Kinase Kinpendium

DOCUMENTATION

Zoonii, Mohomed, Numaan, Raz

About

Kinase Kinpendium is a functioning prototype of a web-based software tool for handling molecular biology data. This tool allows the user to explore background information about human protein kinases and the sites they phosphorylate. It is capable of inferring the level of activity of all human kinases with a known substrate from experimental uploaded datasets. The web application was developed by a group of 5 students under the supervision of Professor Conrad Bessant and Dr. Fabrizio Smeraldi as part of the MSc Bioinformatics Software Development Group Project at Queen Mary University of London.

The Kinase Kinpendium website search tools are based on a curated database that holds general information about Protein Kinases (ID, Uniprot Accession ID, Name, Symbol, Group, Family, Sub Family, Synonym, Function, Genomic Location, Subcellular Location), Inhibitors of Protein Kinases (Name, Chembl ID, Smiles, Kinase Target) and Phosphosites (phosphorylation sites) for each protein kinase (Substrate ID, Kinase Accession ID, Kinase Gene, Kinase Symbol, Substrate, Substrate Accession ID, Substrate Gene, Site, Domain, Field).

The major features on Kinase Kinpendium allow the user to explore the curated database. This includes searching for information about kinases by name or accession number, or by searching for a substrate and finding its kinase. The user can view the protvista annotated features of each protein, as well as explore the neighboring genomic environment using an iframe of UCSC's Genome Browser loaded with uniprot annotated gene tracks and modified residue sites. Finally, the last feature of Kinase Kinpendium is the Data Analysis tool. This allows the user to upload quantitative phosphoproteomics data in csv or tsv format. The web application summarises this data graphically and provides an estimate of the relative activity of the kinases known to phosphorylate the substrates provided in the sample.

