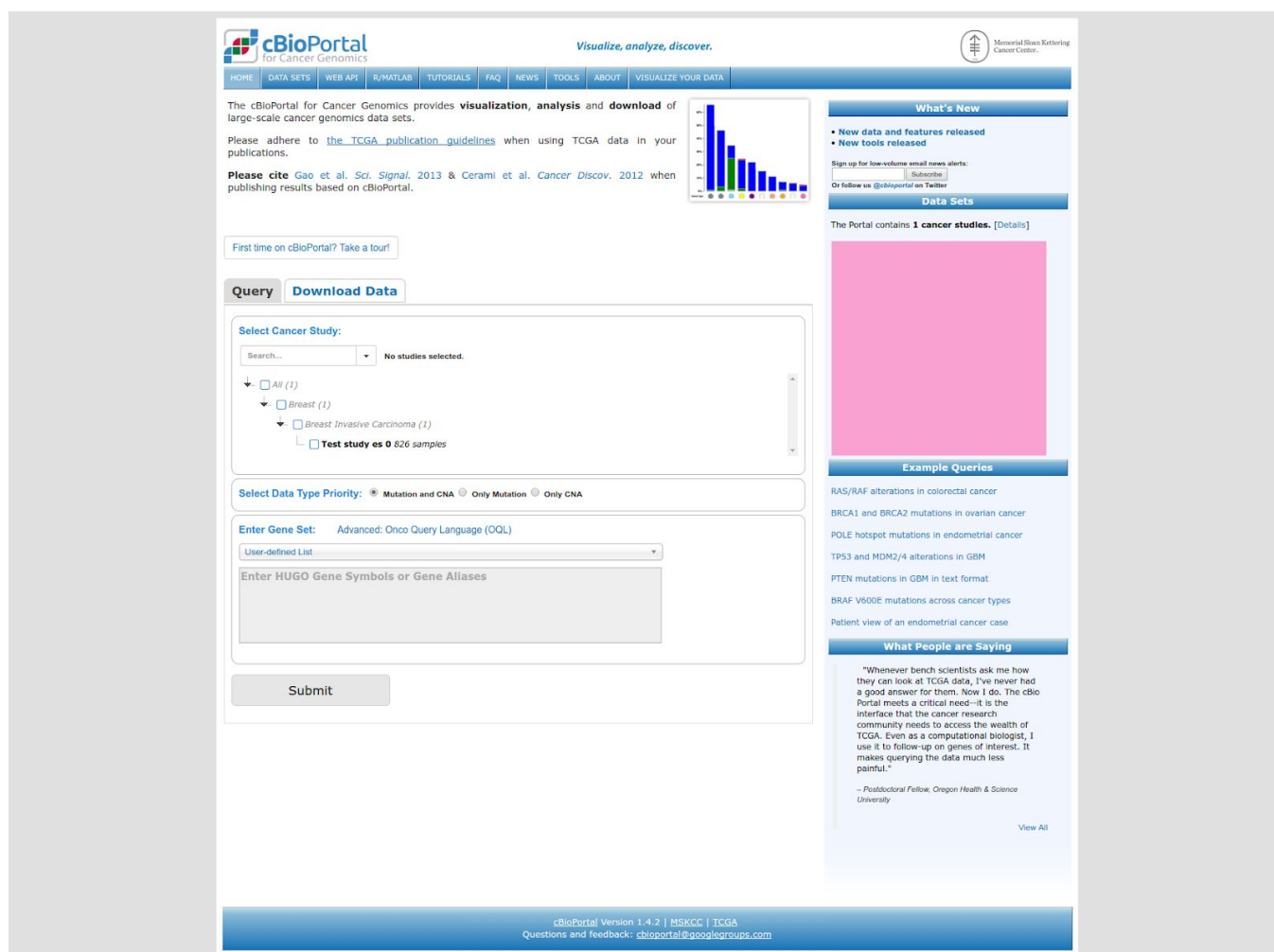
Apart from these tours, any tour suggested by the mentors that they think should exist on cBioPortal can be discussed and implemented.

