PATHWAY CURATOR: AN ONLINE WEBSERVER FOR EXTRACTING GENES AND INTERACTIONS FROM FIGURES

A Thesis

presented to

Electrical Engineering and Computer Science Department

at the University of Missouri-Columbia

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

ZITING MAO

Dr. Dong XU, Thesis Supervisor

MAY 2022

The undersigned, appointed by the dean of the Graduate School, have

examined the thesis entitled

PATHWAY CURATOR: AN ONLINE WEBSERVER FOR EXTRACTING GENES AND INTERACTIONS FROM FIGURES

presented by Ziting Mao

a candidate for the degree of Master of Science,

and hereby certify that, in their opinion, it is worthy of acceptance.

____

Professor Dong Xu

____

Professor Mihail Popescu

____

Dr. Ye Duan

ACKNOWLEDGEMENTS

I would want to express my gratitude to Professor Dong Xu and Dr. Fei He, who assisted me in defining the project's direction at the start of the development process, and also gave me a lot of useful suggestions during the development. I would also want to thank Professor Mihail Popescu, who provided me with a lot of biology advise while I was working on this project. Finally, I'd want to express my gratitude to Dr.Shuai Zeng, who provided me with a lot of innovative suggestions on several technological challenges.

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

ABSTRACT

In the biomedical literature, gene pathways are frequently included. Many high-quality gene pathways are illustrated in the form of visuals and text, making them valuable study tools for biological processes and precision medicine. Pathway maps and literature texts provide researchers with access to a huge number of new biological treatments. For general usage, these pathway maps should be logically ordered, coordinated, and converted into a computer-readable format. Currently, keeping up with the rapid increase of the literature requires laborious extraction of information from a publication at a time.

A gene pathway map recognition system is devised and implemented in this study. Based on the pathway map and relevant information supplied by users, the system extracts gene identity and gene interaction information, and the automated extraction from pathway maps is efficient. Furthermore, the tool offers users with a full view of a certain illness's pathway, which is useful for researchers and can speed up the research process in a variety of biomedical applications.

This thesis first explains the project's goal and provides the background information. The project's design ideas are then presented, as well as an analysis of the system and introductions to related platforms. After that, the system's implementations are described one by one, together with the deployment and testing processes. Finally, potential improvements and future work are discussed.

Keywords: image understanding, biological pathway, information retrieval, web server, Nodejs, React, MySql, RestfulAPI

1. INTRODUCTION

1.1 Project Background

The field of bioinformatics has evolved as a result of the rapid advancement of information technology and biotechnology. Bioinformatics combines biomedical knowledge with statistics and computer science to organize, analyze, and understand data in order to uncover biological secrets hidden in enormous amounts of complicated biological data. Bioinformatics research is data-driven, and the number and quality of data sets are critical to the success of bioinformatics experiments. In bioinformatics, data integration and database development are significant research methods. Building high-quality standardized databases not only saves time and effort for researchers when collecting and processing data in experiments, but it also enhances data use, facilitates information sharing, and encourages bioinformatics growth.

Moreover, in the huge biological literature, nowadays, the number of pathway maps is increasing at an unparalleled rate. PubMed, a medical literature search engine, now has over 30 million articles, many of which have biological route maps. Bio-medical data grows dramatically throughout the course of medical research, and these data are characterized by vast scale, rapid expansion, complex structure, and multidimensional value density, among other characteristics. Traditional data management and mining approaches are no longer adequate to meet the demands of modern scientific growth. The question of how to integrate and mine these huge bio-medical data sets has become a prominent topic in recent years.

Researchers find manual reading of the newest research advancements in the bio-medical sector time-consuming and hard when faced with such a large number of literature. As a result, automatic knowledge extraction from literature pathway maps is a viable solution to this problem. For example, manually sorting out the relationships between many genes in pneumonia disease is a massive project, so it's critical to extract gene information and gene relationships from route maps automatically.

This paper provides a target detection-based pathway map information extraction system for researchers with varying needs, which does not require manual extraction and instead requires only uploading the pathway map to acquire the related prediction results. The method described in this publication can provide a comprehensive cross-literature perspective for a certain gene or condition, avoiding the potential for manual extraction errors, lowering the time required to create route maps, and considerably assisting researchers in increasing their productivity.

1.2 Literature Review

The main problem studied in this paper is about gene identification and relationship extraction in path graphs. The two main approaches to solve this problem are manual collation and text mining. In the past few years, the problem of gene and protein image based detection and inference has attracted increasing attention. Kuenzi et al. performed manual collation of 2070 cancer pathway maps to identify incompletely mined features and discover new genes in known cancer pathway maps. Kozhenkov, et, al proposed a pathway map based biological knowledge extraction, comparison and retrieval management system, however, only a limited amount of information in the pathway map was organized in the paper to form a database, which cannot be flexibly extended. Although it makes sense to collect and maintain the information in pathway maps, the process of constructing, managing, integrating, and labeling pathway maps is time-consuming and laborious. At the same time, the number of PubMed articles grows to over 1 million per year, which is impossible to manage manually.

Another group called "25 Years of Pathways" successfully evaluated biological pathway maps produced in the last 25 years by bulk-building predictive models and organizing the data into a database. The retrieved data was then presented directly on their web server, making it more easier for researchers to get genes from pathway maps.

The following are the major flaws in the domestic and international techniques mentioned above.

1. Text mining accuracy is insufficient, and there are intrinsic flaws such as back-pointing and co-pointing that cannot be solved, as well as poor overall performance.

2. Manual extraction in most situations necessitates the processing of experts in the field with specialized knowledge, the entire operation process will be time-consuming and labor-intensive.

3. The preceding approaches extract only the genes and do not extract the links between genes, making it impossible to properly extract and use the information included in the pathway maps.

4. The approaches described above do not give a public mechanism for extracting pathway maps or software that can be widely used.

In summary, to address the inadequacies of the aforesaid problem, this study provides a tool for extracting genes and their interrelationships from biological literature. From pathway diagrams and texts, the technology in this work can extract information about gene entities and their interactions. Pathway diagrams of articles are frequently shown by simple shapes and clear links to make biological mechanisms more intuitive. The online website built in this study for extracting route diagram genes and their linkages can answer current challenges in scientific research and greatly aid scholars in extracting information about causative entities and their relationships from pathway diagrams.

1.3 Brief introduction of Pathway Curator

Based on the above, we can know that The coreferences and anaphoric expressions are still challenging in text mining tools.

To address this challenge, we propose an integrated informatics bio-purification pipeline that mines genes and their interactions from pathways using figures and texts of biomedical articles. We hypothesize that extracting genes and their interactions from pathway figures and texts will be more accurate and reliable than extracting them only from the texts or figures of articles. In order to visualize biological mechanisms, figures in articles are usually represented by simple shapes and clear relationships, which make the relationships between objects obvious. Since optical character recognition (OCR) of figures may not be accurate, the names of the extracted entities may not always be correct. In contrast, the text in the article provides more detailed information to further interpret the corresponding paths and correctly identify entity names, which can be used to filter out incorrect numerical entity names. At the same time, it is difficult to extract precise relationships between objects from the text due to NLP issues such as co-referencing. The pipe line of the model is shown in the figure1-3.

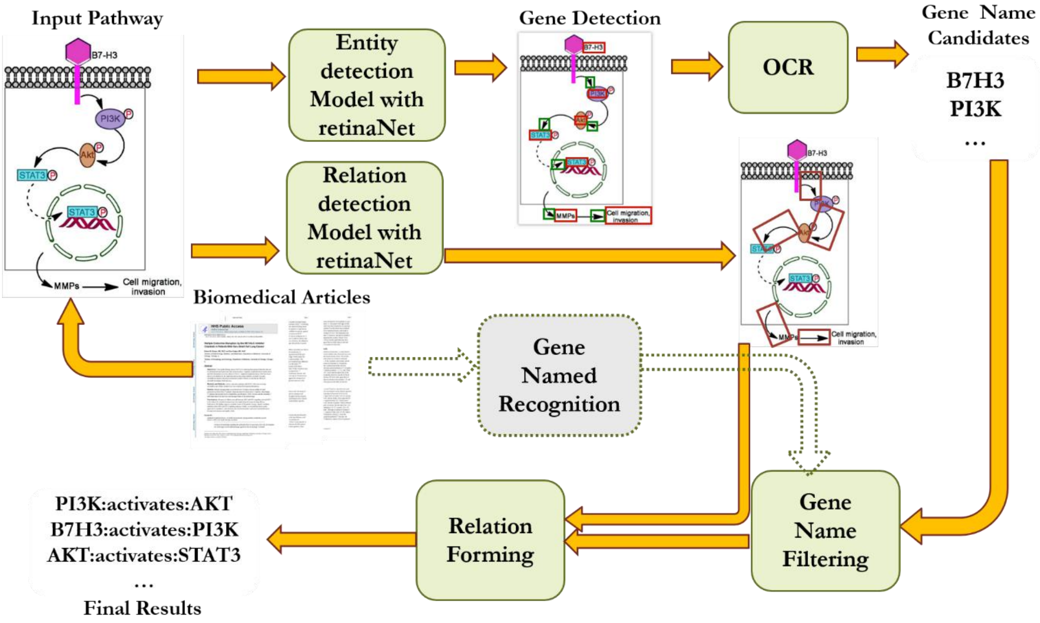


Figure 1-3 Overview of our gene-interaction extraction pipeline

Extracting information from images is a new direction in biocuration. Recently, only a few studies have used OCR to extract genes from publication figures. different extensions and improvements of OCR have been applied to segmentation of biological images, localization and recognition of text in images. In a large-scale analysis of pathway maps, gene names were retrieved from the images of PubMed Central articles, but the interactions between genes were ignored. Even though extraction of biological relationships from images has been described previously, no reproducible details or accessible online resources were provided. Our earlier study showed that it is feasible to retrieve gene names and gene relationships from pathway maps. In this study, we designed a more advanced deep learning-based pipeline to detect genes and their interactions from pathway maps, while using textual information from full papers to filter the results. As an initial attempt at pathway cu-ration, we focus only on genes and two key types of gene interactions, namely "activation" and "repression", which are typically plotted in text boxes with arrows and T-shaped lines. Other types of interactions will be studied in future work. To our knowledge, no previous work has been done to systematically extract biological mechanisms from figures in the literature.

Based on the above algorithm model, we also created a online web-server named Pathway curator. And, This article will mainly focus on this system. Pathway curator is an online tool that can analyze bioinformatics pathway figure.

The algorithm model used in the background is based on the RetinaNet network model and trained on a large number of biological pathway maps, which can extract the information of gene entities and the relationship between gene entities from the bioinformatics pathway figure. On the prediction page, the user can upload up bioinformatics pathway figures (one figure or one zip file which is up to 20MB) to our server. And, the results will appear on the results page, with a table showing all genes in the figure, another table showing all relationships, and a figure showing all genes and associations by circle the genes ad relations as box. Moreover, by clicking on the link in the gene table or the circled gene on the graph, users can go right to the information for each gene. We provide two online dictionaries, GeneCard and Uniport, to link the identified genes. The results can also be saved to the user's computer. All data and outcomes from successful prediction jobs will be saved in a database on our server for easy management.

2. SERVER SYSTEM REQUIREMENTS

Chapter one has already introduced some related background knowledge of pathway curator structure and described basic methods of model. This chapter will describe the requirements of a pathway diagram entity and relationship extraction system.

2.1 System basic functional requirements analysis

The major goal of this approach is to assist researchers in swiftly obtaining information from bioinformatics pathway figure.

The system should first allow users to upload bioinformatics pathway diagrams and related mate-data. After receiving the diagrams and mate-data, this multi-threaded system starts a new thread to call the model to analyze the images, and once it's done, the system returns the results and also displays them as table and graph on the show page. Users can access comprehensive information on each gene by clicking on the links in the gene list or directly on the genes circled on the graph. To link to information about the identified genes, the system provides two online dictionaries, GeneCard and Uniport. Users can also download the results to their computers. All successfully anticipated tasks' information and results are saved to the server's database. In addition, several additional functionality, such as uploading numerous photographs at once through zip, must be developed. Moreover, users have the ability to engage with results and manage their own jobs.

So users requirements can be divided into the following four steps, as shown in Figure 2-1:

(1) Uploading bioinformatics pathway diagrams and related mate-data like figure’s title, resource link, and the paper’s title, resource link, first author, PMC and PM id, publication year and keyword where figures come from.

(2) Predicting pathway diagrams: after user submit the bioinformatics pathway figures to the system, system will call model to predict these figures and then return the result. In the meanwhile, system will also keep the result into database.

(3) Results visualization: After getting the result, the system will visualize the results into chart and graph on the result page.

(4) Job management: Users can review their past job, and also can modify or delete them.

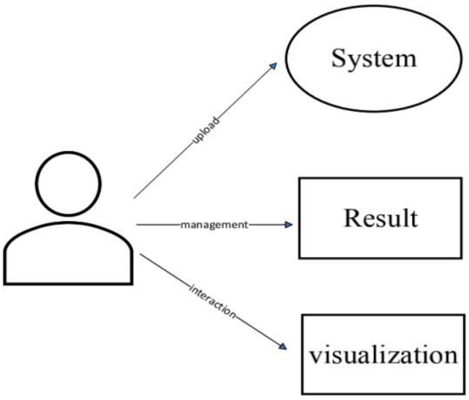


Figure 1 User requirements

Based on requirements analysis, the basic functions of the system are:

(1) Uploading model: Users can upload single bioinformatics pathway diagrams or multiple bioinformatics pathway diagrams as a zip file and optional mate-data to the system.

(2) Predicting model: The system starts a new thread to call the model to analyze the new target job.

(3) Saving data model: Saving result data into specific database.

(4) Results visualization: The system visualizes the predicted outcomes in graphs or charts to provide users with a visual representation of the results.

(5) Interaction with the result module: Users can interact with visualized predicting results, for example in the figure or table, user can modify, delete, add the result and click the specific result to see more details.

(6) Management module: Users can review or delete their past job.