Contents

Software outline

1.1 Site Map

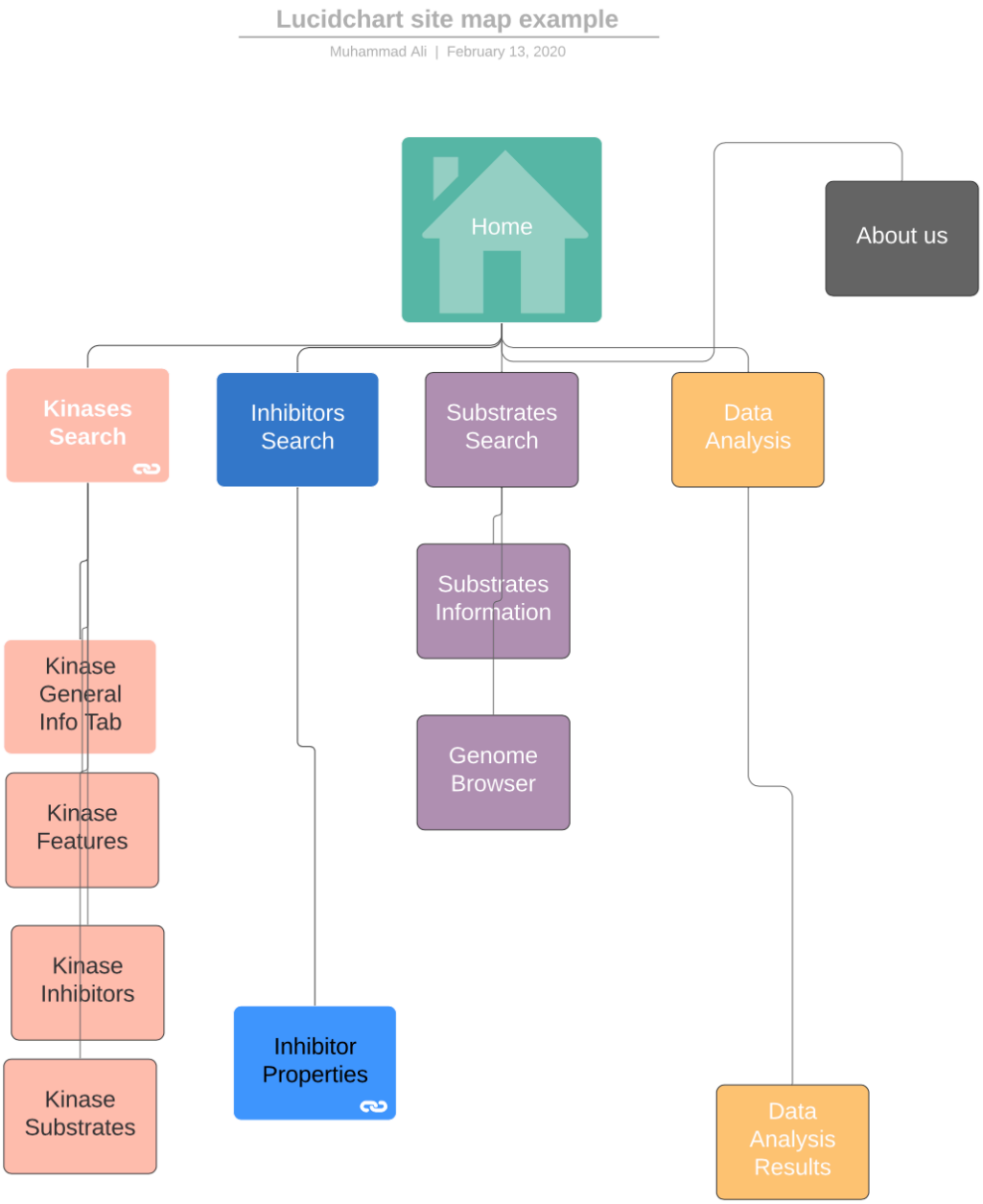

Figure 1 Fig1 shows a concise summary of the site map and an idea of the routes. The Home page and the navbar at the top of each page links to all other pages via the various tabs as well. Each of the Search function will link to one page with relevant tabs.