Proof of Concept

“Performing a query of a single cancer study” (*example 1 in the aforementioned paper*)

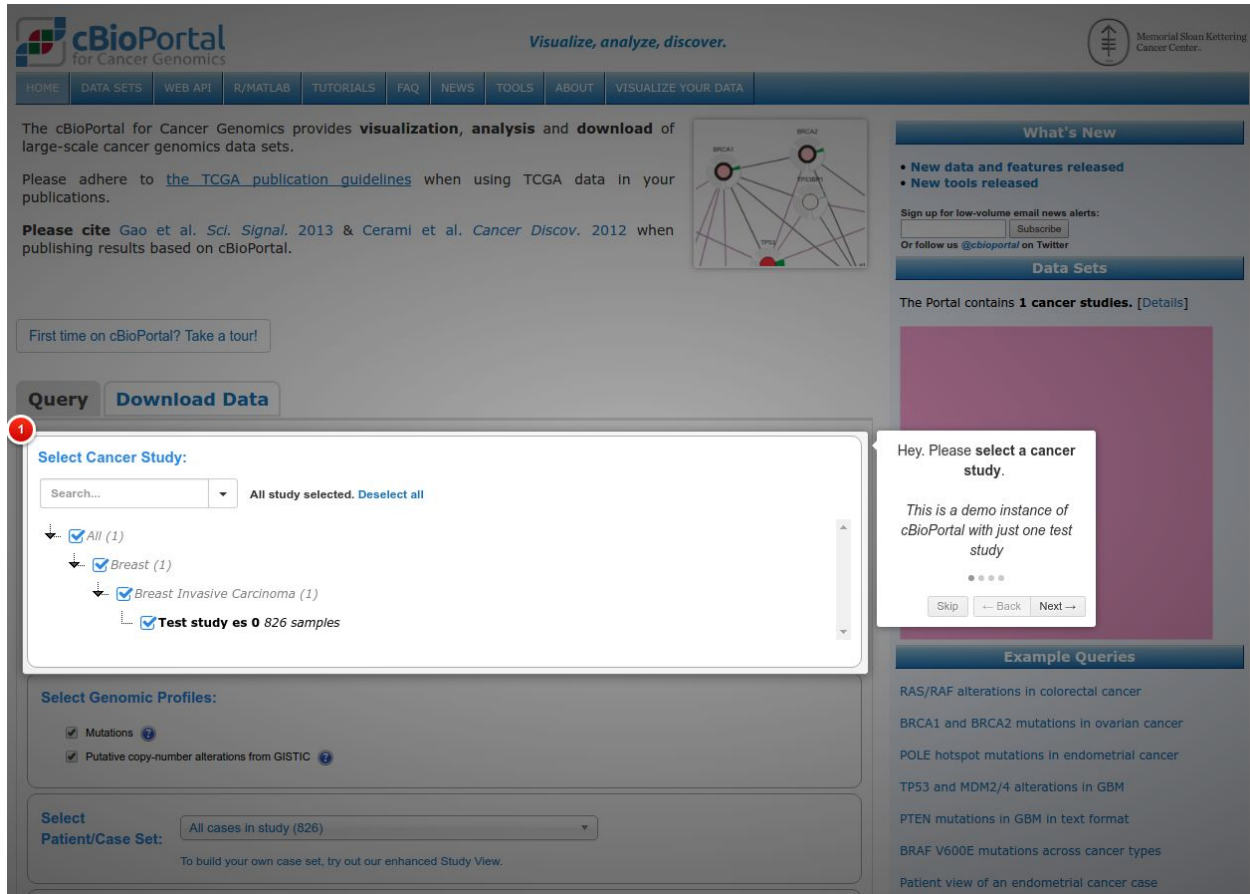
- We begin by adding a bootstrap button right above the “**Performing a query of a single cancer study**” form on the Portal’s index page. The button should say something like “*First time on cBioPortal? Take a tour!*”, that makes new users click on it, if they want to take the web tour. This makes the tour, a choice for the user. If they want to take it, they can. Or else if they feel confident using the portal, they can go ahead and do so without being forced to take a web tour.

The following image shows the position of the mentioned button :



- When the button is clicked, `introJs().start();` function is called that begins the interactive web tour.

Image showing the first step of the web tour, asking the user to select a cancer study.



- The following three images show the other three steps of **“Performing a query of a single cancer study”** as per the steps mentioned in the paper *“Integrative Analysis of Complex Cancer Genomics and Clinical Profiles Using the cBioPortal”*.

Navigating to the next or the previous step can be done either by clicking on the buttons that pop-up beneath the instruction or by using the right-left arrow keys on the keyboard

- Step 2 : Select Genomic profile(s)

Query **Download Data**

Select Cancer Study:

Search... All study selected, Deselect all

☒ All (1)
☒ Breast (1)
☒ Breast Invasive Carcinoma (1)
☒ Test studies 0 826 samples

2 **Select Genomic Profiles:**

☒ Mutations
☒ Putative copy-number alterations from GISTIC

Select Genomic profile(s)

.
 Skip ← Back Next →

Select Patient/Case Set: All cases in study (826)

To build your own case set, try out our enhanced Study View.