2.2 System basic non-functional requirements analysis

The performance requirements of the system and the user experience must also be considered when analyzing the functional requirements of the system, so the non-functional requirements of the system are as follows.

(1) Timely: reduce forecast time while maintaining pathway map identification accuracy.

(2) Scalability: Allows for the addition of new functional modules and secondary development on the existing system's foundation.

(3) Simplicity of use: The interactive interface is straightforward and

welcoming, allowing users to accomplish their goals with as few steps as possible.

(4) Reliability: The technology can produce accurate predictions for all types of access map file formats.

2.3 Use Case Descriptions

2.3.1 Register

1. Use Case Name: Register user
2. Summary: System will created a new account for new users.
3. Basic Flow:

   a): The use case start when a new user firstly access the system.

   b): The system will create a new ID which is generated by a random eight-digit number and the time when the user accesses the system.

   c): The system will store the ID both into the new user's local storage and server's database.

   d): The system will start a loginsd session and display the homepage based on the use's preference.
4. Alternative Flows: None.
5. Extension Points: Register preferences.
6. Preconditions: None
7. Post-conditions: Now user can start the prediction.

2.3.2 Log-in

1. Use Case Name: Log-in user

2. Summary: The system will automatically log-in when the user accesses the system for the first time.

3. Basic Flow:

a): User accesses the system.

b): The system will start a login session and display the homepage based on the use's preference.

4. Alternative Flows: None.

5. Extension Points: None.

6. Preconditions: None.

7. Post-conditions: Now user can start the prediction and review his predicted jobs.

2.3.3 Prediction

1. Use Case Name: Prediction

2. Summary: User uploads the figures and mate-data to server, then the system will do the prediction and save the result into database.

3. Basic Flow:

a): User uploads one figure or one zip file which include multiple figures. Figure 2-3-3.1.

b): User uploads the figures' meta-data. Figure 2-3-3.2

c): User clicks the submit button to start the prediction.

4. Alternative Flows: None.

5. Extension Points:

a): The figure file must me a .jpg .png .Jpeg or .zip

b): The type of the input must be same as the example which in the input box.

6. Preconditions:

a): The figure file is required

b): The figure file is up to 20MB.

7. Post-conditions: If the figure file is not a zip file, the system will jump to result page immediately, and show the result when finished the prediction. If the figure file is a zip file the system will alter a massage prompt user go to history page to view the project's progress and review the result of the figures which have been predicted.

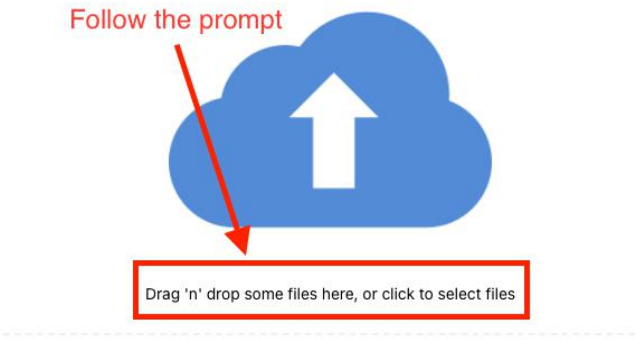
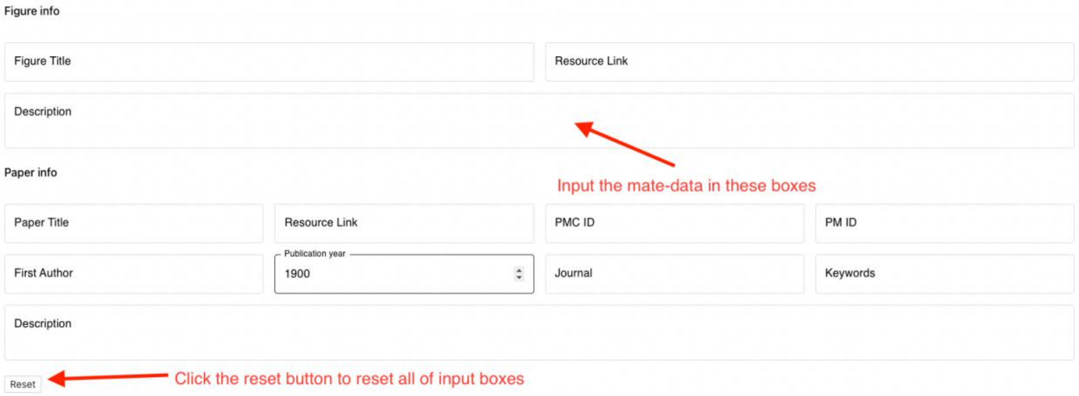


Figure 2-3-3.1 upload the figures



Publication year
1900

Input the mate-data in these boxes

Click the reset button to reset all of input boxes

Figure 2-3-3.2 mate-data input box

2.3.4 Show result

1. Use Case Name: Show result

2. Summary: User can see the result of the figures in this page. There is a table to present all the genes in the figure, another table to show all the relations, and a figure circling all the genes and relations on it as shown in Figure 5. Moreover, users can access the detail of each gene by clicking the link on the gene table or the circled gene on the figure directly. We provide two online dictionaries GeneCard and Uniport to link the identified genes. Users can also download the result to their local drive.

3. Basic Flow:

   a) User submitted a new job.

   b) The system will jump to the result page

   c) After seconds, the result will be displayed in this page.

4. Alternative Flows:

   a) User select one figure in the history page.

b) The system will jump to the result page

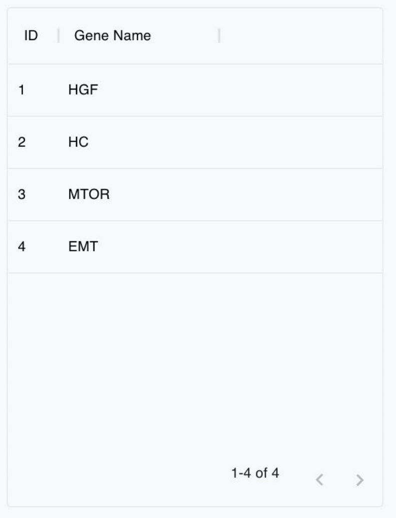
c) The back end system will query the result data in the database, and return the result data to front end

d) Display the result in the result page. A gene table, relation table and a figure. Figure 2-3-4.1, Figure 2-3-4.2,Figure 2-3-4.3.

5. Extension Points: None.

6. Preconditions: None.

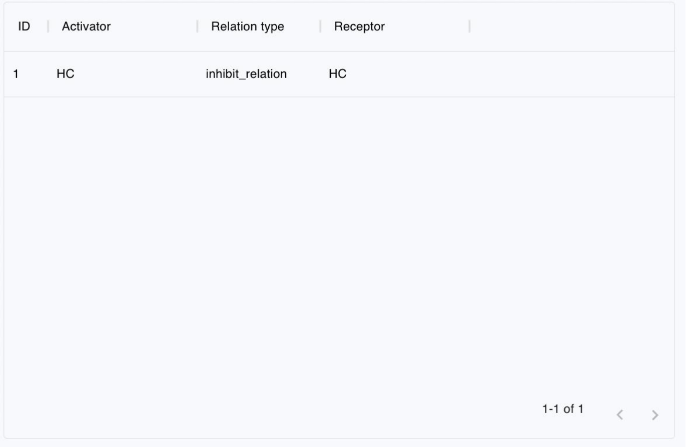
7. Post-conditions: User can view the prediction result of the figure and the matedata of the job in this page.



ID	Gene Name
1	HGF
2	HC
3	MTOR
4	EMT

1-4 of 4

Figure 2-3-4.1 Gene table



ID	Activator	Relation type	Receptor
1	HC	inhibit_relation	HC

1-1 of 1

Figure 2-3-4.2 relation table

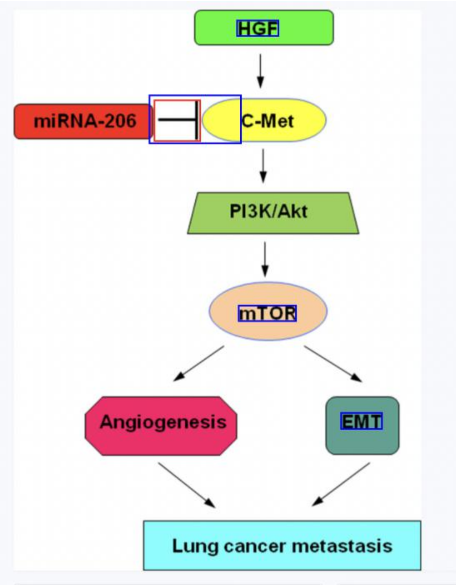


Figure 2-3-4.3 Figure

2.3.5 Review the past job

1. Use Case Name: Review past jobs

2. Summary: User can see the basic information of all of past jobs, and review the result of each one of them by click the button review.

3. Basic Flow:

   a) User access the history page.

   b) Select the preferred figure.

   c) Click the button review. Figure 2-3-5.

   d) The system will query the result data according the selected figure ID.

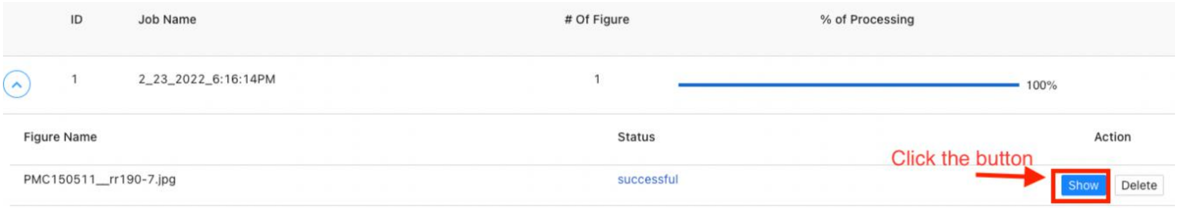
   e) The system will jump to the result page and display result data and mate-data.

4. Alliterative Flows: None

5. Extension Points: None.

6. Preconditions: The figure have been successfully predicted.

7. Post-conditions: User can view the prediction result of the figure and the mate-data of the job in this page.



ID	Job Name	# Of Figure	% of Processing
1	2_23_2022_6:16:14PM	1	100%

Figure Name	Status	Action
PMC150511__rr190-7.jpg	successful	Show Delete

Click the button

Figure 2-3-5 Show the result of the figure

2.3.6 Show mate-data

1. Use Case Name: Show mate-data

2. Summary: Show the mate-data of the job in the result page

3. Basic Flow:

   a) User accesses the result page.

   b) User clicks the button “show mate-data”.

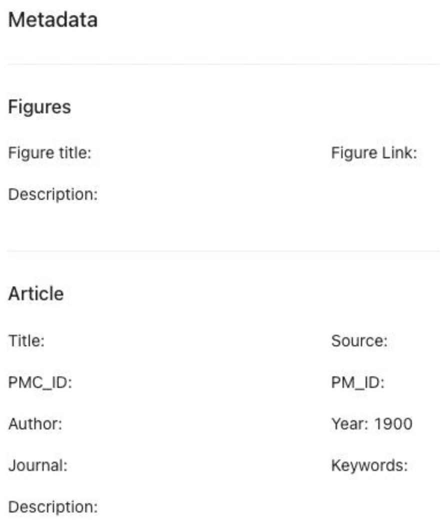
   c) The mate-data will be displayed in the left box. Figure 2-3-6

4. Alternative Flows: None.

5. Extension Points: None.

6. Preconditions: None.

7. Post-conditions: Users can view the mate-data of the job.



Metadata

Figures

Figure title: Figure Link:

Description:

Article

Title: Source:

PMC_ID: PM_ID:

Author: Year: 1900

Journal: Keywords:

Description:

Figure 2-3-6 Mate data box

3. SERVER SYSTEM DESIGN

The system adopts the browser/server (Brower/Server) architecture. the B/S architecture omits the special installation, and the system can be easily accessed through the browser. The core part of the system's functional implementation is structurally divided into three layers, as shown in Figure 3.

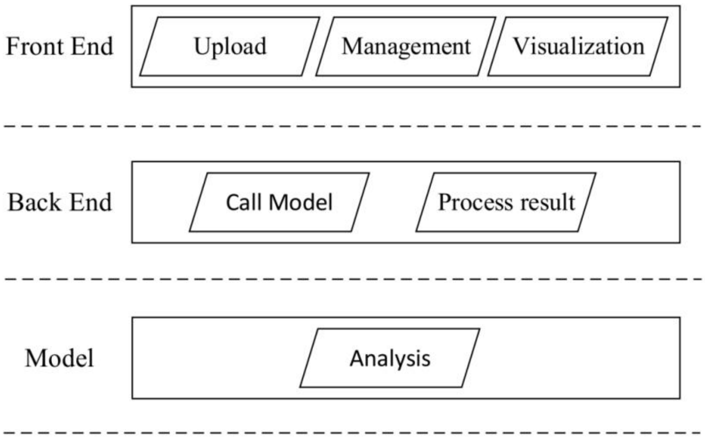


Figure 3: Architecture Diagram

The front-end display layer provides the operation interface for users, mainly with the function of uploading pictures, visualizing results and project management.

The back-end control layer is the core of the tool, which is responsible for connecting the front-end and back-end and processing the business logic in the tool, such as reading in the pathway diagram, adding, deleting and checking the prediction results, and implementing other core functions.

The model layer is to process the input pathway diagram, identify entity types for matching and return the identification results.

3.1.1 Front end functional design

According to the functional requirement analysis of the system, system front end can be divided into the following five functional modules, which are the home module, the prediction function module, the result module, the history module, and the Event module, Front-end functional design diagram is shown in Figure 3-1-1.

