MirWorkbench: an Integrated Environment for MiRNA Data Analysis

SUPANAT KAMALES 55090050

Software Engineer Bachelor Degree
International College
King Mongkut's Institute of Technology Ladkrabang

B.Eng THESIS
title
MirWorkbench: an Integrated Environment for
MiRNA Data Analysis
by
Mr.Supanat Kamales

Signature.....

Asst.Prof.Dr. Kulwadee Somboonviwat
International College
King Mongkut's Institute of Technology Ladkrabang
Advisor

Contents

List of Figures

List of Tables

ABSTRACT

MicroRNAs (miRNAs) are a class of small (about 22 nucleotides long), non-protein-coding RNAs that play an important role in post-transcriptional regulation of many different genes across numerous species. As miRNAs are likely to be a key element of preventive as well as therapeutic technologies in animal diseases, much effort has been made to identify novel miRNAs and to better understand their functions by analyzing miRNAs sequence data.

Analysing microRNA data poses several challenges to scientisets as follows. First a scientist has to execute many command-line tools to complete a workflow. This forces him/her to type in a lot of commands, which is very cumbersome and error-prone. Second, the command-line tools typically generate a lot of output files, making it very difficult to manage and organise these files manually. Third, a lot of algorithms are available for each execution step within the miRNA data analysis workflow. Because there is no universally superior algorithm for each execution step, the scientist needs to try appling several algorithms to the same data within a single execution step and then manually select the most relevant results.

To address these challenges in miRNA data analysis, this thesis aims to develop an integrated software environment, called MirWorkbench, that can manage the experiment data and support miRNA data analysis of the Centre of Excellence for Molecular Biology and Genomics of Shrimp (CEMS laboratory) at Chulalongkorn University. The user requirements were collected by conducting several interview sessions and user interface prototyping with the scientists at the CEMS laboratory. The software design and construction were conducted based on the Object-Oriented Analysis and Design (OOAD) as well as the Object-Oriented Programming (OOP) concepts. The resulting software, i.e. the MirWorkbench, addresses the requirements of the end-users (i.e. the scientists at the CEMS laboratory) with its three key features: (1) configuration and management of experiments using projects, (2) customizable workflow, and (3) visualization of experimental results.

ACKNOWLEDGEMENT

This research was funded by the Ratchadapisek Sompoch Endowment Fund (2014), Chulalongkorn University (CU-57-019-FW).

I would like to express my sincere thanks to Asst.Prof.Dr. Kunlaya Somboonwiwat (Department of Biochemistry, Faculty of Science, Chulalongkorn University) who has provided a lot of helpful advices during the user interviews, data collection, and user interface prototyping at the CEMS laboratory as well as Asst.Prof.Dr. Kulwadee Somboonviwat for providing support in every way.

Chapter 1

INTRODUCTION

1.1 Background and Significance

MicroRNAs or in short miRNAs are argonaut bound small RNAs that play an important role in development of many fatal disease such as cancer and heart disease. This 21 - 23 length of nucleotides microRNAs can be found in almost every organism such as plants, animals, viruses and human. MicroRNA role is to control the regulation of gene expression. By stopping messengerRNA to do its job, a protein will not be produced. Consequently, this action lead to development of many disease.

MicroRNAs gain a lot of attention from scientist since they have been first discovered. In the recent years, the advancement of sequencing technology allow scientist to generated a large volume of sequencing data. Furthermore a lot of new algorithms for identifying miRNAs and predicted novel miRNAs have been proposed in order to help scientist identified new miRNAs for example miRDeep2, miRExpress and miRanalyzer.

As mentioned in a previous paragraph, a lot of algorithms for identifying miRNAs and predicted novel miRNAs have been proposed. This mean that there are a lot of tools availiable for scientist to used. However there are not many tools that provide an environment for miRNA data analysis and to choosing the right tools for the job is even harder since it comes in many forms such as software packages or web services. Each of them has their own strong and weak point. Therefore a tools that provide a software environment to support miRNA data analysis and conduct a performance comparison of those algorithms would help scientists a lot.