1.2 Software Architecture

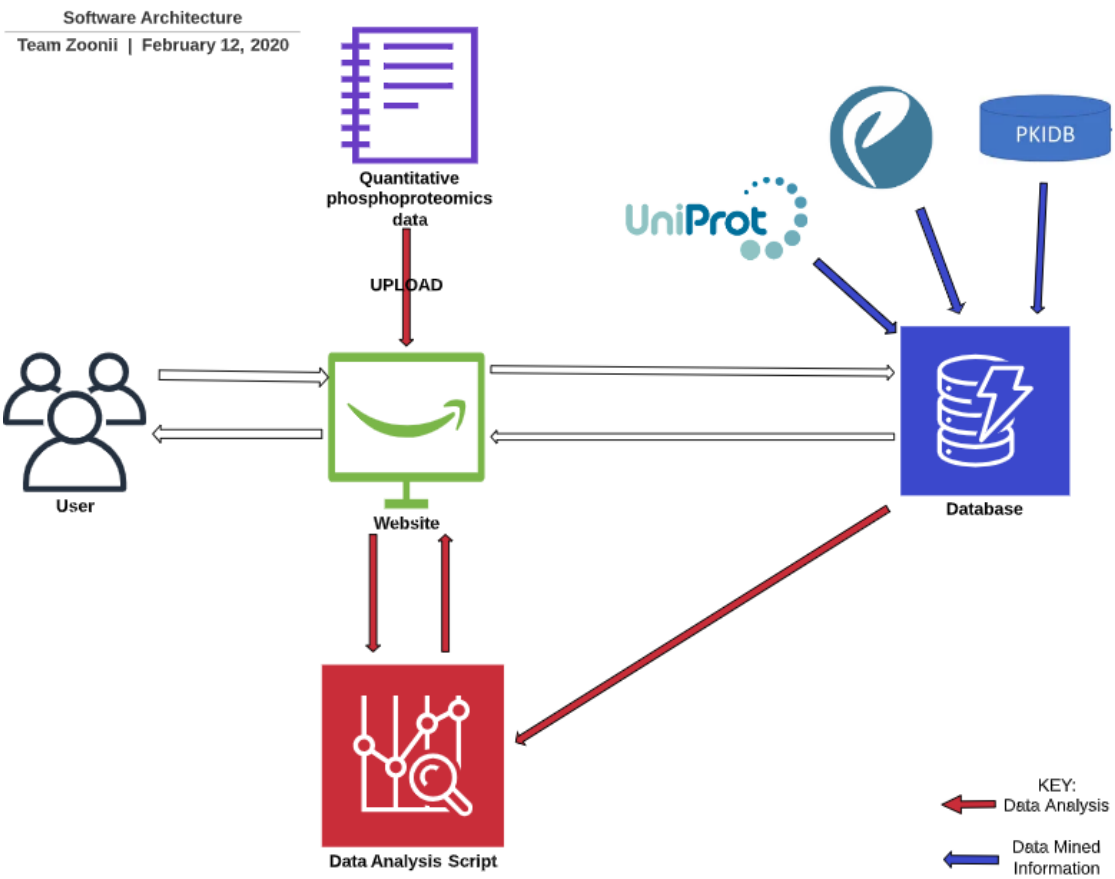

Figure 2. The Kinase Kinpendium software architecture shows the how the front end and back end of the website connect.

The software was developed using Flask, a web server gateway interface (WSGI) web application framework. The website page routes are defined using HTML language and CSS was also used for website design. It was decided that the best way to retrieve information was from a database which was created using SQLite 3 and the data was retrieved using SQL-Alchemy. The database itself would be comprised of data from the following three sources: UniProt, PhosphositePlus and PKID. The website would be deployed using Amazon Web Services (AWS) Elastic Beanstalk.

1.3 Information on how to run Kinase Kinpendium

Kinase KInpendium has been succesfully deployed and running. Click the link below to access the Kinase Kinpendium.

http://kinase-kinpendium.us-east-2.elasticbeanstalk.com/

In order to deploy the website locally, the user must ensure the requirements and the aforementioned packages are installed. Following this, you can download kinase_kin directory from the github page:

https://github.com/zooniikayler/BIO727P-Group-Project

Local deployment for Windows:

```
set FLASK_APP=main.py
set FLASK_DEBUG=1
flask run
```

Local deployment for Linux:

```
export FLASK_APP=main.py
export FLASK_DEBUG=1
flask run
```

Using the data analysis tool:
To use the Data Analysis tool, please upload a file in 'csv' or 'tsv' format as specified in Box 1. Visualizations can be downloaded using the icons in the top right corner of each plot.

Box 1. Accepted data format

substrate	control_mean	condition_mean	fold_change	p_value	control_cv	condition_cv
GENE_NAME(S455*)**	Value	Value	Value	Value	Value	Value

* this is the modified residue by letter followed by its position in the protein sequence
** gene name OR protein name can be accepted

1.4 Packages Use

Flask 1.1.1
Flask-SQLAlchemy 2.41.
Flask-Table 0.5.0
Flask-WTF 0.14.2
Jinja2 2.10.3
Jupyter 1.0.0
Pandas 0.25.1
Requests 2.21.0
Beautifulsoup4 4.8.0
Bokeh 1.3.4

Pipreqs
One of the requirements to deploy a website to AWS Elastic Beanstalk was to generate a requirements.txt file. Pipreqs allows you to automatically generate a requirement.txt file based on the imports within the application script.

Flask
Kinase Kinpendium was built using Flask as our main web development toolkit. We choose Flask for it's ease of use and simple functionality & how it connected the web framework & database queries written in Python to the web pages front end via the Jinja2 template

engine. We could also utilise FLASK Forms and Bootstrap templates to create the Searchfields that enabled us to filter key information from the database.

1.5 CSS/HTML

HTML was used with the CSS languages to make the website more presentable and professional looking. Bootstrap was also utilised to provide relevant CSS and HTML design templates for search forms, buttons, tables, navigation bars and image carousels. W3-CSS was also used to enable built-in responsiveness CSS framework that also contains built in responsiveness. We used a layout.html file to serve as our base template to feed in relevant html bootstrap components into other pages such as the navbar and footing. This layout was inherited in all other html pages as well as the main.css style sheet which all pages used as well. We utilised Jinja2 a templating engine to feed in special placeholders from the main.py python script into relevant pages. This enabled us to occupy different html tags with outputs from the database filters/queries which were generated in python. All our templates called on the static folder which held the main.css file which was our principle style sheet, we also had a tabs.css file in the static folder to provide CSS style guides for the relevant tabs and tables. A combination of Bootstrap and W3-CSS was essential to display information in tables and relevant tabs and Jinja2 enables us to occupy those fields with relevant information from our database.
JavaScript was utilised to make the website more dynamic and add an additional feature such as the Feature Viewer embedding from Uniprot. This enabled users to explore the annotated Kinase sequence.