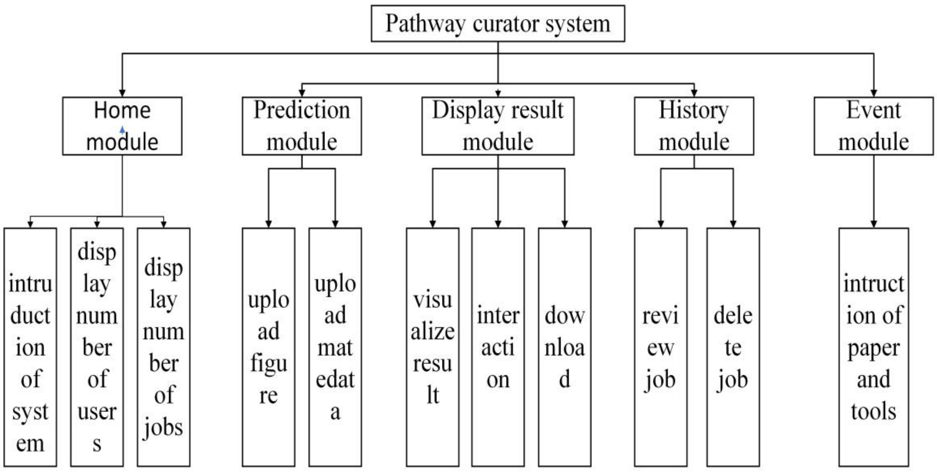


Figure 3-1-1: front end functional design diagram

(1) Home module:

(a) Display the introduction and the tutorial of this pathway curator system. Help users to better understand and better use pathway curator system.

(b) Display the total number of the users this pathway curator system has.

(c) Display the total number of the jobs this pathway curator system has.

(2) Prediction module:

a)Upload bioinformatics pathway figures.

b) Upload figures' mate-data.

(3) Result module:

a) Visualization the result.

b) Users can interact with visualized predicting results to understand results better.

c) Users can download the result as a JSON file.

d) Users can modify the result directly in the result page

(4) History module:

a) Display all of the jobs that users have uploaded.

b) Users can delete any specific jobs.

c) Users can review any jobs he want.

(5) Event module:

a) Display the instruction of our development group's publication paper and other tools.

3.1.2 Back-end functional design

According to the functional requirement analysis of the system, system back end can be divided into the following three functional modules, which are the call algorithm model module and the receive bioinformatics figures module and the processing result module. Back-end functional design diagram is shown in Figure 3-1-2.

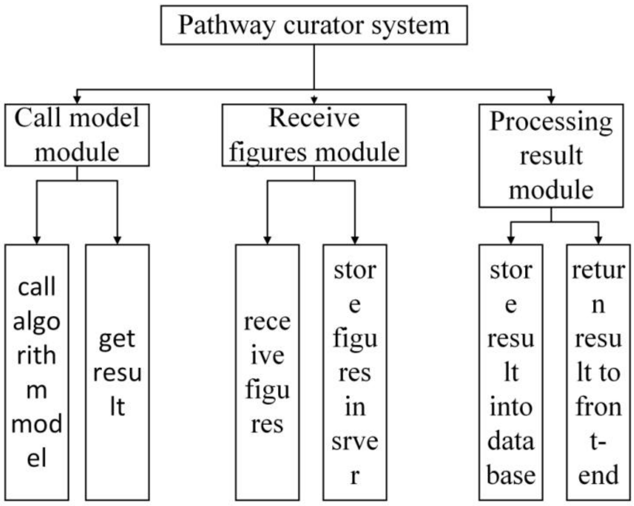


Figure 3-1-2: back-end functional design diagram

(1) Call algorithm model module:

a) call algorithm model by NPM package “child_process”.

b) Building result files as a JSON file and store it in server.

(2) Receive figures module:

a) Receiving the bioinformatics figures from front-end.

b) Storing the bioinformatcs figures in server machine.

(3) Processing result module:

a) Reading the result from the result file.

b) Storing the result into pathway curator’s database.

c) Return the result to front-end.

3.2 System framework Design

The common design patterns are MVC (model-view-controller) and MVP(model-view-presenter/controller). There is a significant distinction between MVP and MVC, in MVP, the View does not communicate directly with the

Model; instead, all interaction occurs within the Presenter (Controller in MVC), whereas in MVC, the View reads data directly from the Model rather than through the Presenter. Because the View in MVC may directly access the Model, the View will contain Model information as well as some business logic. The Model is not dependent on the View in the MVC model, but the View is dependent on the Model. Furthermore, because the View contains some business logic, changing the View is challenging.

Pathway curator system framework is based on the MVP pattern. The MVP design pattern divides the system framework into three parts: model, view, and control, which considerably reduces dependency between system functionality and data processing, resulting in increased development efficiency and code maintainability. MVC design pattern diagram is shown as Figure 3-2.1. MVP design pattern diagram is shown as Figure 3-2.2.

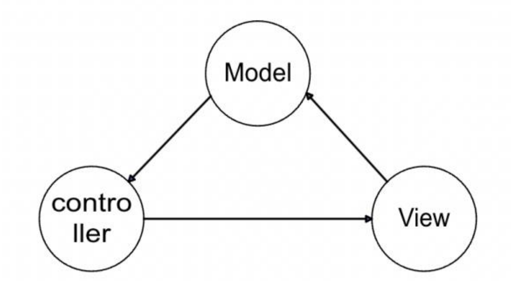


Figure 3-2.1 MVC design pattern diagram

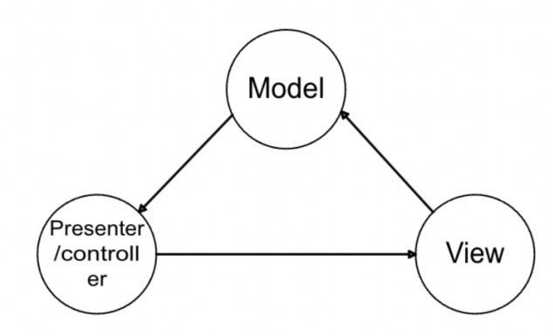


Figure 3-2.2 MVP design pattern diagram

1. Model: refers to the data model of the data transfer process, such as a specific object; for data transfer and data invocation, we normally need to change the records of a table stored in the database into the corresponding object.

2. View: a method of displaying data to the user, usually data sent by the controller via the page for display.

3. Presenter/Controller: data processing; the role is for the user's request; after data processing through the controller, the requested data is transmitted to the view, which then displays it to the user.

The MVP design pattern has many advantages, such as low coupling, high percentage of code reuse, low development life-cycle costs, and high maintainability.

The MVP design pattern has the following four advantages

1. Low coupling: The view and business layers are separated, which allows changes to the view layer code without recompiling the model and controller code. Similarly, changes to an application's business processes or business rules only require changes to the model layer of MVC. Because the model is separated from the controller and view, it is easy to change the data layer and business rules of the application.

2. High re-usability of code: In the MVP design pattern, the three modules are independent of each other, and the interaction between them is realized through the interface, so different data and functions can interact with each

other through the same interface, so this greatly improves the re-usability of the code.

3. Low life cycle costs: Each development phase is independent, so it facilitates self-development and testing. This significantly reduces the time cost of development.

4. High maintainability: Separating the view and business logic layers also makes the WEB application easier to maintain and modify. The traditional design pattern where each development phase is mixed together makes modifying the code a painful task, as modifying a small feature may modify the entire life-cycle. However, in the MVP design pattern, because each phase is independent of each other. Thus, the maintenance cost is greatly reduced.

3.2.1 Front-end framework design

The front-end of pathway curator system is based on a high-performance front-end framework React. In order to effectively implement the front-end user requirements, this system uses a component-based programming design pattern, in which one by one independent requirements are designed into one by one equally independent components, and finally these components are assembled together to realize a complete system. Component programming has many advantages, for example, highly cohesive, code reusing and free combination of components. component-based programming design pattern diagram is shown as figure 2-2-1.

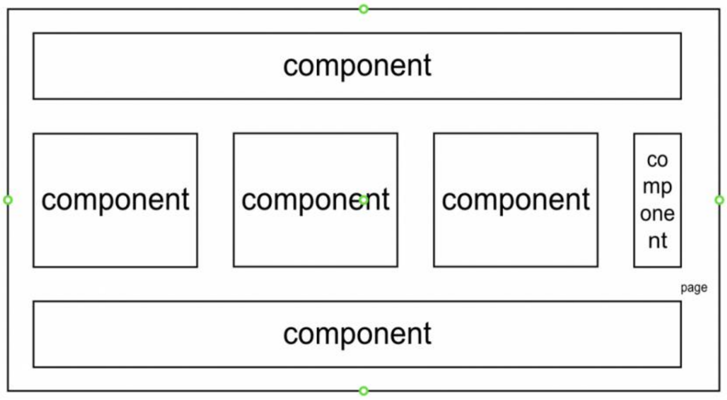


Figure 3-2-1 component-based programming design pattern diagram

The component-based programming design pattern has the following four advantages.

1. Highly cohesive: Organizing some related functions together to encapsulate everything, and in the case of components, it could be related functional logic and static resources: JavaScript, HTML, CSS, and images, etc. This is what we call cohesion. This approach will make the component easier to maintain and, having done so, the component will be more reliable. It also makes the functionality of the component clear and increases the possibility of component reuse.

2. Reuse：The functionality of each individual component is apparent, as is the implementation, and the API is simple to grasp. As a result, component re-usability is considerably enhanced. Development efficiency can be considerably increased by creating reusable components.

3. Composable：The component-based architecture makes it easier to combine components into new ones. This design makes components more independent and allows for better utilization of functionality built and exposed in other

components. It increases the flexibility of front-end development and greatly reduces development costs.

3.2.2 Back-end framework design

The back-end of pathway curator system is based on a high-performance back-end framework KOA. In order to effectively improve the efficiency of back-end development and achieve the back-end user requirements, the system adopts the commonJS modular design, a single file is a module, while a single function is a module. The module is loaded using the require method, which reads a file and executes it, and finally returns the exports object inside the file. This greatly improves the maintainability of the code and greatly reduces the coupling between functions. It makes the whole project very clean and powerful. commonJS design pattern diagram is shown as figure 3-2-2

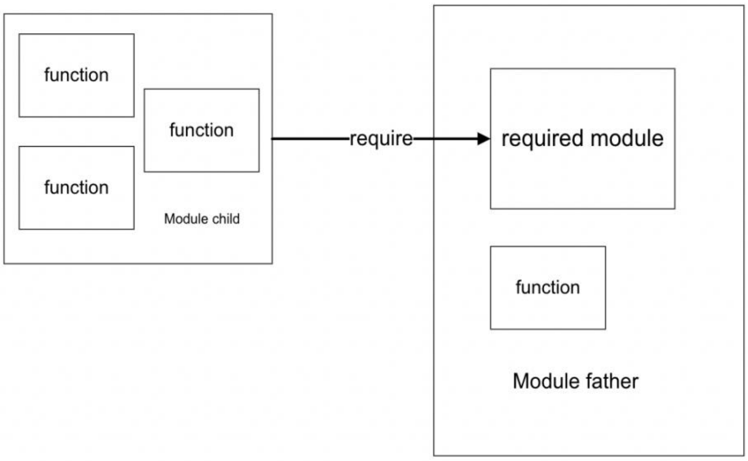


Figure 3-2-2 commonJS design pattern diagram

The commonJS modular design pattern have following two advantages.

1. Server-side modules are easy to reuse, which can reduce development

time and cost.

2. There are already nearly 200,000 available module packages in NPM, and using these packages can greatly improve development efficiency.

3.2.3 Database design

The database design is crucial to a system, and how to design the database well is also a key consideration in the development of this system. In order to store data better and interact with data better, this system adopts the design pattern based on MySQL relational database.

The objects managed by this pathway curator system are various kinds of data, so the core of the system architecture is the process of modeling the data. In this subsection, a data flow diagram will be used to describe the data flow of the system, as shown in Figure 3-2-3.1

1. User Login: When a user logs in for the first time, the system allocates the user an ID and simultaneously enters the user ID and related user information into the user table. Conditionally, the received user table information will also flow into the login management system.
2. History system: Using the current user ID, all of the user's historical items are retrieved from the item database, and the results are returned to the history system. At the same time, the information in the project table will flow conditionally into the historical system based on the retrieval.
3. Upload pathway diagram system: After uploading the user ID and

pathway diagram into the pathway diagram system, the pathway diagram will first enter the predicted pathway diagram model to begin pathway diagram analysis, and the results generated by the pathway diagram model will flow into the pathway diagram, gene table, and intergenic relationship table, along with the user ID and related user data.

4. Result display system: The recovered intergenic connection table as well as relevant data from the gene table will be sent into the result display system. The result presentation system can also update the related error data by transferring changed gene data and intergenic relationship data into the gene table and intergenic relationship table.

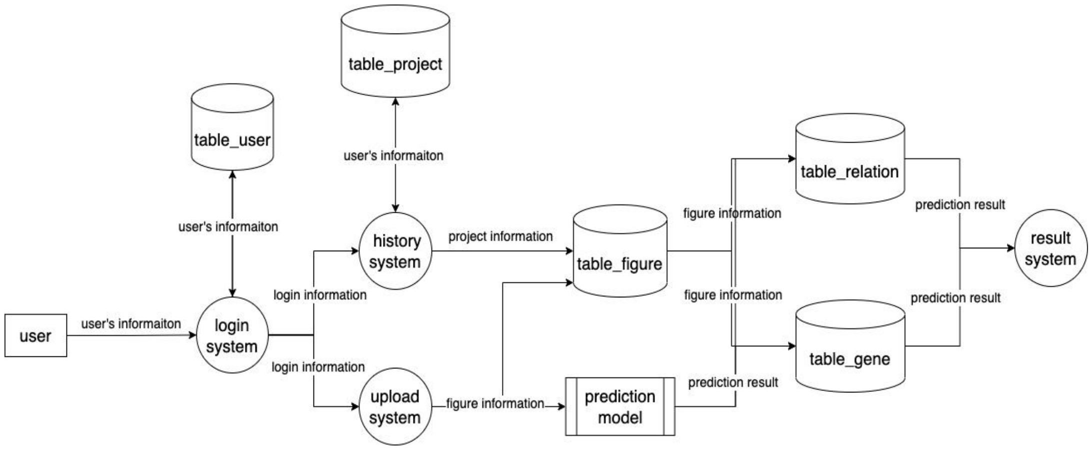


Figure 3-2-3.1 System data pipeline diagram

According to the requirement analysis and system design, it is especially important to design a database that can meet the functions of storing, querying, modifying and deleting the pathway map identification results. The database of this system mainly contains several tables, which store user account information, submitted work information, pathway map information, gene information, relationship information, literature information, OCR result table, gene dictionary table, etc. The Entity Relationship (ER) diagram of the database is shown in Figure 3-2-3.2.