1.2 Scope of Thesis

This thesis address two problems in miRNA data analysis which are,

- In miRNA data analysis, there is a workflow that researchers have to follow. This workflow can be slightly different depent on the goal. But when looking at the bigger picture. They still follow the general workflow. The problem that this project are going to address is to design the workflow that comfortable to the user's lab the most.

- The automatic software for miRNA data analysis and effeciently manage experiment data is needed. Because there are several steps in miRNA data analysis and each step has its own tools. Since there is not many tools that provide a graphic user interface. Scientist have to remember a lot of commands. When they type a command. If there is a typo in it. They have to type it all again. This is inconvenient. They should use their time to focus on something else. We can address this problem by creating a software with graphic user interface that can manage the experiment data and support miRNA data analysis.

1.3 Outline of Thesis

Before we progress further, let's look at the overview of the whole thesis.

- Chapter 2 will focus mainly on background knowledge of microRNA like What is microRNA?, the biogenesis of microRNA, what is the function of microRNA and the important file format in RNA and microRNA analysis.
- Chapter 3 will discuss about the user requirement. What is the goal that user wanted to achieve from this project and the User Interface of the program.
- Chapter 4 the System analysis and Design. This chapter will show the design of the software. The design of the software contain system requirement, class diagram and sequence diagram.
- Chapter 5 will report the results of this thesis by walking through all key features of the MirWorkbench program.

Chapter 2

LITERATURE REVIEW

2.1 What is MicroRNA?

MicroRNA is a non-coding RNA that play important part in development of disease such as cancer, heart diseases. Normally our body use RNA to produce protein. But some microRNAs stop the production of protein instead of produce one by suppressed messenger RNA. Normally the length of microRNA is 21 - 23 nucleotides.

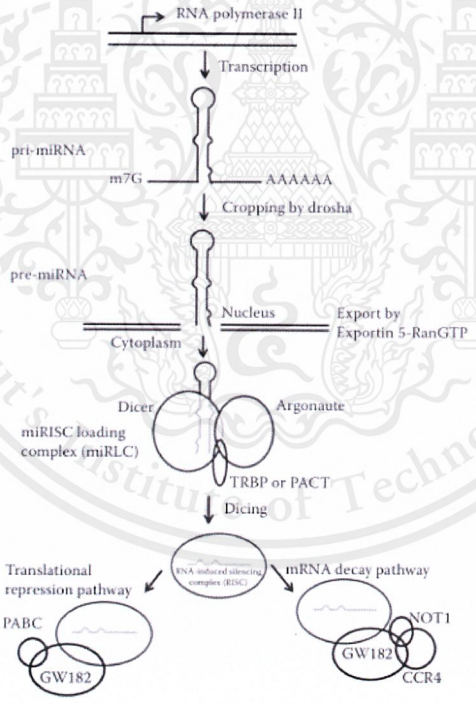

Figure 2.1: shows biogenesis and processing pathway of microRNAs. In the remaining of this section, we will explain the details of microRNA biogenesis and their functions.

2.1.1 Biogenesis

The creation of miRNA started within the nucleus (Figure 2.1). miRNA gene mostly transcribed by an enzymes called Polymerase II. After Polymerase II produced single stranded RNA. It will fold into hairpin double stranded structure with a length of 100 to 120. This double stranded called pri-miRNA. One of it may contain upto six miRNA precursors. An enzyme name drosha combine with another enzyme called DGCR8 to trimmed a pri-miRNA. After drosha and DGCR8 trim a pri-miRNA. It is now called pre-miRNA instead of pri-miRNA. The length of it will decrease to about 70 nt long. Pre-miRNA will then delivered to a cytoplasm by a protein called Exportin 5. In a cytoplasm a Dicer will bind with a Pre-miRNA and cleave out a hairpin of a pre-miRNA. Now a pre-miRNA will be break down in to miRNA: miRNA* duplex. Now the miRNA is almost ready to do its function.