Data collection

2.1 Protein Kinase Data

The raw data for Protein Kinases was obtained from the KinHub website (http://www.kinhub.org/kinases.html). This version of the csv file contained information such as the Kinase Symbol, Name, Group, Family, Subfamily and Uniprot Accession Number. Furthermore, to approach the requirements of the group project, additional information such as the Synonym, Function, Genomic location, Subcellular location and a PDB image link were added and the data was curated. The gathering of the additional information was obtained by using the following websites: UniProt (https://www.uniprot.org), Ensembl (https://www.ensembl.org/index.html) and Protein Data Bank (http://www.rcsb.org). This final version of the file is available in both excel and csv data formats. Protein Kinase data was used for the Kinases search function to browse general information about protein kinases.

2.2 Phosphosites Data

Available Phosphosites data (Kinase_Substrate_Dataset.gz and Phosphorylation_site_dataset.gz) were downloaded from the PhosphoSitePlus website (https://www.phosphosite.org/homeAction.action). In order to obtain the relevant information from the raw downloaded files (human data and protein kinase post-

translational modifications only), the datasets were filtered. Phosphosites data was mainly used for the Data Analysis tool as well as for the Substrates search function therefore allowing the user to browse for substrates phosphorylated by a protein kinase.

2.3 Inhibitor Data

The inhibitor data was gathered from PKIDB – A Curated, Annotated and Updated Database of Protein Kinase Inhibitors in Clinical Trials (http://www.icoa.fr/pkidb/index.html). This curated data contains information such as the Inhibitor Name, ChEMBL ID, Smiles and Kinase Target. This was copied into a csv file then Pandas was used to clean the data. Webscraping via Beatfiful Soup was used to extract the image links as well as the ChEMBL ID's. Unwanted columns were deleted in order to get rid of inhibitors without a Kinase target or without a ChEMBL ID.. Inhibitor name and ChEMBL ID data were used as the search parameters for users to peruse Kinase Inhibitor information.

Database

3.1. Database creation and data retrieval

SQLite 3 was used to build the database due to its numerous advantages. Some of the key advantages was its portability, reliability and accessibility. Due to the ease of accessibility, Third-party application such as DB Browser was essential in checking if the database generated correctly.

SQLAlchemy-FLask was preferred over SQLite3 for data retrieval due to its ease of access with python and allows for sessions to be created which connect to the database. Theses session can be accessed to query through the data. For Kinase Kinpendium three type of queries were used. The first being a simple query where it searches through an entire table by a given string. The second type of query was a filter query, this would search through column within a table by a given string. The final type of query used was the join query, this was used to retrieve information from two tables that were linked together by a foreign key. An example of this when the user searches a given kinase it would retrieve information from the kinase table and the substrate table

3.2 Database Schema

For all the CSV files that were obtained from the data mining stage were used to populate and create the tables for the database. In total there are four tables as shown in *Figure.2,* each table has a primary key which is unique to each field. The tables are linked to each other using a foreign key and eventually are joined together for use in queries using either inner join or .join. Certain fields in the tables that are essential to have conditions statements for example in the KinaseInfo table the Kinase_Name is set as Not Null to ensure no field is empty in that column. Many-to-One relationships were also set up as shown in the figure below.