Enter Gene Set: Advanced: Onco Query Language (OQL)

User-defined List

Select Genes from Recurrent CNAs (Gistic)

Enter HUGO Gene Symbols or Gene Aliases

Example Queries

RAS/RAF alterations in colorectal cancer
 BRCA1 and BRCA2 mutations in ovarian cancer
 POLE hotspot mutations in endometrial cancer
 TP53 and MDM2/4 alterations in GBM
 PTEN mutations in GBM in text format
 BRAF V600E mutations across cancer types
 Patient view of an endometrial cancer case

What People are Saying

"Whenever bench scientists ask me how they can look at TCGA data, I've never had a good answer for them. Now I do. The cBio Portal meets a critical need--it is the interface that the cancer research community needs to access the wealth of TCGA. Even as a computational biologist, I use it to follow-up on genes of interest. It makes querying the data much less painful."
 — Postdoctoral Fellow, Oregon Health & Science University

- Step 3 : Define a Patient / Case set

Select Genomic Profiles:

☒ Mutations
☒ Putative copy-number alterations from GISTIC

3 **Select Patient/Case Set:** All cases in study (826)

To build your own case set, try out our enhanced Study View.

Define a Patient / Case set

.
 Skip ← Back Next →

Enter Gene Set: Advanced: Onco Query Language (OQL)

User-defined List

Select Genes from Recurrent CNAs (Gistic)

Enter HUGO Gene Symbols or Gene Aliases

Submit

Example Queries

RAS/RAF alterations in colorectal cancer
 BRCA1 and BRCA2 mutations in ovarian cancer
 POLE hotspot mutations in endometrial cancer
 TP53 and MDM2/4 alterations in GBM
 PTEN mutations in GBM in text format
 BRAF V600E mutations across cancer types
 Patient view of an endometrial cancer case

What People are Saying

"Whenever bench scientists ask me how they can look at TCGA data, I've never had a good answer for them. Now I do. The cBio Portal meets a critical need--it is the interface that the cancer research community needs to access the wealth of TCGA. Even as a computational biologist, I use it to follow-up on genes of interest. It makes querying the data much less painful."
 — Postdoctoral Fellow, Oregon Health & Science University

View All

- Step 4 : Enter Genes of interest. All queries must include at least one gene.

Select Patient/Case Set: All cases in study (826)

To build your own case set, try out our enhanced Study View.

Enter Gene Set: Advanced: Onco Query Language (OQL)

User-defined List

Select Genes from Recurrent CNAs (Gistic)

Enter HUGO Gene Symbols or Gene Aliases

Submit

Enter Genes of interest.

All queries must include at least one gene.

Done Back Next

cBioPortal Version 1.4.2 | MSKCC | TCGA
Questions and feedback: cbioportal@googlegroups.com

The above example is just to serve as a *Proof of Concept* for this project. Changes can be made to the whole implementation if need be. The recipes for the tour can be made more detailed and precise when the tour is added to the main website.

Starting here, the aim would be to work closely with the mentors over the summer, and take the project to it's finish line.

Update to the cBioPortal Documentation

My project proposal, which has been covered in the preceding parts of this proposal, deals with implementing interactive web tours across cBioPortal. This required me to set up and **deploy a local instance of the web application** on my machine running Ubuntu 16.04 LTS.

During the process however, I learned that some of the instructions written in the official documentation (<https://cbioportal.readthedocs.io/>) have become obsolete. This might be due to newer versions of Ubuntu coming out every year while the Documentation still considers Ubuntu 14.04 as the standard. Furthermore, there are a few places in the Documentation that seem to be written for the *power users* as they lack even the slightest of explanations. These sections will definitely benefit if a few necessary details are added to them.

The sole purpose of any Documentation is to make any task simpler and easier for the user. Having gone through the process of *deploying a local instance of the web application recently*, I have it quite fresh in my mind and would like to put my experience down in words, by adding to the existing doc, so that it helps others like me to deploy cBioPortal with ease.

This section of my proposal only mentions the corresponding sections of the official cBioPortal Documentation that need any form of updating based on my experience. The labelling should further help clear any ambiguities.

2.1 Deployment

Hardware and Software Requirements

Apache Tomcat

As of this writing, the cBioPortal runs properly on Apache Tomcat version 6 and above. The software can be found and downloaded from the Apache Tomcat website.