1. User: User information schema includes four parameters, which used to store Users' information.
   a) U_name: type is "varchar(20)", store user's ID which is system generated.
   b) Pass_word: type is "varchar(20)" , store user's account's password which is set by user.
   c) Email: type is "varchar(20)", store user's email which will receive the result of the job.
   d) Last_login: type is "timestamp", store the last time of user access pathway curator system.
2. Job: Job information schema includes seven parameters, which used to store the submitted jobs' information.
   a) Job_id: type is "int", assign IDs to the jobs in the order they are submitted.
   b) Fig_id: type is "int", assign IDs to the figures in the order they are submitted.
   c) U_name: type is "varchar(20)", store users' ID in here, in order to select the history job by user_ID.

d) Start_time: type is “timestamp”, store the time of the job start to be analyzed.

e) End_time: type is “timestamp”, store the time of the job completed analysis.

f) IP: type is “varchar(20)”, store this IP address.

g) Country: type is “varchar(20)”, store the country according to the IP address.

3. Figure: Figure information schema includes eight parameters, which used to store the information of figures which in the job.

a) Fig_id: type is “int”, assign IDs to the figure in order they are submitted.

b) Paper_id: type is “int”, assign IDs to the paper in order they are submitted.

c) Fig_link: type is “varchar(20)”, store the source link where the figures came from.

d) Height: type is “int”, store the height of the figure.

e) Width: type is “int”, store the width of the figure.

f) Fig_title: type is “varchar(20)”, store the title of the figure.

g) Fig_caption: type is “text”, store the caption of the figure.

h) Fig_path: type is “varchar(20)”, store the path of the figure in the server machine.

4. Gene: Gene information schema includes seven parameters, which used to store the result of extracted genetic information

a) Gene_id: type is “int”, store the IDs of the genes in order they are added.

b) Fig_id: type is “int”, store the IDs of the figures in order they are added.

c) Dict_id: type is “int”.

d) Ocr_id: type is “int”.

e) Gene_Bbox: type is “varchar(20)”,store the location of the box for the gene entity in the figure.

f) Is_match: type is “int”, if the gene name in the result information can be found in the table gene_Dictionary, Is_match will be 1. On the contrary, Is_match will be 0.

g) Gene_name: type is “varchar(20)”, if the gene name of the result information can be matched with the gene name in table gene_Dictionary, this Gene_name will be the official gene name listed in the table gene_Dictionary. On the contrary, it will be the extract gene name (the gene name from result directly)

5. Relation: Relation information schema includes seven parameters, which used to store the result of extracted genetic relation information.

a) Relation_id: type is “int”, store the IDs of the relation in order they are added.

b) Activator: type is “int”, from the gene_id in table Gene.

c) Receptor: type is “int”, from the gene_id in table Gene.

d) Fig_id: type is “int”, store the IDs of the figures in order they are added.

e) Symbol_BBox: type is “varchar(20)”, store the location of the box for the relation symbol.

f) Relation_BBox: type is “varchar(20)”, stores the location of the box for the entire relationship.

g) Relation_type: type is “varchar(20)”, store the relation type of each relationship entity in the figure.

6. Gene_dictionary: Gene_dictionary information schema includes seven parameters, which is used to match the gene names extracted from the analysis, remove the non-genetic names and leave the correct genetic information. Following are the three key parameters. This dictionary is based on HUGO database.

a) Dict_id: type is “int”.

b) Gene_name: type is “varchar(20)”.

c) Alias_name: type is “varchar(20)”.

7. Ocr: Oct information schema includes four parameters, which is used to store the result of Ocr analysis. Following are two key parameters.

a) Ocr_id: type is “int”, assign IDs to the paper in order they are submitted.

b) Ocr_reault: type is “varchar(20)”, store the gene name from the OCR result directly.

8. Article: Article information schema includes six parameters, which is used to store the information of the paper which includes the submitted figures.

a) Author: type is “varchar(20)”, store the name of the first author.

b) Publication_year: type is “int”, store the publication year.

c) Key_words: type is “varchar(20)”, store the key words in this paper.

d) Abstract: type is “text”, store the abstract of this paper.

e) Paper_id: type is “int”, assign IDs to the paper in order they are submitted.

f) Title: type is “varchar(20)”, store the title of this paper.

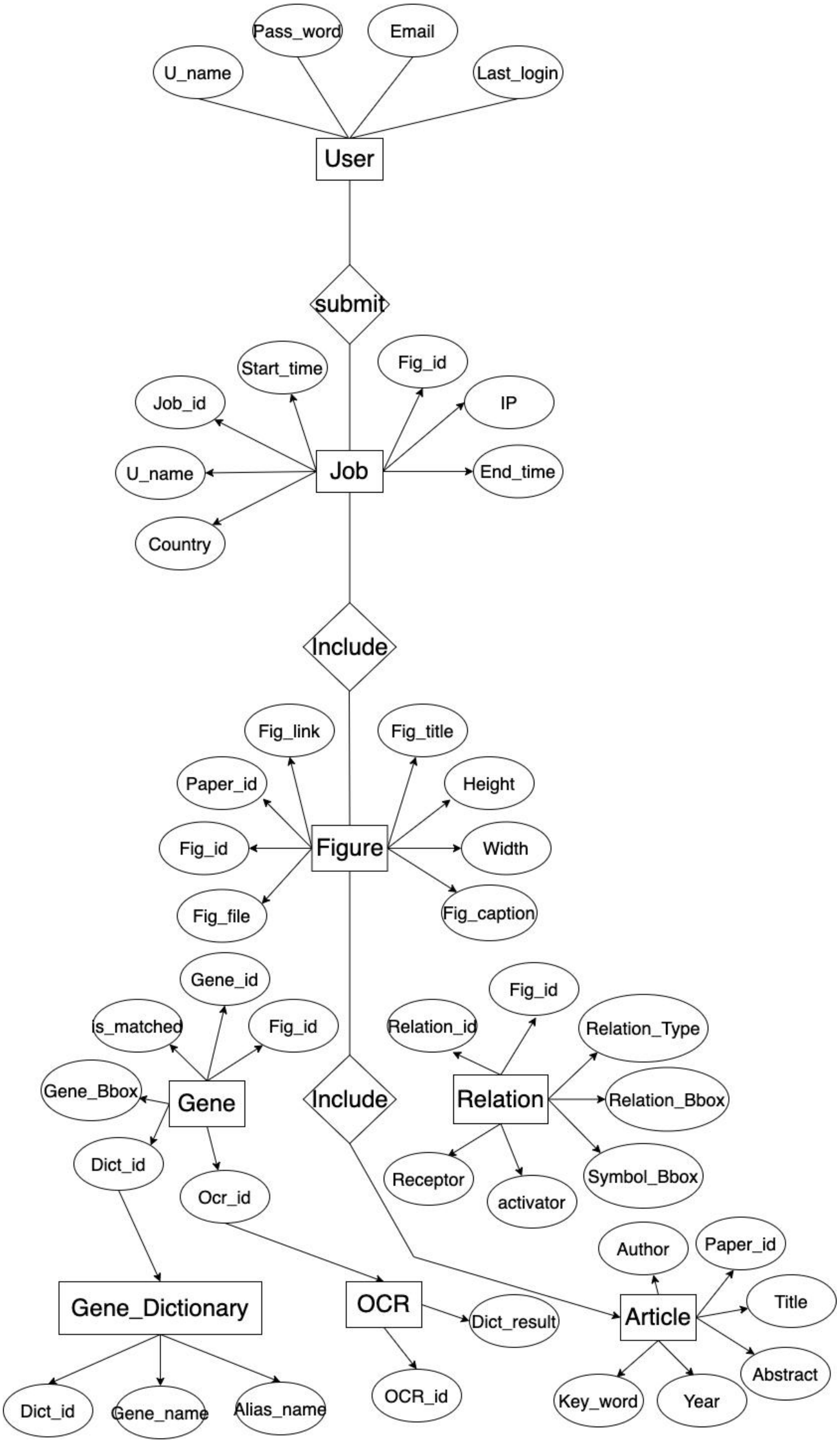


Figure 3-2-3.2 System data ER diagram

3.2.4 Restful API design

The preceding section introduced the front-end and back-end framework designs, and this section will explain how the system's front and back ends communicate.

This paper uses the design pattern of separating the front and back ends, so the coupling between the front and back ends is zero, considerably improving deployment and development efficiency. The system is built in the style of a restful API. RESTFUL is an HTTP-based web application design and development technique that can be described in XML or JSON format. RESTFUL is appropriate for mobile Internet vendors as a business interface scenario, achieving the function of third-party OTT calls to mobile network resources, with action kinds of Add, Change, and Delete the invoked resources.

Restful API is a straightforward API design technique based on the HTTP protocol. The heart of Restful is that everything is a "resource," that all HTTP actions should be modified and processed on the relevant resource, and that the API is the resource management operation, and that this exact activity is indicated by the HTTP action.

To express the addition, deletion, and checking of resources, use HTTP GET, POST, DELETE, and PUT.

1. GET: Read

2. POST: Create

3. PUT: update

4. DELETE: Delete

The Pathway curator system includes a plethora of APIs, of which I will highlight a few below.

1. /predict: It is a POST request. It will Receive a form-data which includes an image file, a user_name and a job_name. It will return a JSON includes gene entity information, relation entity information.

2. /getAllHistoryInfo: It is a GET request. It will Receive a form-data which includes a user-name. It will return a JSON file containing all of the user's submitted jobs.

3. /get_numbers: It is a GET request. It will receive nothing. It will return a JSON includes the number of users and figures.

4. /delete_history: It is a DELETE request. It will get form-data with the job id, which should be removed. It will return a JSON object containing the remove action status.

3.3 Development Environments and Tools

The web server is developed based on back end framework “NodeJS”, database tool” MySQL” and front end framework “React”. The main programming languages are “JavaScript”, “HTML5”, “CSS3” and “SQL”. Tools used in system are shown as the figure 3-3

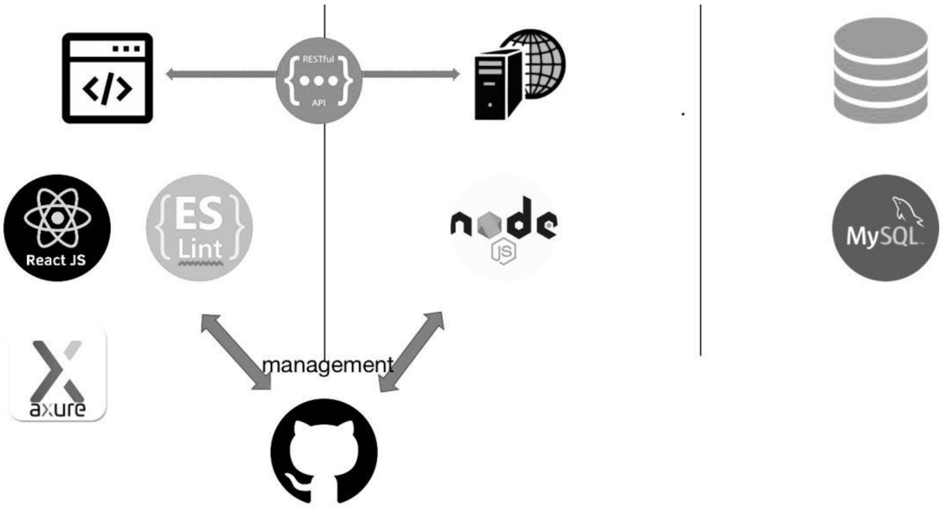


Figure 3-3 Tools used the system

3.3.1 React

React is a JAVASCRIPT library for building user interfaces. It is mainly used to build UI and is the V (view) in MVC. React started as an internal Facebook project to build the Instagram website and was open sourced in May 2013. React has a high performance and very simple code logic, and more and more people have started to pay attention to it and use it.

Following is seven advantages of React framework.

1.Declarative design -React uses a declarative paradigm to easily describe applications.

2.Efficient -React minimizes interaction with the DOM by mimicking the DOM.

3.Flexible -React can work well with known libraries or frameworks.

4.JSX - JSX is an extension to JavaScript syntax. react development does not necessarily use JSX, but we recommend it.

5.Components - Building components with React makes it easier to reuse code, and can be used well in large projects.

6. One-way responsive data flow - React implements one-way responsive data flow, which reduces repetitive code, which is why it is simpler than traditional data binding.

3.3.2 NodeJS

NodeJS is a platform based on the Chrome JavaScript engine that runs JavaScript on the server side. It is an event-driven I/O server-side JavaScript environment based on Google's V8 engine, which executes Javascript quickly and efficiently.

3.3.3 MySQL

A database is a repository for organizing, storing, and managing data based on a data structure. Each database contains one or more APIs for generating, accessing, managing, searching, and copying data. Mysql is a relational database, or a database based on the relational model, that handles data in the database using mathematical principles and methods such as set algebra.

3.3.4 Management tool GitHub

To boost development efficiency, we usually need to undertake project version control and code management throughout development. As a code management tool for my project, I chose GitHub, a hosting platform for open source and private software projects. I separated the origin into three branches: origin/dev, origin/test, and origin/production. The development process is to do development on branch1, such as modifying features, adding features or deleting features, merging branch1 to branch2 after development is completed, then deploying branch2 to the development server to test whether the system works properly, merging branch2 to branch3 after successful deployment, and finally deploying branch3 to the public server. Code management processing diagram is shown as figure 3-3-4.