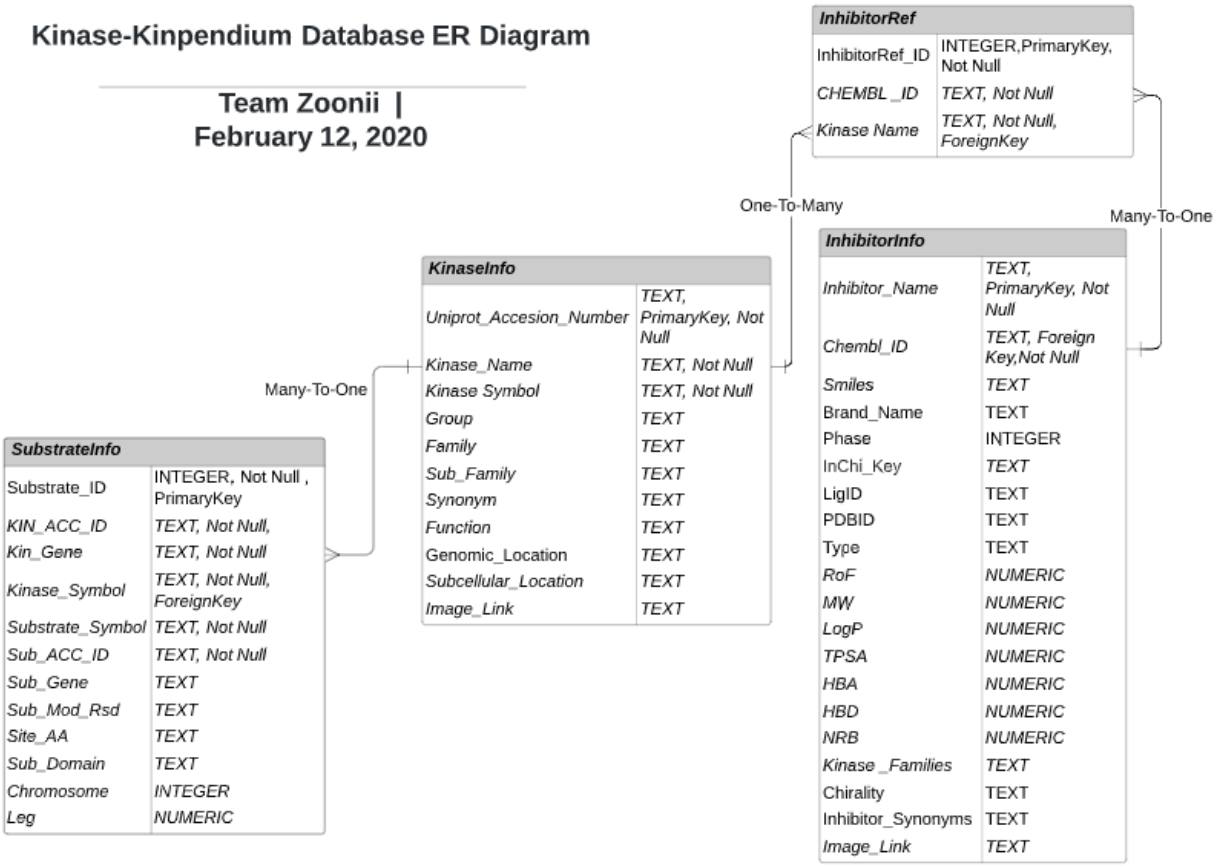

Figure 3 Kinase-Kinpendium Database ER Diagram showing the relationship between the different tables within the database.

Kinase Kinpendium Features

4.1 Kinases search function

Kinase Kinpendium allows users to search for Kinases via the Kinase search bar as well as inputting the Kinase Symbol into the URL when on the Kinase initial search page (e.g /Kinases/AKT1). This will produce the same results as if utilising the Kinase search bar. Users can choose from the search bar whether they search for Kinase Symbol or Uniprot Accession Number. Kinase Symbol name was chosen as it's shorter than the written out full name of the kinase and a key identifier anyone hoping to search for information about a specific kinase should know. Uniprot Accession Number was chosen as this links to the well annotated Uniprot database and each Kinase has a unique accession number, making it ideal to identify. The user has a choice which one of these they wish to utilise to search for a certain protein kinase. Both search queries will query the 'KinaseInfo' table that will then deliver information on that specific kinase. Once Kinase or Uniprot search has been completed the user will see the following tabs with various information; General Information, Kinase Features, Inhibitors, Substrates. In General Information tab users can find information on the following; Kinase name, family, sub family, genomic location, cellular location, function, known synonyms & image generated from protein data bank. They will also see Uniprot Accession Number which an external hyperlink to the Uniprot page for that particular kinase is. On the Features tab users will find an embedded Feature Viewer from Uniprot/ProtVista. This BioJs web-based widget lays out the maps, orients and provides position-based annotations for sequences for the searched Kinase. On the Inhibitor Tab users will get a list of inhibitors

(presented as a table with a column for Inhibitor name and ChEMBL ID) the inhibitor name is an internal hyperlink to the inhibitor search results page for that specific inhibitor. The Substrates tab will show a table of the relevant substrates for that Kinase. The Substrates are also internal hyperlinks that if clicked will lead to the substrates search results for that specific substrate.