On Ubuntu:

[See Recommended Installation Instructions for Ubuntu 14.04](#)

[See Recommended Installation Instructions for Ubuntu 16.04](#)

Pre-Build Steps

Prepare the log4j.properties File

Update the following lines with paths that make sense for your local system.

```
log4j.appender.a.rollingPolicy.FileNamePattern =  
${java.io.tmpdir}/cbioportal.log.%d.gz  
log4j.appender.a.File = ${java.io.tmpdir}/cbioportal.log
```

You have to replace the *`${java.io.tmpdir}`* with its appropriate location on your local system. To do so, run the command

```
ps -ef | grep java.io.tmpdir
```

Particularly with tomcat apps, you may get an output like this:

```
-Djava.io.tmpdir=/usr/local/tomcat/temp
```

`${java.io.tmpdir}` is the highlighted part.

Set the PORTAL_HOME Variable

Prior to building, you must specify an environment variable for `PORTAL_HOME`. This must point to the root directory containing the portal source code.

For example, add the following to your `.bash_profile`

```
export PORTAL_HOME=/Users/ecerami/dev/cbioportal
```

Ubuntu however uses `.profile` by default. You can create `.bash_profile`, but then `.profile` will not be read.

After setting up the variable, make sure to check if the variable has been successfully created by running the following command in terminal. You should see the path to your root cBioPortal folder as the output.

```
echo $PORTAL_HOME
```


Loading a Sample Study

Importing the Sample Study

To import the sample study:

```
cd <your_cbioportal_dir>/core/src/main/scripts/importer
```

and then run the following command:

```
./metaImport.py -s ../../../../test/scripts/test_data/study_es_0/ -n -o
```

You will see a series of output messages, hopefully ending with a status message like this:

Done.

Total time: 7742 ms

During this step, you might be asked to run the *migrate_db.py* script located in :

```
<your_cbioportal_dir>/core/src/main/scripts/
```

You need to give in two parameters for the script to run properly:

1. -p : the location of the *portal.properties* file, which is located in
 <your_cbioportal_dir>/src/main/resources/
2. -s : the location of the *migration.sql* file, located in
 <your_cbioportal_dir>/core/src/main/resources/db/

Then run the command from your terminal. An example final version of the command might look like this :

```
./migrate_db.py -p ../../../../src/main/resources/portal.properties  
-s ../resources/db/migration.sql
```

Deploying the Web Application

Set Environment Variables

The following environment variable is referenced in this document and is required for successful portal setup :

`CATALINA_HOME :`

Which points to the home directory of your Apache Tomcat installation.

To make it available to your bash shell, add the following to your `.bash_profile`

```
export CATALINA_HOME=/path/to/tomcat
```

Ubuntu however uses `.profile` by default. You can create `.bash_profile`, but then `.profile` will not be read.

After setting up the variable, make sure to check if the variable has been successfully created by running the following command in terminal. You should see the path to your Tomcat installation folder as the output.

```
echo $CATALINA_HOME
```

Note: If you are following the *recommended Ubuntu instructions* you should set :

```
export CATALINA_HOME=/opt/tomcat
```

Add PORTAL_HOME to Tomcat

The `PORTAL_HOME` environment variable needs to be available to the `cbiportal.war` file which runs within the Tomcat server. To make it available to Tomcat, edit your Tomcat startup file (typically `$CATALINA_HOME/bin/catalina.sh`) and add the following line anywhere within this file (we typically add it near the `JAVA_OPTS` statements):

```
export PORTAL_HOME= $CATALINA_HOME +
"/webapps/cbiportal/WEB-INF/classes/"
```

A good place to add the above statement is shown in the following snippet.

```
# Uncomment the following line to make the umask available when using
the

# org.apache.catalina.security.SecurityListener

#JAVA_OPTS="$JAVA_OPTS
-Dorg.apache.catalina.security.SecurityListener.UMASK=`umask`"
```

```
export PORTAL_HOME=$CATALINA_HOME +  
"/webapps/cbioportal/WEB-INF/classes/"
```

If after adding the statement, your Tomcat server starts to show some errors, you can try pointing `PORTAL_HOME` to just the root of your cbioportal source code instead of using the variables. In this case the statement you should be using would be :

```
export PORTAL_HOME=/Users/ecerami/dev/cbioportal
```

Add the MySQL JDBC Driver to Apache Tomcat

A proper JDBC driver will also need to be accessible by Apache Tomcat. If using MySQL, the [Connector/J](#) driver jar file should be placed in `$CATALINA_HOME/lib`

Download the file from the given link, and then extract its contents to a folder. You will find ***mysql-connector-java-*-bin.jar*** file inside. Place this file in `$CATALINA_HOME/lib`

More information on configuring Apache Tomcat connection pooling can be found [here](#).