2.1.2 Functions

MicroRNAs main function are to regulated the degradation of gene in living organism. This is called gene silencing. Gene silencing process occur in a cytoplasm. After a dicer cleave out a hairpin of pre-miRNA leave only an imperfect double stranded miRNA: miRNA* duplex. The miRNA use only one stranded to do the job. So one must be detach from another. This is where the argonaut protein comes. The miRNA: miRNA* duplex then loaded into RiSC or RNA-induced silencing complex and bound with the argonaut protein. An Argonaut protein consist of two domain. The fist domain is PAZ domain. This domain attach with one strand of miRNA: miRNA* duplex. The second domain of argonaut is PIWI domain breakdown miRNA: miRNA* duplex into single strand. The strand that has been selected by an argonaut protein will be called guide strand and another one that is not selected will be called passenger strand. An argonaut protein will then bring an active miRNA to the target messenger RNA. It will form a double strand with a part with complimentary bas pair cause the messengerRNA to degradaed.

2.2 Biological Data Formats

2.2.1 FASTQ

FASTQ file is a text-based format used to store nucleotide sequences. The structure of the file is split into four main parts. The first part is the sequence identifier part. This part begin with ">" sign follow by sequence identifier. The second part is the sequence. The third part start with "+" sign and optionally follow by description. The last part is the encrypted quality score for each bases in the sequence. The quality score also known as "Phred or Q" is the encoded integer value that represent the probability that base is incorrent. This integer value can be translated in to probability value with following formular :

$$P = -10^{-Q/10}$$

to translated P value back to Q value use this formular :

$$Q = -10 * log_{10}(P)$$

Range of Q value can be different depending on the version of tools, for Sanger, Illumina v.1.3 to v.1.7 sequencing tools. Range of Q value is 1 to 40. Range of Q value of Illumina v.1.8 and later is 1 to 41. The character of this two version is also different.

2.2.2 FASTA

FASTA (pronounced fast-ah or fast-a) file format is also a text-based file format used to store any kind of sequences such as reference genome files, protein sequences, coding DNA sequences, transcript sequences. It was created by William R. Pearson and David J. Lipman Format [4]. Format of FASTA file is quite similar to FASTQ file. However FASTA file does not hold the quality score of the sequence. FASTA file only has two important parts. The first part begin with ">" follow by sequence identifier. There should not be any space in between the sign and sequence identifier. It is also possible to add description to the sequence by adding space or semicolon(;) in front of the description. The next line after the name of the sequence and description will be the sequence data. Unfortunately the format of FASTA file is not stable. So the problem can occur if you creating your own FASTA file reader. Instead of implementing your own FASTA file reader. It is better to use FASTA/FASTQ parsing library. [4]

2.3 MicroRNA Data Analysis Workflow

The micraRNA data analysis workflow represent the step to analyze microRNA [6]. miRNA data analysis workflow diagram is shown in Figure 3.1. The workflow start with pre-processing step. In this step, the input data in fastq, fasta, sam format will be clean from adapter, contaminate artifact, noise or unwanted artifact in input reads and converted into fasta file format. There are a lot of tools such as fastqtofasta for converting fastq file to fasta file, cutadapt for remove adapter from the reads and fastqc to remove low quality nucleotide The next step is microRNAs Discovery. The known and novel will be identify in this step using machine-learning based on sequence conservation and/or structural similarity. After distinguish between known and novel miRNAs, the data will be proceed to the target prediction step. The target prediction step will try to idetify miRNAs. There are several factor that can used to identify the miRNA.