4.2 Substrates search function

Kinase Kinpendium enables users to search for Substrates/Phosphosite data via the Substrate search bar as well as inputting the substrate symbol name into the URL when on the Substrate search page (e.g. /Substrates/EIF2-ALPHA). Both options leads to the same Substrates results page which shows 2 tabs; General Information & Genome Browser. General Information is a table output of all the substrates with information about; the Accession Number (this is an external hyperlink to the Uniprot page for that Substrate), Phosphorylator, modified residue, site of Phosphorylation, chromosome number & chromosome Region. The Genome Browser is an embedded widget of the UCSC Genome Browser, there's instruction on this page on how to utilise it and see individual phosphosites and see modified residue track markings.

4.3 Inhibitors search function

Another major feature of Kinase Kinpendium is the ability to search specific protein kinase inhibitors. Users can search by Inhibitor International Non-proprietary Name (INN). Inhibitor data was specific to the name stated in the source document used for inhibitor data, which was collected from the following website; (http://www.icoa.fr/pkidb/index.html). Users can also input the Inhibitor International Nonproprietary Name (INN) into the url when on the inhibitor search bar page (e.g. /Inhibitors/Afuresertib). Both search options will lead to the inhibitors results page. This search queries will query the 'InhibitorInfo' table. Once searched the user see a singular tab (inhibitor properties) which houses the main information about the will get information on a number of properties of that specific inhibitor, such as; Smiles, InCHI Key, molecular weight, rule of five, LogP, TPSA, HBA, NRB & Kinase family of that specific kinase target. Chembl ID is also shown and this serves as an external link to the Chemble report of that respective inhibitors ChEMBL ID. There is a limitation on this feature as we were unable to link information about Inhibitor and its respective Kinase Targets. This explained more in the limitations section.

4.4 Data Analysis tool: Uploading Phosphoproteomic data

The data analysis tool calculates 2 scores with a set significance threshold (0.05) and produces 4 visualisations. All plots will auto-scale to the size of the user's browser, potentially hiding some scores/data points. To view a sub-section of any plot, one can click and drag to select a window to see all data within that range. Prior to all calculations, rows with 0 or infinity in any column of the user data are removed.

1. The first visualisation is a bar plot called, "Kinase Activity Mean Score," this shows the computed data points from the user's uploaded file in ascending order in terms of their "kinase activity". Kinase activity is defined here as the mean of the $\log_2$ of the fold change between all substrates known to be affected by the kinase.
2. The second visualisation, "Top Ten Kinase Activities," shows only the top ten highest, and bottom ten lowest mean kinase scores, allowing comparison between the kinases showing notable fold change.
3. The third visualisation shows the "Alternative Kinase Score." This is the number of significant phosphosites with a positive fold change minus the number of significant down-regulated phosphosites in each substrate set. Substrate sets are defined as all substrates affected by a kinase in our database. Therefore, this score is useful to see

which kinases have an overall positive or negative affect on phosphorylation. Those with an alternative score of 0 are not shown.

4. The fourth visualisation is a volcano plot allowing the user to easily evaluate how much of their data is above the significance threshold, and how much of their data they can expect to see at the corresponding level of fold change. This plot takes the $\log_2$ of the fold change on the x-axis and the p-value on the y-axis. The y = 0.05 line is shown in orange for reference.

Deploying Kinase Kinpendium

5.1 Amazon Web Service Elastic Beanstalk Deployment

Amazon Web Services (AWS) has a wide range of services that each have their own advantage. AWS Elastic Beanstalk was chosen as the preferred method of deployment due to it numerous advantages. It allows support for a variety of different programming languages such as Python. It also allows for quick deployment and management with the AWS Cloud console or AWS command line interface. Another advantage of using AWS Elastic beanstalk was the process of deployment is straight forward as it automates the setup, configuration and load balancing.

AWS Elastic Beanstalk deployment interface was intuitive and easy to use. Once the application was deployed your able to check the health status of the webserver. For any reason the health status changes there is plethora of tools that can be used to analyse the change in health status. The most useful tool was the ability to request the last 100 lines from the logs, allowed us to work through any errors.