We have reports that the Tomcat package that comes with (at least) Ubuntu 14.04 cannot handle the connection pool from resources. If you are encountering this, we suggest you [download the Tomcat archive from Apache and install from there](#).

Deploy the cBioPortal WAR

A tomcat server is usually started by running the following command :

```
$CATALINA_HOME/bin/catalina.sh start
```


or, if you are following the *recommended Ubuntu instructions*

For Ubuntu 14.04: `sudo initctl restart tomcat`

For Ubuntu 16.04: `sudo systemctl restart tomcat`

After the tomcat server has been started, to deploy the WAR file, run the following command :

```
sudo cp portal/target/cbioportal-*.war  
$CATALINA_HOME/webapps/cbioportal.war
```



After doing this, you can look in the tomcat log file
(`$CATALINA_HOME/logs/catalina.out`) to see if the portal has been proper deployed.
You should see something like :

```
INFO: Deployment of web application archive  
/Users/ecerami/libraries/apache-tomcat-7.0.59/webapps/cbioportal.war  
has finished in 13,009 ms
```

Personal background

Computer Science has been the only subject that I ever had fun studying. I liked it because I could relate to the things and understand them. I am in my third year of undergrad now, and I spend much of time learning, experimenting and exploring new things in my field. I am specially fond of algorithms and their applications in the real world simply because how interesting and challenging they are, and because of the infinite number of problems they are able to solve.

I also like to develop for the web. I am currently in the process of acquiring a Full Stack Specialization from Coursera.org. Being a MOOC (Massive Open Online Courses) enthusiast, I have 5 Certificates of Accomplishments on Coursera. Languages I command include C/C++ and Java, but I am always partial towards C++ as that's how it started for me. Other technologies I have a working knowledge of include SQL, Python, Git, Latex. HTML, CSS, Javascript, jQuery, Bootstrap, Wordpress and Angular2. I am comfortable working in both Linux based and Windows environments.

I can work independently on projects which require interdisciplinary skills. Having said that I won't brag to be proficient in all the technologies I claim to have knowledge of. But I do believe that I am very flexible and intuitive, and can relate things with each other and make them work. In the true sense, I believe this is what software engineering is all about; putting into use everything at your disposal in an efficient way that gets the job done.

I want to keep learning and improving myself. As of now, I would love a real world test of my skills and **Google Summer of Code 2017** would be just the right platform for it. I believe in results and seeing projects through to completion, come what may. I think that is the most comprehensive way to learn and completing a project with cBioPortal will be an immensely fulfilling experience.

Personal Information

Name: Abhinav Pratik

Email: pratikabhinav@gmail.com

Date of birth: December 20, 1995

Mobile: +91-704-271-9924

Course: Bachelor of Technology (B.Tech.) in Computer Science (2014-18)

College: Amity School of Engineering and Technology (ASET), Amity University, Noida, Uttar Pradesh

What are your languages of choice and how do they relate to the project?

I have experience working with HTML, CSS and Javascript in the past. I have certification from The Hong Kong University of Science and Technology for the same, and several independent projects to validate my credentials. My project involves all of the three languages and I feel quite comfortable moving forward with a project which uses them extensively.

I keep my work on GitHub. The link is : <https://github.com/pratikabhinav>

Any prior experience in bioinformatics or cancer genomics (this is not a deal-breaker, we just want to understand your background)

No, I have no experience in bioinformatics or cancer genomics. But I have been reading on the two topics and making myself aware about the basics to start with.

My Availability

This is my first attempt at GSoC. I want to cherish the experience and learn as much as I can from my time here.

I have my ongoing classes which would be followed by my exams, all of which would end on 13th May. After that my summer break begins, which would end in July. Until May I can easily give in a couple of hours everyday to my project. Once my vacation starts, this project will be my primary task and I can, spend majority of my time working on it.

Do you have any school / college related activities scheduled during the coding period?

No, I do not have any college related activities scheduled during the coding period.

Do you have a full- or part-time job or internship planned for this summer?

No, I do not have a full time job or internship planned for the summer. I might do a part-time internship though, to further utilize my time and make the best use of my vacation.

How many hours per week do you have for a summer project?

I plan to give in 30-40 hours per week for this GSoC project.

Credits for Intro.js

Website : <http://introjs.com/>

GitHub : <https://github.com/usablica/intro.js>

Docs : <http://introjs.com/docs/>

Open-source license : GNU AGPLv3

Author : Afshin Mehrabani (afshin.meh@gmail.com)

And many other [contributors](#).