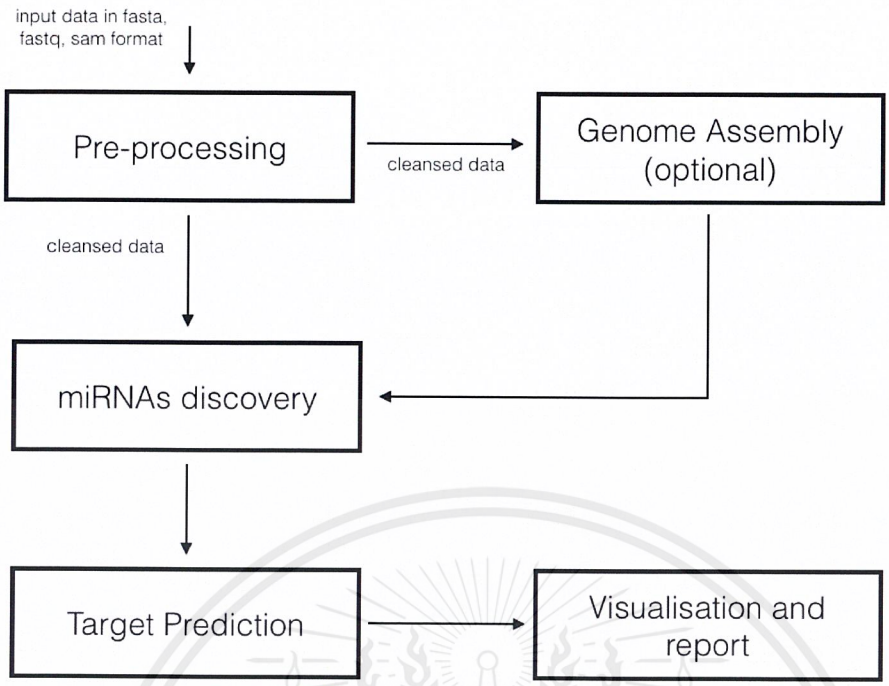

Figure 2.2: miRNA Data analysis workflow

Chapter 3

Requirements Analysis

3.1 MiRNA Data Analysis Workflow of the CEMS LAB

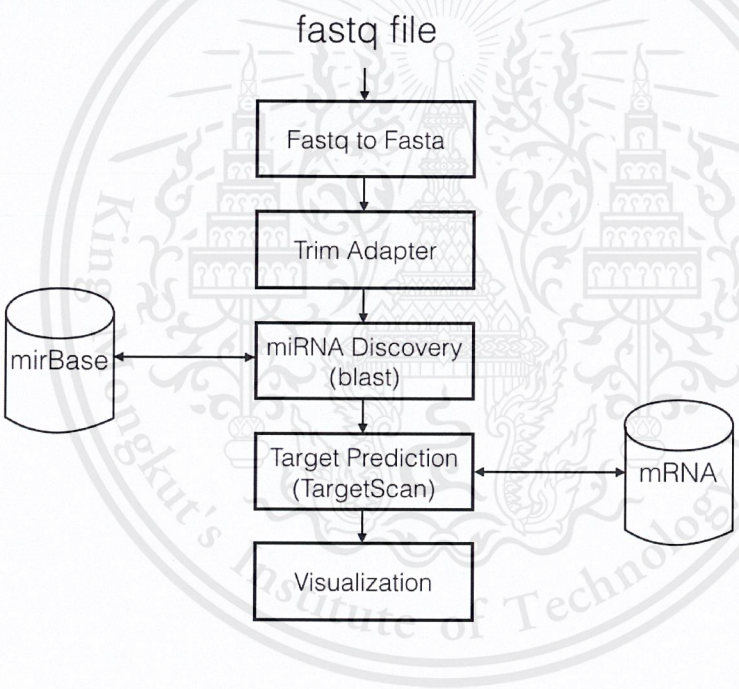

Figure 3.1: miRNA Data analysis workflow of Center of Excellence for Molecular Biology and Genomic of Shrimp

The analysis process of Center of Excellence for Molecular Biology and Genomic of Shrimp of Chulalongkorn University start when the user obtain the data from the Next Generation Sequencer or NGS in short. This is where the workflow start figure 3.1. The workflow begin