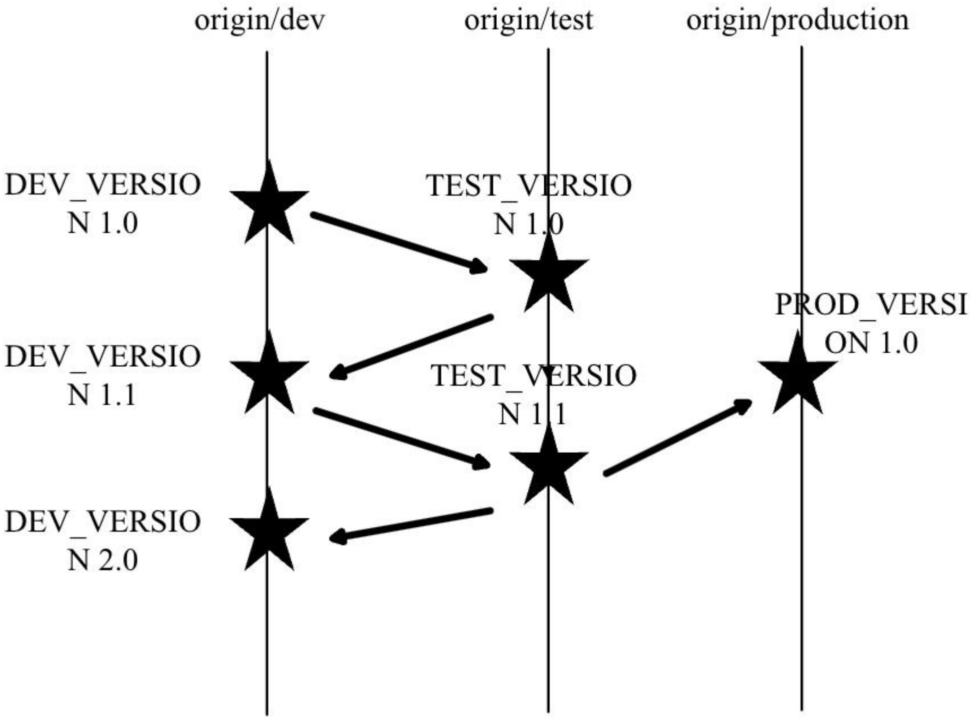


Figure 3-3-4 Code management processing diagram

4. SERVER SYSTEM IMPLEMENT

The previous two chapters covered the necessary basic knowledge and system design foundation. This chapter will detail how the system is implemented based on this. The system is divided into four modules: the user login module, the prediction module, the history module, and the result display module.

4.1 Run-time environment

This is a lightweight system. The development environment of the system is shown in Table 4-1.

Development Environment	**Description**
Operating system	Linux/Windows/IOS
Processor	Intel i5-8565U 1.60GHz and above
Running memory	1GB or more
Storage space	10 GB and above
Development tools	Visual studio code, GitHub
Required python packages	PyTorch 1.5, detectron2
Python environment	Python 3.6.13

Table 4-1 Run-time environment

4.2 User login module

The system will automatically assign accounts to users during the process of using the system, so that they can use the system functions without registering; if they have already logged into the system, they can use the system functions directly, and they can view the historical submission records under this account, as shown in Figure 4-2.1.

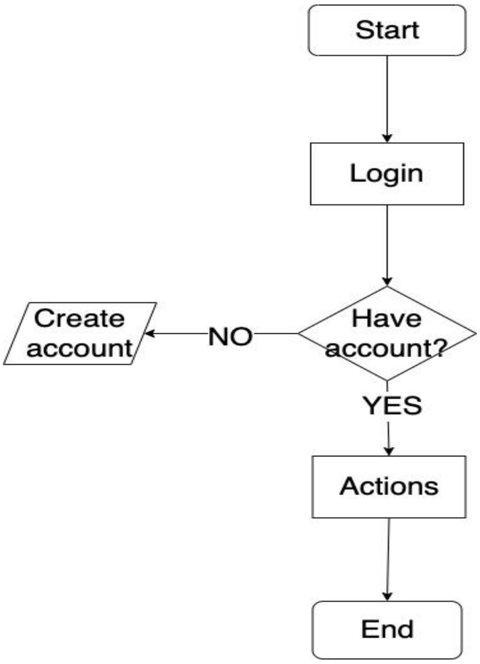


Figure 4-2.1 Login module flow chart

When user access the pathway curator system, the system will first start the Login Action, get the user ID from user's local storage, if there no user id can be found in users local storage, the system automatically creates the user ID which is generated by login time and a secure random number. Then system will check the user table in the query server to see if the ID is legal, return the query information to the user, store the user information in Session, and finally display the main interface after successful login. Sequence diagram as shown in Figure 4-2.2

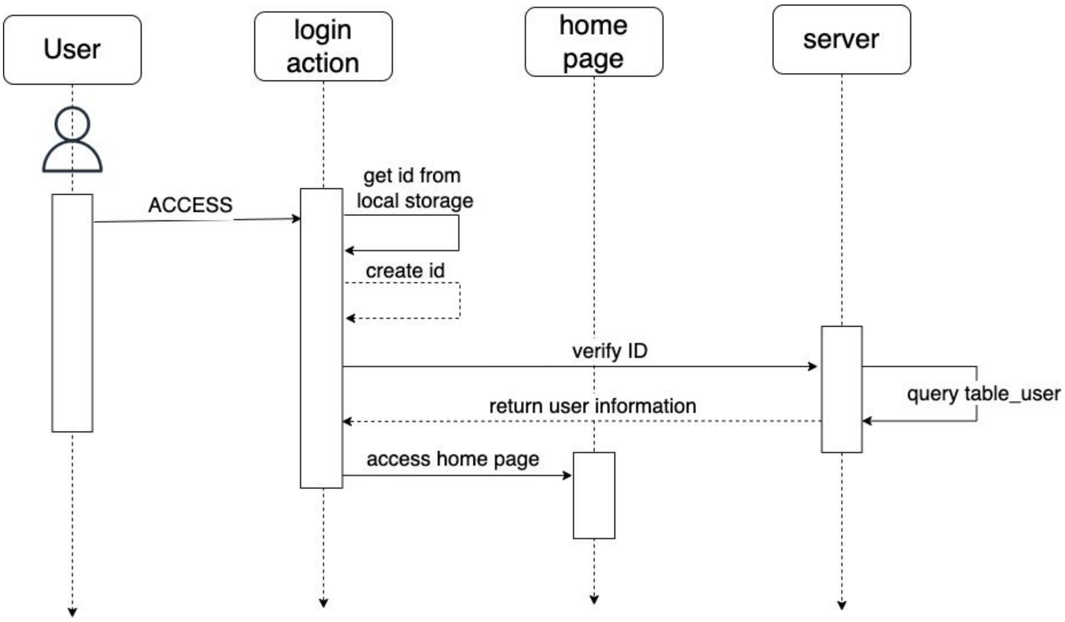
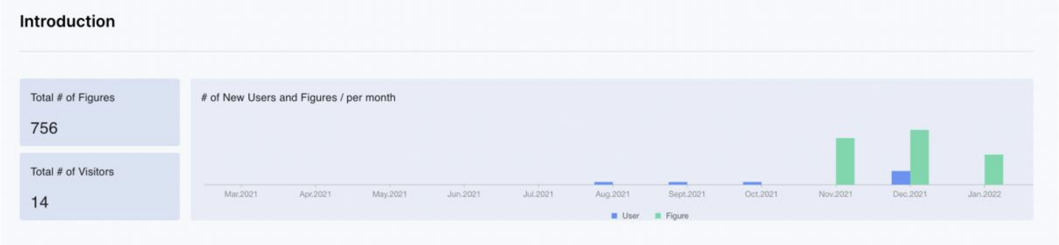


Figure 4-2.2 User login sequence diagram

The home page of the pathway curator system includes two parts, the model of show total number of the users and the model of introduction of the pathway curator algorithm. They are shown in Figure 4-2.3 and figure 4-2.4. The home page shows the flowchart of the tool's work, the current number of system users, and the current number of images received by the system. The flowchart contains input pathway map, gene detection, relationship detection and result matching and collation. The home page is not only a guide to the system, but also facilitates users to better understand the design idea and implementation principle of the tool.


Introduction

Total # of Figures

756

Total # of Visitors

14

Figure 4-2.3 Total number of the users and figures

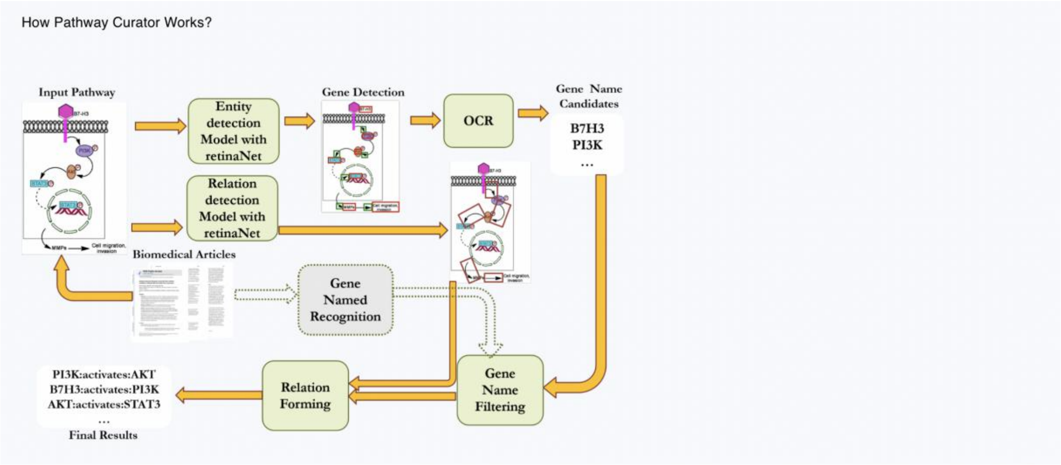


Figure 4-2.4 introduction of the pathway curator algorithm model

In this user login module, the module will call the api named "/verify_user_id". The information of this API as shown in the table 4-2.

Name	URL	Verb	Description
Verify user's ID	/verify_user_id	POST	Detecting the legitimacy of user IDs

Table 4-2 API of verify ID

4.3 Uploading and predicting module

The upload and prediction pathway map function is used to upload data. Users can upload an image or a zip file containing many images on this page, and the image or zip package is required. Also the user can optionally upload some meta-data about the uploaded figures, like figures title, the source link of the target figure or the paper's title.

The user first uploads the image file and metadata to the system, after which the system transfers these data to the server, saves the image under the specified directory, and then passes the path where the image is saved to the algorithm model to analyze the target image. The system will save the result, along with the image and metadata, to the database after it has been obtained. Uploading and predicting module's sequence diagram as shown in figure 4-3.1.

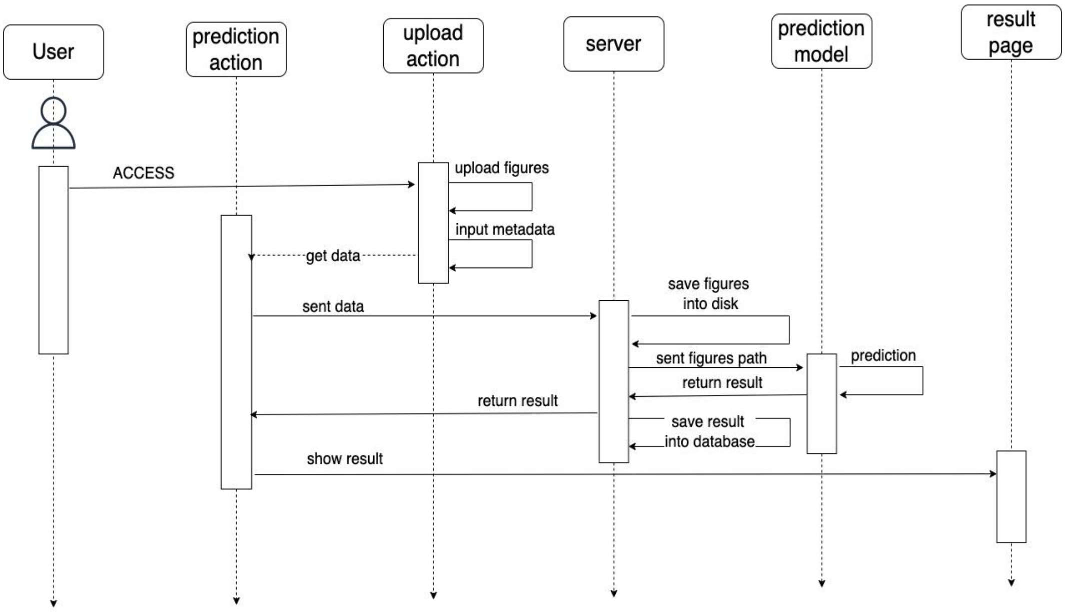


Figure 4-3.1 Uploading and predicting module's sequence diagram

The system provides two types of file uploads, single upload and bulk zip upload, to meet the usage needs of different researchers in different ways. In addition, the system is also designed with two upload methods, drag-and-drop file upload and click-select file upload, to meet the user's usage habits as much as possible. The upload pathway map interface is shown in Figure 4-3.2.