In order to successfully deploy the application, there are 3 essential steps. The first step is to create a requirement text file which is essential for deployment. This allows AWS Elastic beanstalk to import any packages needed into the virtual environment. There are many ways to go about this. The simplest way we found was using pipreqs to generate the requirements.txt file.

```
$ pipreqs /kinase_kin/main.py
```

This generated all the packages used for that application as opposed to using pip3 freeze > requirements.txt, which included the packages in the environment.

The next step was to ensure the WGSI path of AWS elastic beanstalk search for the application. In order to make this work we had to change the file name from main.py to application.py. This requirement also caused us to change variable names with main.py from app to application (e.g. @app.route changed to @application.route). It was possible to avoid this by modifying the WGSI path to main.py but it was highly recommended to change the Flask app name to application.

The final step to successfully deploy the application was to zip the contents of the application folder. The zip file should include the aforementioned requirements.txt and application files. The zip file is then uploaded, and deployment is almost instantaneous.

Once these steps are completed the URL provided by AWS Elastic Beanstalk should be live (e.g. http://kinase-kinpendium.us-east-2.elasticbeanstalk.com/) and the health status shown in the AWS cloud console should be shown as ok.

Limitations

6.1 Web feature limitations

Kinase Kinpendium is currently unable to show the kinase targets for the inhibitors searched by the user. This may cause an issue as it doesn't rely to the user a key piece of information. However, there is a workaround in place. The user can search for a kinase and select the known inhibitors tab to view the inhibitors that are known to inhibit the searched kinase. Kinase Kinpendium does not give the user flexibility in data analysis parameters. All analyses are completed with a significance threshold of 0.05 for the alternative scoring system and includes all data points in the primary scoring system. This may limit the usability of the tool in some situations. The genome browser does not open directly to the coordinates of the searched substrate, this causes a minor inconvenience as the user must search the substrate in the UCSC browser iframe after having searched it on Kinase Kinpendium. This is due to missing chromosome coordinates.

Further Development

The data analysis tool leaves room for much development. The current prototype includes no parameters for varying analyses. Improvements could include: a setting for significance threshold, function to analyse data with more than one inhibitor, a function to upload one's own kinase-substrate relationship data and a drop-down menu to select which visualisations are necessary for the user's analysis that would save processing time by eliminating the creation of unnecessary plots.

The Inhibitor Search results needs to pull out the Kinase targets for each Inhibitor, although we have this information in our database it proved difficult to incorporate this into the website, by the time we identified a workout by inputting the results from the queries into a dictionary it proved too long to incorporate in the time we had.

There appears to be a difference between searching through the Kinases via the search function and typing said kinase name into the URL when on the Kinases page e.g.(/Kinases/AKT1). When using this feature of typing the Kinase into the URL it presents the same results page but for some reason the Inhibitor Tab & Substrate Tab shows all the inhibitors & substrates on the website, instead of the specific inhibitor and substrate for that Kinase – which searching via the Kinase search button provides.

The Substrates results page presents the results in a clunky table, for further development we should have presented this in a cleaner table. The Substrates results table shows lots of information, one being the Kinase which Phosphorylates that substrate, ideally the user would be able to click on this and it would lead to the Kinase results page for that Kinase.

The Kinase Results, Substrates Results & Inhibitor Results need to be downloadable as a csv file for users to download directly. The Genome Browser needs to specifically show the genomic location of that specific Substrate we needed to webscrape that from Ensemble add it as an additional column and add that to the table for SubstrateInfo.

References

1. Carles F, Bourg S, Meyer C, Bonnet. (2018). “PPKIDB: A Curated, Annotated and Updated Database of Protein Kinase Inhibitors in Clinical Trials.” Molecules. 15;23(4). pii: E908. doi: 10.3390/molecules23040908.
2. Casado P, Rodriguez-Prados J-C, Cosulich SC, Guichard S, Vanhaesebroeck B, Joel S, and Cutillas PR. (2013). “Kinase-substrate enrichment analysis provides insights into the heterogeneity of signaling pathway activation in leukemia cells.” Science. 6(268):6.
1. Hornbeck PV, Zhang B, Murray B, Kornhauser JM, Latham V, Skrzypek E. (2015). “PhosphoSitePlus, 2014: mutations, PTMs and recalibrations." Nucleic Acids Res. 43(D5)12-20.

Awknowledgements

The UniProt Consortium
UniProt: a worldwide hub of protein knowledge
Nucleic Acids Res. 47: D506-515 (2019)