with the Pre-Processing cleaning and preparing the input reads for the up coming analysis. NGS usually generated the result in fastq file format. Fastq file format contain the sequence name, sequence and the quality of each nucleotide. Now since the input data is in fastq file format. It must be good to begin by removing the low quality nucleotide suing fastqc. After low quality nucleotides have been removed from sequences. We can either convert from fastq to fasta first or remove the adapter from the sequence. Let assume that we convert the file first. We can convert fastq file by using fastqtofasta tool. After we obtain the fasta file we cannow begin trimming the adapter from the sequences using cutadapt. Now the input data must be cleaned already. We can move on to the next step which is MicroRNA Discovery. In miRNA discovery step, CEMS use blast tools identify the known and novel miRNA. Known miRNA is a miRNA that is already in the database. But novel miRNA is possibly a new kind of miRNA that has never been found before. After we identify the known and novel miRNA. Research will proceed to the next step. The novel miRNA will be mapped with the genome of the target species that researcher studying to find the binding site in the target species genome. Becuase microRNA will not work if there is no messengerRNA to bind with. So if researcher can find the potential binding site in the genome researcher can predict the function of certain miRNA. This step is called TargetPrediction. The tool that CEMS lab use is for this step is TargetScanS. Since files that generated by tools are unorganize. Researcher has to rearrange them manually.

3.2 Use Cases

Use case diagram show who can use this system and what they can do with the system. The use case diagram consist of actor and action. Actor represent the user. Actor can be everythings for example sciencetist, administrator, customer etc. In this case the actor is a user that is going to use this program. An action is the thing that user can do with the system. Use case diagram is shown in Figure 3.2. According to the use case diagram, an actor name user can create project, open project, close project. User can use the program to perform pre-processing, miRNA Discovery and Target Prediction and see the result of it, but before the user is going to do those task the user have to change a parameter that will be passed to an algorithm or choose the default profile. If the user choose to adjust the parameters by hand. User can choose whether they wanted to save it into the profile for later use or not. User can perform systematic comparison between algorithms. User can add file into the project such as input read, genome etc.

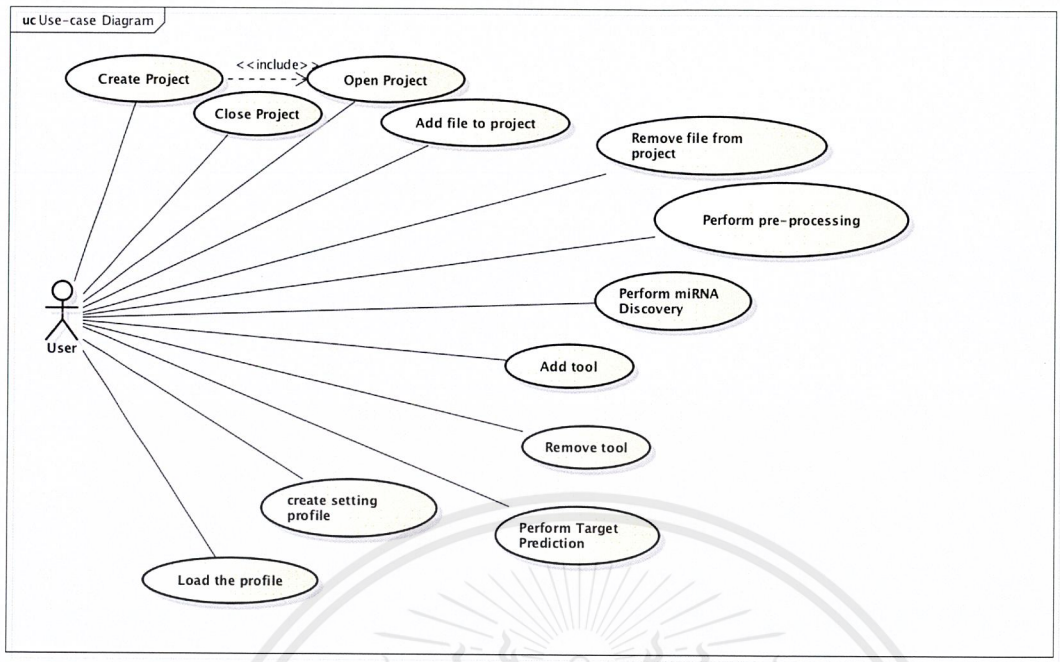

Figure 3.2: Use case Diagram

3.2.1 Use Case Description

Use case description show the use case in more detail. It show the primary actor of certain use case. The pre and post condition. What should achieve before actor can do this use case and what will happen after. The flow of event show what the program will do if actor provide some input to the program. An alternative flow is the flow that different to the flow of event.