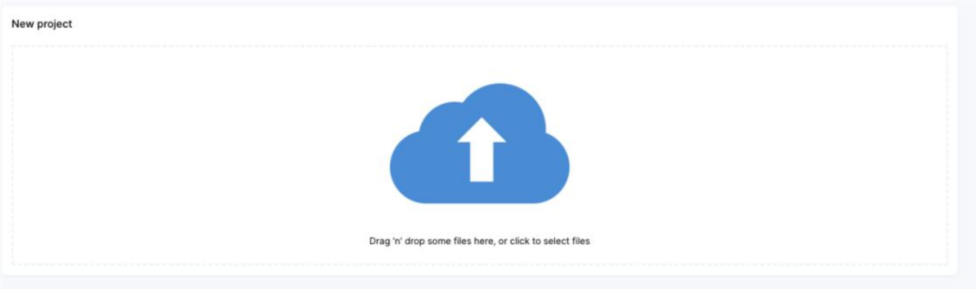
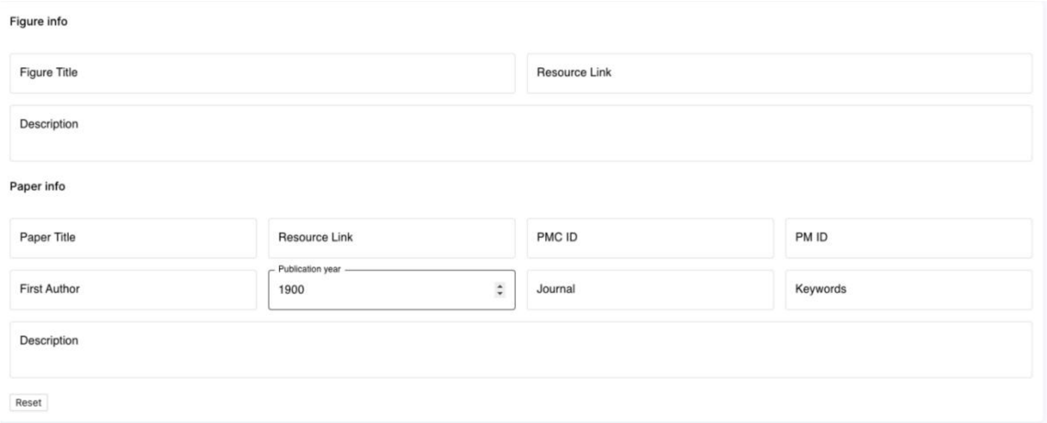


Figure 4-3.2 Uploading figures component

The system provides two kinds of optional information boxes for uploading meta-data of access diagram files as shown in Figure 4-3.3, which are the access diagram information fill box and the paper information fill box. The access diagram information includes access diagram title, access diagram link, access diagram description; the paper information includes paper title, paper link, PMC ID, PM ID, first author, publication year, journal name, keywords and paper description. All the above information is optional, in order to collect more comprehensive information and help the system to develop and improve the subsequent extension functions, which can provide more and more comprehensive ways to retrieve information.



Publication year 1900

Figure 4-3.3 Uploading meta-data component

Uploading and predicting module includes one API named "/predict". The information of this API as shown in the table 4-3.

Name	URL	Verb	Description
Predicting	/predict	POST	Uploading the figures and the metadata to server, then call the prediction function to

			analysis the target figure

Table 4-3 Prediction API

4.4 History module

To view the history submission records, the system searches the server using the user ID, and the server delivers data that meets the query conditions, and all of the current user's history records are presented in the history interface. Furthermore, it offers the function of querying the predicted result of a specific pathway map based on the pathway map ID, and the server responds to the result display interface based on the gene table and the gene connection table based on the pathway map ID, as shown in Figure 4-4.1.

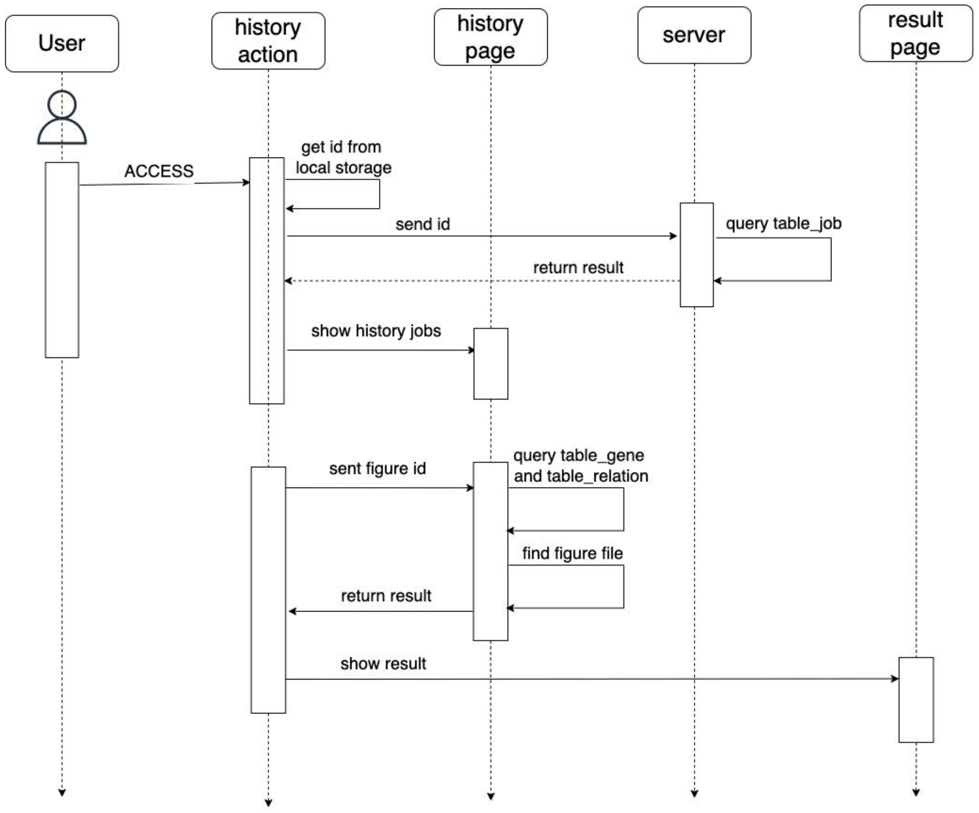
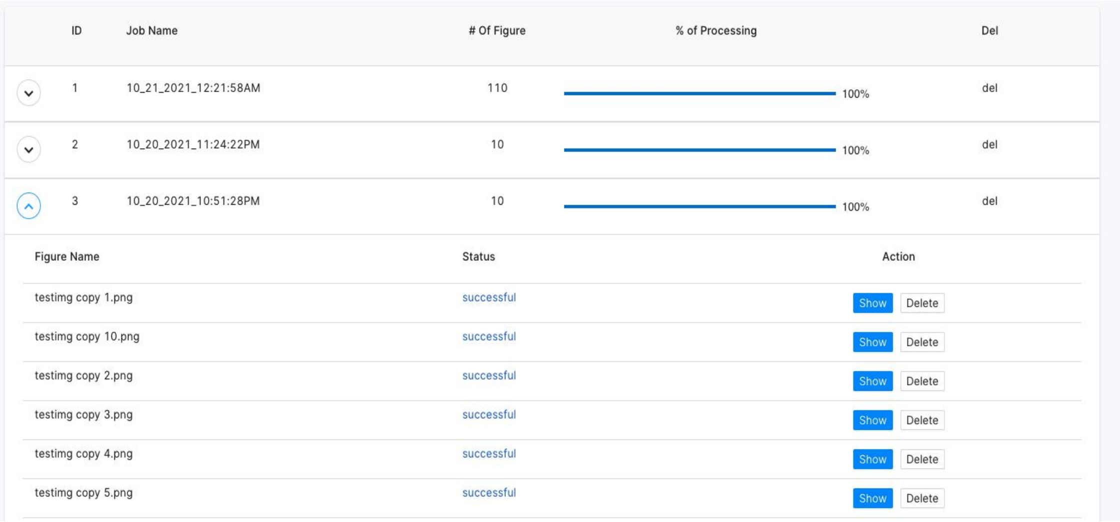


Figure 4-4.1 History sequence diagram

The view history screen is shown in Figure 4-4.2, where users can view all the pathway diagrams they have submitted, including the name, metadata, and prediction results of the pathway diagrams. Users can also delete the specified jobs.



	ID	Job Name	# Of Figure	% of Processing	Del
⌄	1	10_21_2021_12:21:58AM	110	100%	del
⌄	2	10_20_2021_11:24:22PM	10	100%	del
⌃	3	10_20_2021_10:51:28PM	10	100%	del

Figure Name	Status	Action
testimg copy 1.png	successful	Show Delete
testimg copy 10.png	successful	Show Delete
testimg copy 2.png	successful	Show Delete
testimg copy 3.png	successful	Show Delete
testimg copy 4.png	successful	Show Delete
testimg copy 5.png	successful	Show Delete

Figure 4-4.2 History component

History module includes three API named ”/get_result”, “/get_all_history_info”, ”/delete_history”. The information of these APIs are shown as table 4-4.

Name	URL	Verb	Description
Get Result	/get_result	GET	Get the result generated by algorithm model as a JSON.
Get History jobs	/get_all_history_info	GET	Get all of the jobs that user has

			submitted.
Delete history jobs	/delete_history	DELETE	Remove the job user want to delete.

Table 4-4 History API

4.5 Result display module

Result display model is used for displaying results. Users can examine the analysis results in two ways. The first is that after submitting an item on the analysis page, the system will automatically jump from the analysis page to the result page and display the received findings to the user. The second is that if the user hits the view item button on the history page, the system will instantly navigate to the result display page and visually present the obtained results to the user. Although the front-end logic of these two routes differs, their back-end logic is the same: both retrieve appropriate result information in the database based on the project ID and then return to the front-end visual display.

In the prediction result interface, the gene entities are circled in blue boxes and the relationship entities are circled in red boxes on the right side of the pathway map, and the corresponding gene information and relationship information are recorded in the table on the left side, as shown in Figure 4.13. At the same time, the predicted results can be customized by clicking on the gene name; clicking on the "Show Detail" button to view the detailed information of the pathway map; clicking on the "Dictionary" drop-down box to specify the gene dictionary for gene screening; clicking on the "Dictionaries" drop-down box to specify the gene dictionary for gene screening; and clicking on the "Show Detail" button to view the details of the pathway map. Click "Dictionary" drop-down box to specify the gene dictionary for gene screening; click "Download" to download the JSON file corresponding to the prediction results.

Moreover, on this page, users can also change, add, or delete results.

Modify the result: After double-clicking the item in the gene table and the relationship table, you can modify it directly in the input box.

Add results: click on the add gene or add relationship button and modify directly on the displayed image.

Delete result: Click the button to delete the item in the gene table and relationship table.

After user modify the result, the new result data will be sent to back end and update the data in the database. Result display module’s sequence diagram as shown in figure 4-5.1.

On this page have four component, they are figure component, gene table component, relation table component and mate-data component. As shown in figure 4-5.2, figure 4-5.3, figure 4-5.4, figure 4-5.5

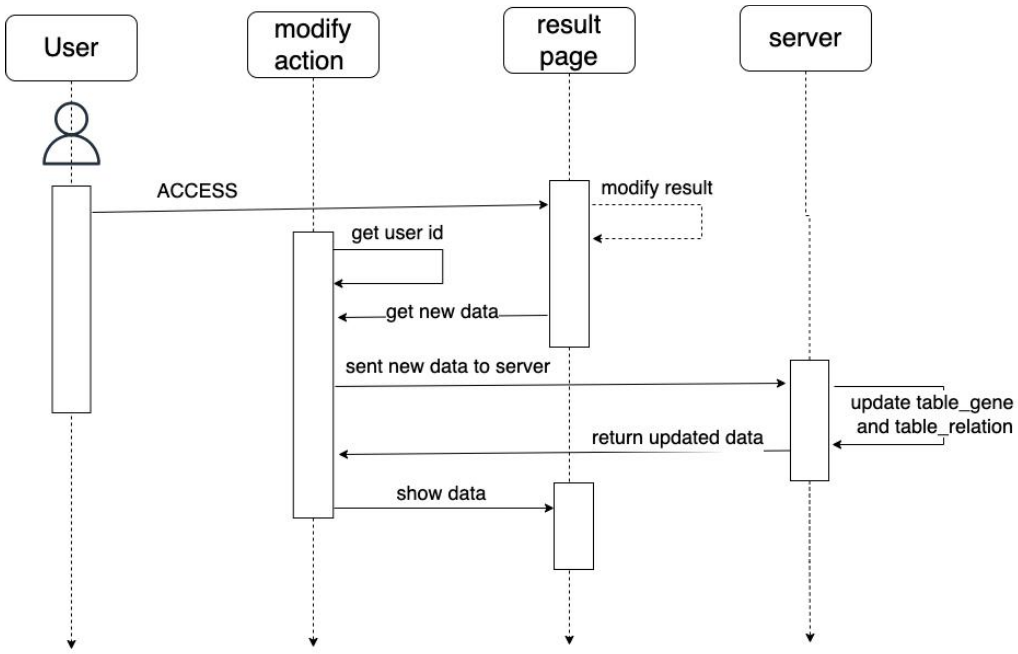


Figure 4-5.1 Result display sequence diagram

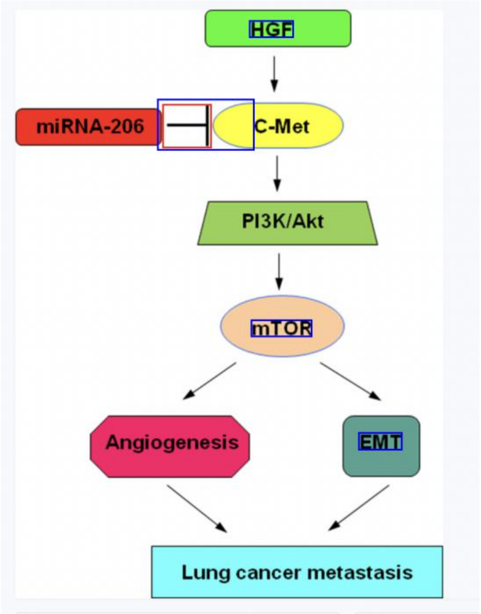
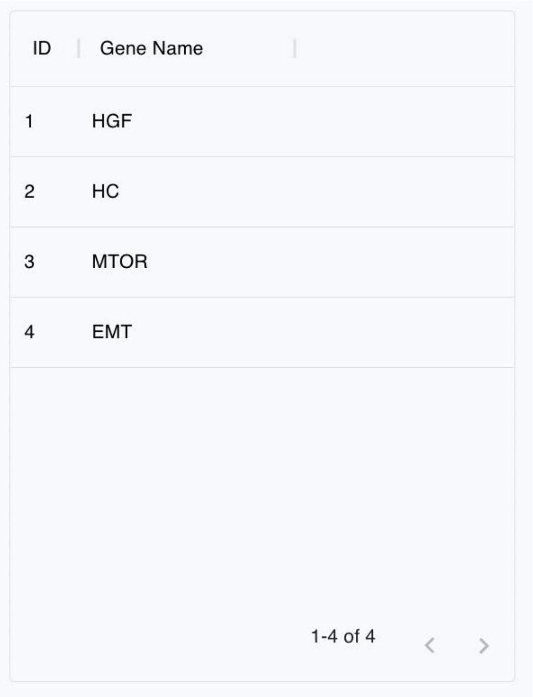
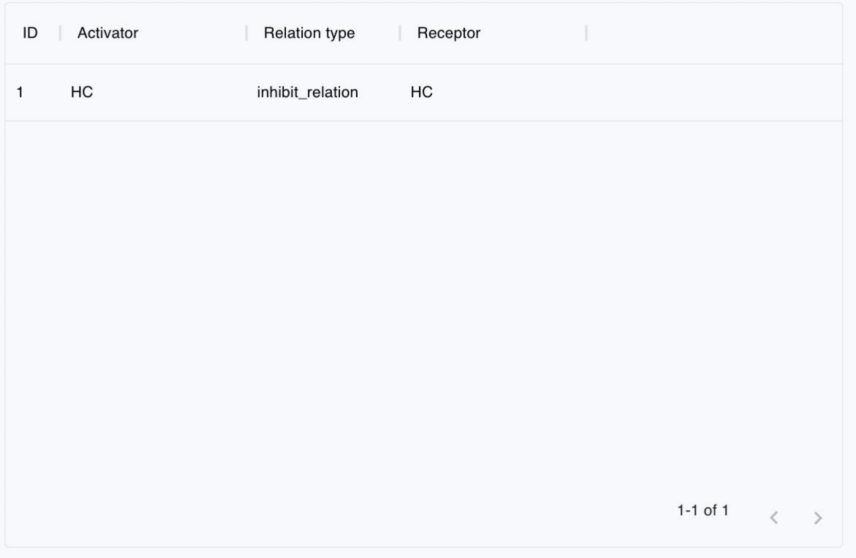


Figure 4-5.2 figure component



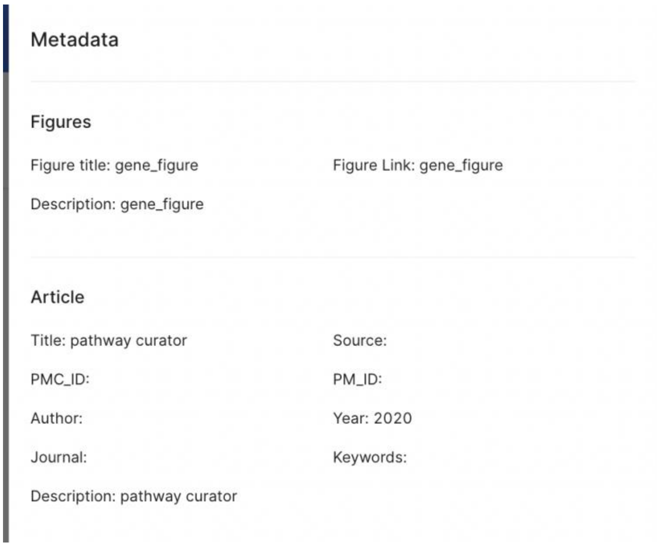
ID	Gene Name
1	HGF
2	HC
3	MTOR
4	EMT

Figure 4-5.3 gene table component



ID	Activator	Relation type	Receptor
1	HC	inhibit_relation	HC

Figure 4-5.4 relation table component



Metadata

Figures

Figure title: gene_figure

Figure Link: gene_figure

Description: gene_figure

Article

Title: pathway curator

Source:

PMC_ID:

PM_ID:

Author:

Year: 2020

Journal:

Keywords:

Description: pathway curator

Figure 4-5.5 meta-data component

Result display module includes six API named ”/add_gene_info”, “/edit_gene_info”, ”/delete_gene_info”, “/add_relation_info”, “/download”, “/edit_relation_info” , “/delete_relation_info”. The information of these APIs are shown as table 4-5.

Name	URL	Verb	Description
Add gene info	/add_gene_info	POST	Add gene information to database
Edit gene info	/edit_gene_info	POST	Update gene information in database
Delete gene info	/delete_gene_info	DELETE	Delete gene information in database
Add relation info	/add_relation_info	POST	Add relation information in database
Edit relation info	/edit_relation_info	POST	Update relation information in database
Delete relation info	/delete_relation_info	DELETE	Delete relation information in database

Download info	/download	GET	Get result information in database

Table 4-5 Result display module APIs

4.6 Website deployment

The system in this article uses the MVC development paradigm, and it is totally separated from the front and back ends, so when deploying it, it is divided into four parts: front-end layer, back-end data processing layer, back-end logic algorithm layer, and database layer.

4.6.1 Server configuration

The configuration parameters of the server machine is shown as the table 4-6-1.

Parameters	Value
Operation system	Ubuntu 18.0
Size of disk	100G
Memory	4G
CPU	

Table 4-6-1 configuration of the server

4.6.2 Running environment configuration

The process of configuring the running environment is divided into three steps: back-end environment setup, database configuration, and algorithm environment configuration.

4.6.3 Front-end environment configuration

Front-end of this system is built on the React front-end framework, while deploying the front-end system, the static files must first be converted and then deployed on the server apache2.

```
1.  Convert front-end files to static files：

    NPM build
```

4.6.4 Back-end environment configuration

Back-end of this system is built on the Koa front-end framework, while deploying the Back-end, put the files of Back-end system on the server machine then run it. In order to run the back-end forever. This paper uses a npm package named forever. App.js is the entrance of the back-end system.

```
1. Run Back-end system：

    Forever start app.js
```

4.6.5 Algorithm model environment configuration

Algorithm model of this system is based on python3.7 and some python packages, while deploying the model, install the python3.7 and these python package like detectron2 and PyTorch 1.5 is required.

5. SYSTEM TESTING

The process of adopting software tools is complex and error-prone, and failure to identify and eliminate such problems on time can result in significant time and money waste later on. Software testing is a critical component of the software development lifecycle, as it lowers the impact of preventable errors and so reduces costs. As a result, the experiments were carried out in order to test the tool from various elements of the program.

5.1 Common test method

The purpose of software testing is to find errors in program execution, so it is necessary to use fewer test cases to find out various potential errors and defects in the software, to meet the specified requirements or to figure out the difference between expected and actual results, in order to ensure the quality of the software. Common testing methods include black box testing, white box testing, interface testing, etc.

1. Black Box Testing

Black-box testing treats the tested software as a "black box" with input and output functions. The tester only needs to know what functionality the software under test should implement, not how it does it. Therefore, black-box testing is also known as specification-based testing techniques or input-output driven testing techniques.

2. White Box Testing

White-box testing treats the software system being tested as a "transparent box" where the tester understands the detailed code, the internal structure of the program and how it works. It is also known as a structure-based testing technique, because in white-box testing, the tester needs to understand how the software is implemented. In contrast to black-box testing, testers focus on how the software works in white-box testing.

3. Interface Testing

Interface testing is the process of testing software systems that use GUIs, mainly to test whether a product's graphical user interface meets its design requirements. This type of testing is common when designing scenarios such as checking image prototypes, the size of buttons, and the alignment of web pages. Since the design of GUI pages often requires consideration of user experience, resulting in complex interaction logic, comprehensive coverage and marginal scenarios need to be considered when writing test cases.

5.2 Functional Testing

The system designed and implemented in this paper was programmed mainly in JavaScript language. In terms of testing, a combination of black-box testing and GUI testing was chosen to conduct exhaustive testing of the core functions implemented by the tool.

In the testing process, corresponding test cases were designed for different function points to be tested. The prediction of the pathway diagram by using the tool is the core requirement required by the user. The specific test cases and the results of the corresponding tests are shown in Table 5-2.

ID	Function	Test cases	Input	Response	Output	Result
1	Upload pathway figures	Check did system receive the pathway figures	Pathway figures	Store the file into server machine	Response upload successful	Pass
2	Reset input data	Check if data is reset	Click reset button	Reset all of the data in this page	Reset input data	Pass
3	Upload paper information	Check if paper data is uploaded	Input paper data	Store paper data into database	Paper data stored successful	Pass
4	Predict pathway figures	Check if prediction successful	Click predict button	Call back end algorithm model	Generate result successful	Pass
5	View history jobs	Check did system list all of the past jobs	Click button history in menu	Select past jobs in database	List all of the past jobs in the system	Pass
6	Delete history jobs	Check did delete the jobs successful	Click the button delete in the history page	Remove the job in database	Remove the job successful	Pass
7	Download the result	Check could download the result as a JSON	Click the button download in result	Select and download the result in the database	Download the result as a JSON file	Pass

			page			
8	Update the result	Check did update succeed	Update the result in the result page	Update the result in the database	Update the result both in the database and in the result page	Pass
9	Show the total number of the users and jobs	Check does homepage shows the number of the users and jobs	Access into the home page	Select the total number of the users and jobs in the database	Homepage shows the correct total number of the users and jobs	Pass
10	View the mate data of the job	Check does system can show the mate-data of the jobs	Click the button “show mate sdata” in int result page	Select the mate-data in the database	Displaying the mate-data in the result page	Pass

Table 5-3 functional test

According to the results of functional tests, the function of this system is flawless. The system will then be stress tested and security tested.

5.3 Performance test

The performance of a website is critical for its operation, which is mainly decided by the server's performance and the optimization of the website itself. By running performance tests on a website, the developer can determine the number of concurrent users that the website can support and then optimize the website or the server again. Blazameter was used as a testing tool in this post to test the performance of the website during the development phase.

The website's performance is evaluated primarily through two factors: the response speed time test and the stress test.

5.3.1 Response speed test

The network environment in which a user connects to a Web application determines the speed with which he or she connects. Users must wait for the page to load when accessing the system of this article, but if the response time of the Web system is too long (for example, more than 5 seconds), users will leave because they do not have the patience to wait, so the response speed of the page is a very important part of the user experience. Furthermore, some pages have a timeout limit, and if the response speed is too sluggish, it may cause data loss, resulting in viewers not seeing the actual page and, as a result, error reporting.

When the number of users accessing the system at the same time is small, the page response speed of the system in this article is quick, thus it is tested numerous times during the development process for the case of large concurrency.

Two response speed experiments were carried out in this research, and their original test configuration were described at the following.

	Response speed test 1	Response speed test 2
Max number of users	20	50
Total duration time	9min	6min

Ramp up duration	3min	3min
Ramp up of steps	5	10
Location	US:50%, Asian:50%	US:50%, Asian:50%

Table 5-3-1.1 Response test configuration

1. Max number of users: Maximum number of users using the site at the same time, maximum concurrency.

2. Total duration time: Duration of continuous online access to the system for all users.

3. Ramp up duration: Duration of access after the user reaches the maximum number of users.

4. Ramp up of steps:Time of continuous increase of users until the maximum number of users.

5. Location: Regional distribution of the systems accessed.

Max users	20
Avg.Throughout	290.29Hits/S
Errors	0%
Avg.Response time	59.48ms
90% Response time	61ms
Avg.Bandwidth	971.80Kib/S

Table 5-3-1.2 Response speed test (Max number of user: 20)

Max users	50

Avg.Throughout	458.18Hits/S
Errors	11.99%
Avg.Response time	78.99ms
90% Response time	125ms
Avg.Bandwidth	1460Kib/S

Table 5-3-1.3 Response speed test (Max number of user: 20)

The above figure shows that when the number of concurrent users reaches 40, the website begins to report issues and lags at a specific rate. When the number of concurrent users approaches 50, the error rate rises to 10%. The explanation for this problem is most likely that the server's hardware equipment is insufficient to sustain the massive amount of concurrency, hence this work has conducted additional research on the server equipment.

5.3.2 Stress test

Stress testing is used to assess the system's constraints and fault resilience. System constraints and fault resilience, i.e. determining whether and under what conditions a web application will crash. What situations will cause it to crash. The web application will crash to some extent after a certain number of concurrent users, hence this must be tested before the website goes live.

The stress test is configured in the same way as the reaction time test shown as the table 5-3-2.1.

	Response speed test 1	Response speed test 2

Max number of users	20	50
Total duration time	9min	6min
Ramp up duration	3min	3min
Ramp up of steps	5	10
Location	US:50%,　Asian:50%	US:50%,　Asian:50%

Table 5-3-2.1 stress test configuration

The stress test in this article looked at four important server hardware factors.

1. CPU: The CPU utilization rate is the percentage of CPU resources used by running applications, suggesting that your machine is executing programs at any one time; the higher the occupancy rate, the more noticeable the website latency.

2. Network I/O： The higher the data reading and writing speed, the better the website performance.

3. Memory： Memory is an important part of a computer, also known as internal memory and main memory, which is used to temporarily store the computing data in the CPU. The higher the memory usage, the worse the website performance.

The above graphical data show that when the number of concurrent users hits 40, the server CPU consumption might reach up to 80%. Memory utilization is also 40%, which is twice as much as when the number of users is not 20. As can be observed, the modest server employed in this article can support a maximum of 40 concurrent users. This is consistent with the server configuration's maximum number of concurrent users.

In conclusion, Pathway curator system can support a maximum of 40 concurrent users.

5.3.3 Browser test

The Web client's most important component is the browser. Different vendors' browsers support Java, Java Script, ActiveX, plug-ins, and HTML specifications in different ways.

To test, I utilize the browser testing application "browserling." The testing cases are listed below.

1. Browser: Chrome
2. Browser: Edge
3. Browser: Firefox
4. Browser: Firefox
5. Browser: Opera

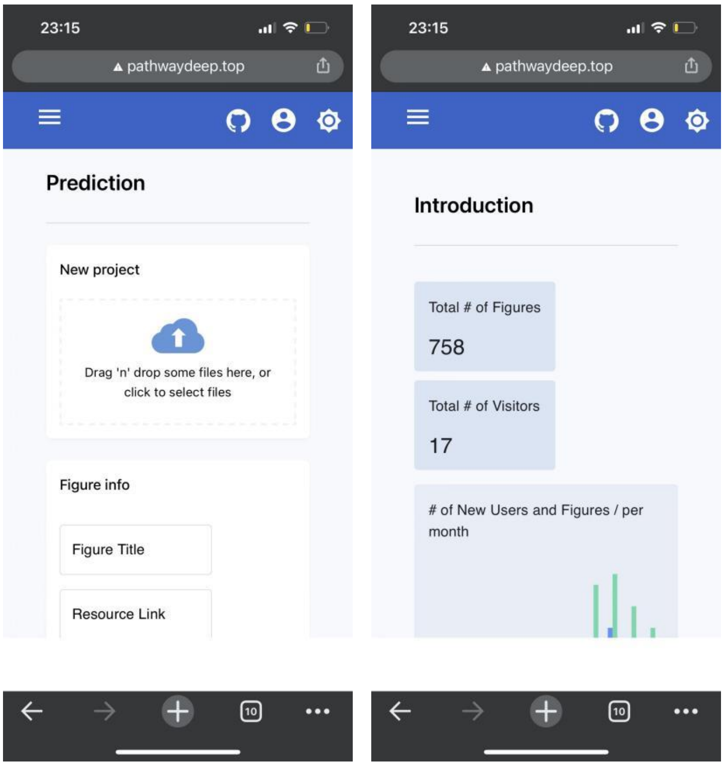
Each test case returned the correct page and results.

In conclusion, this system has been tested and found to be compatible with Chrome, Edge, Opera and Firefox.

5.3.4 Mobile test

In modern society, cell phones have become an important public tool in people's lives, so it is important for the system to support cell phone versions. So the cell phone test was also conducted.

After testing, found that all of the pages are working well in the mobile’s browser. Figure 5-3-4



Prediction

New project

Drag 'n' drop some files here, or click to select files

Figure info

Figure Title

Resource Link

Introduction

Total # of Figures

758

Total # of Visitors

17

# of New Users and Figures / per month

Figure 5-3-4 Mobile version

6. CONCLUSION AND FUTURE WORK

In this paper, we proposed a pipeline to extract genes and their gene interactions from pathway figures with deep learning models. We also created a high performance online web server named Pathway Curator, users can use this web server to mine the gene's information and gene's interactions from the pathway figures online.

This article also focuses on the website development process. KOA, a high-performance back-end framework, React, a front-end framework, and MySql, a database management tool, were used to create the site. It also follows the current development model of separating the front-end and back-end. Moreover, in this web server system, also used Restful API which can help Front and back-end to interaction data. I also gained a lot of knowledge from it. Each cycle of the software development lifecycle, including requirements analysis, project design, project execution, and project testing, is vital and cannot be overlooked.

The tool described in this research can achieve high target detection rates, but it is still a long way from flawless prediction of gene interactions. Target detection, text recognition, and gene relationship pairing are all integrated in this experiment, and the algorithmic models will be improved in the future to integrate all models into one end-to-end deep architecture with global optimization in the training phase to minimize error in each step. Second, various aspects of this website will be improved.

1. Improvements to the user administration mechanism： In this paper, the pathway curator server will give user a random ID. So that, When a user visits the site, the system first looks in the user's local storage for the ID; if no ID is found, the user is assigned a random secure ID, and the user is handled on that basis. If an ID is present, the user's data is controlled directly in the system. In the future a password for each ID and The function of retrieving ID according to user's email will be added in this system.

2. Improve the logic of the calling algorithm model： In this paper, pathway curator system use the node package “exec” to call the algorithm model which is based on python. Although it is a already a good way to call the model, it still need another system file management system to assist to achieve. In the future using parameter to transfer exchange input and output will be better.

3. improving the server's performance： During testing, it was discovered that the server on which the system was placed was underperforming, limiting the number of concurrent users it could support, hence using a server with higher performance in future will be better.

4. Improve the algorithm model: In this paper, entities and their relationships are extracted from the pathway map; in the future, we can add natural language processing-based methods to extract entity relationships for the paper's content, as well as mine genes and their interactions from images and texts of bio-medical articles.

REFERENCES

[1] Home-PubMed-NCBI [https://pubmed.ncbi.nlm.nih.gov/]

[3] WEI C H, ALLOT A, LEAMAN R, et al. PubTator central: automated concept annotation for biomedical full text articles [J]. Nucleic Acids Res, 2019, 47(W1): W587-W93.

[4] TURRIFF S L. Multiple Pathways to Bio-based Products [J]. Pulp Pap-Canada, 2011, 112(2): 16-7.

[5] OSHIMURA M, UNO N, KAZUKI Y, et al. A pathway from chromosome transfer to engineering resulting in human and mouse artificial chromosomes for a variety of applications to bio-medical challenges [J]. Chromosome Res, 2015, 23(1): 111-33.

[6] SHIN D, ARTHUR G, POPESCU M, et al. Uncovering influence links in molecular knowledge networks to streamline personalized medicine [J]. J Biomed Inform, 2014, 52: 394-405.

[7] AMMARI M, ARYAMONTRI A C, ATTRILL H, et al. Biocuration: Distilling data into knowledge [J]. Plos Biol, 2018, 16(4).

[8] THANINTORN N, WANG J X, ERSOY I, et al. Rdf Sketch Maps - Knowledge Complexity Reduction for Precision Medicine Analytics [J]. Biocomput-Pac Sym, 2016: 417-28.

[9] PENG H J B. Bioimage informatics: a new area of engineering biology [J]. 2008, 24(17): 1827-36.

[10] RODRIGUEZ-ESTEBAN R， IOSSIFOV I J B. Figure mining for biomedical research [J]. 2009.

[11] NEUMANN B， HELD M， LIEBEL U， et al. High-throughput RNAi screening by time-lapse imaging of live human cells [J]. 2006， 3(5)：385-90.

[12] RINALDI F， LITHGOW O， GAMA-CASTRO S， et al. Strategies towards digital and semi-automated curation in RegulonDB [J]. Database-Oxford， 2017.

[13] KUENZI B M， IDEKER T. A census of pathway maps in cancer systems biology [J]. Nat Rev Cancer， 2020， 20(4)： 233-46.

[14] JULIA P， AMARNATH G， MAYYA S， et al. BiologicalNetworks 2.0 - an integrative view of genome biology data [J]. 2010， 11.

[15] Pathway Commons， a web resource for biological pathway data. %J Nucleic acids research [J]. 2011.

[16] VARDAKAS K Z， TSOPANAKIS G， POULOPOULOU A， et al. An analysis of factors contributing to PubMed's growth [J]. J Informetr， 2015， 9(3)： 592-617.

[17] LI C， LIAKATA M， REBHOLZ-SCHUHMANN D. Biological network extraction from scientific literature： state of the art and challenges [J]. Brief Bioinform， 2014， 15(5)： 856-77.

[18] KARP P D. Can we replace curation with information extraction software? [J]. Database-Oxford， 2016.

[19] SZOSTAK J， ANSARI S， MADAN S， et al. Construction of biological networks from unstructured information based on a semi-automated curation workflow [J]. Database-Oxford， 2015.

[20] ANANIADOU S， THOMPSON P， NAWAZ R， et al. Event-based text mining for biology and functional genomics [J]. Brief Funct Genomics， 2015， 14(3)： 213-30.

[21] LI J， SUN Y P， JOHNSON R J， et al. BioCreative V CDR task corpus： a resource for chemical disease relation extraction [J]. Database-Oxford， 2016.

[22] AHMED Z， ZEESHAN S， DANDEKAR T. Mining biomedical images towards valuable information retrieval in biomedical and life sciences [J]. Database-Oxford， 2016.

[23] KIM D， YU H. Figure Text Extraction in Biomedical Literature [J]. Plos One， 2011， 6(1).

[24] KOZHENKOV S， BAITALUK M. Mining and integration of pathway diagrams from imaging data [J]. Bioinformatics， 2012， 28(5)： 739-42.

[25] LENC K， VEDALDI A J C S. R-CNN minus R [J]. 2015.

[26] PURKAIT P， ZHAO C， ZACH C. SPP-Net： Deep Absolute Pose Regression with Synthetic Views; proceedings of the British Machine Vision Conference(BMVC 2018)， F， 2017 [C].

[27] 曹诗雨， 刘跃虎， 中国图象图形学报 李 J. 基于Fast R-CNN的车辆目标检测 [J]. 2017， 22(5)： 7.

[28] REN S， HE K， GIRSHICK R， et al. Faster R-CNN： Towards Real-Time Object Detection with Region Proposal Networks [J]. 2017， 39(6)： 1137-49.

[29] ZHANG H， KYAW Z， YU J， et al. PPR-FCN： Weakly Supervised Visual Relation Detection via Parallel Pairwise R-FCN; proceedings of the Ieee I Conf Comp Vis， F， 2017 [C].

[30] SERMANET P， EIGEN D， ZHANG X， et al. OverFeat： Integrated Recognition [J]. 2013.

[31] REDMON J， DIVVALA S， GIRSHICK R， et al. You Only Look Once： Unified， Real-Time Object Detection [J]. 2016.

[32] REDMON J， FARHADI A J I. YOLO9000： Better， Faster， Stronger [J]. 2017： 6517-25.

[33] REDMON J， FARHADI A J A E-P. YOLOv3： An Incremental Improvement [J]. 2018.

[34] LIU W， ANGUELOV D， ERHAN D， et al. SSD： Single Shot MultiBox Detector [J]. 2015.

[35] LAW H， DENG J J I J O C V. CornerNet： Detecting Objects as Paired Keypoints [J]. 2020， 128(3)： 642-56.

[36] LIN T Y， GOYAL P， GIRSHICK R， et al. Focal Loss for Dense Object Detection [J]. 2017， PP(99)： 2999-3007.

[37] LECUN Y， BENGIO Y， HINTON G. Deep learning [J]. Nature， 2015， 521(7553)： 436-44.

[38] ROSENBLATT F. The perceptron： a probabilistic model for information storage and organization in the brain [J]. Psychol Rev， 1958， 65(6)： 386-408.

[39] RAWAT W， WANG Z. Deep Convolutional Neural Networks for Image Classification： A Comprehensive Review [J]. Neural Comput， 2017， 29(9)： 2352-449.

[40] LIN T Y， GOYAL P， GIRSHICK R， et al. Focal Loss for Dense Object Detection [J]. Ieee T Pattern Anal， 2020， 42(2)： 318-27.

[41] HE K M， ZHANG X Y， REN S Q， et al. Deep Residual Learning for Image Recognition [J]. 2016 Ieee Conference on Computer Vision and Pattern Recognition (Cvpr)， 2016： 770-8.

[42] PAVLIDIS T， MORI S. Optical Character-Recognition [J]. P Ieee， 1992， 80(7)： 1027-8.

[43] WANG B C， ZHANG X F， CAI Y H， et al. Optical Character Detection and Recognition for Image-Based in Natural Scene [J]. Lect Notes Artif Int， 2018， 10956： 360-9.

[44] TORRALBA A， RUSSELL B C， YUEN J. LabelMe： Online Image Annotation and Applications [J]. P Ieee， 2010， 98(8)： 1467-84.

[45] HE K， GKIOXARI G， DOLLAR P， et al. Mask R-CNN [J]. IEEE Trans Pattern Anal Mach Intell， 2020， 42(2)： 386-97.

[46] JIANG L， CHEN J， TODO H， et al. Application of a Fast RCNN Based on Upper and Lower Layers in Face Recognition [J]. Comput Intell

Neurosci， 2021， 2021： 9945934.

[47] VELUMANI K， LOPEZ-LOZANO R， MADEC S， et al. Estimates of Maize Plant Density from UAV RGB Images Using Faster-RCNN Detection Model： Impact of the Spatial Resolution [J]. Plant Phenomics， 2021， 2021： 9824843.

[48] JIANG X， ZENG Y， XIAO S， et al. Automatic Detection of Coronary Metallic Stent Struts Based on YOLOv3 and R-FCN [J]. Comput Math Methods Med， 2020， 2020： 1793517.

[49] FENG T， LIU J， FANG X， et al. A Double-Branch Surface Detection System for Armatures in Vibration Motors with Miniature Volume Based on ResNet-101 and FPN [J]. Sensors (Basel)， 2020， 20(8).

[50] HOANG T M， NGUYEN P H， TRUONG N Q， et al. Deep RetinaNet-Based Detection and Classification of Road Markings by Visible Light Camera Sensors [J]. Sensors (Basel)， 2019， 19(2).

[51] YU X， KANG C， GUTTERY D S， et al. ResNet-SCDA-50 for Breast Abnormality Classification [J]. IEEE/ACM Trans Comput Biol Bioinform， 2021， 18(1)： 94-102.

[52] CHEN M， ZHAO C， TIAN X， et al. Placental Super Micro-vessels Segmentation Based on ResNeXt with Convolutional Block Attention and U-Net [J]. Annu Int Conf IEEE Eng Med Biol Soc， 2021， 2021： 4015-8.

[53] SHUGUROV I， ZAKHAROV S， ILIC S. DPODv2： Dense

Correspondence-Based 6 DoF Pose Estimation [J]. IEEE Trans Pattern Anal Mach Intell， 2021， PP.

[54] YANG M， JIAO L， LIU F， et al. DPFL-Nets： Deep Pyramid Feature Learning Networks for Multiscale Change Detection [J]. IEEE Trans Neural Netw Learn Syst， 2021， PP.

[55] ROTHE R ， GUILLAUMIN M ， GOOL L V. Non-maximum Suppression for Object Detection by Passing Messages Between Windows [J]. Computer Vision - Accv 2014， Pt I， 2015， 9003： 290-306.