Use case	Create Project		
Primary Actor	User		
Pre-Condition			
Post-Condition	1. Program create a project 2. Program open a project		
Flow of Event		Action Input	System response
	1	Click Create project button	
	2		Show create project dialog
	3	Create directory	
	4	Enter Project Name	
	5	OK button clicked	
	6		Close project dialog
	7		Create project in selected directory
Alternative Flow	A : If the project name is not given 1. Prompt the user to enter a project name 2. Back to create project dialog B : If cancel button in the dialog has been clicked 1. Close create project dialog 2. Do nothing		

Figure 3.3: Create project description

Use case	Open Project		
Primary Actor	User		
Pre-Condition			
Post-Condition	1. Project has been open		
Flow of Event		Action Input	System response
	1	Click open project button	
	2		Show open project dialog
	3	Select project file with pmi extension	
	4	Press OK	
	5		Program close open project dialog
	6		Program open selected project
Alternative Flow	A : Cancel button clicked 1. Close open project dialog		

Figure 3.4: Open project description

Use case	Close Project		
Primary Actor	User		
Pre-Condition	1. The program must have a project opened		
Post-Condition	1. Program close a project		
Flow of Event		Action Input	System response
	1	Click close project button	
	2		Close project

Figure 3.5: Close project description

Use case	Add file to project		
Primary Actor	User		
Pre-Condition	1. The program must have a project opened		
Post-Condition	1. Selected file added to the project		
Flow of Event		Action Input	System response
	1	Click add file button	
	2		Show add file dialog
	3	Select file and press ok	
	4		Close dialog and asked whether to move the file to the project or not
	5	Yes/No	
	6		Move the file in case of Yes.
	7		Add selected file to the project
	8		Save the project
Alternative Flow	A : Cancel button clicked 1. Close add file dialog		

Figure 3.6: Add file description

Use case	Remove file from project		
Primary Actor	User		
Pre-Condition	1. File selected		
Post-Condition	1. Selected file removed from the project		
Flow of Event		Action Input	System response
	1	Select file to remove	
	2	Click remove file button	
	3		File removed from the project

Figure 3.7: Remove selected file from project description

Use case	Perform Pre-Processing		
Primary Actor	User		
Pre-Condition	1. There are at least one tool in the pre-process tab 2. All required parameter is given		
Post-Condition			
Flow of Event		Action Input	System response
	1	Start button pressed	
	2		Put pre-processing tool queue into a main queue
	3		Add result file to project
Alternative Flow	A : If run all tool check box checked 1. put pre-processing tool queue into a main queue 2. put miRNA discovery tool queue to main queue after pre-processing 3. put target prediction tool queue into main queue after miRNA discovery 4. execute all tool in main tool queue sequentially		

Figure 3.8: Perform Pre-processing description

Use case	Perform miRNA discovery		
Primary Actor	User		
Pre-Condition	1. There are at least one tool in the miRNA Discovery tab 2. All required parameter is given		
Post-Condition			
Flow of Event		Action Input	System response
	1	Start button pressed	
	2		Put miRNA Discovery tool queue into a main queue
	3		Add result file to project
Alternative Flow	A : If run all tool check box checked 1. put pre-processing tool queue into a main queue 2. put miRNA discovery tool queue to main queue after pre-processing 3. put target prediction tool queue into main queue after miRNA discovery 4. execute all tool in main tool queue sequentially		

Figure 3.9: Perform miRNA Discovery description

Use case	Perform Target Prediction		
Primary Actor	User		
Pre-Condition	1. There are at least one tool in the Target Prediction tab 2. All required parameter is given		
Post-Condition			
Flow of Event		Action Input	System response
	1	Start button pressed	
	2		Put Target Prediction tool queue into a main queue
	3		Add result file to project
Alternative Flow	A : If run all tool check box checked 1. put pre-processing tool queue into a main queue 2. put miRNA discovery tool queue to main queue after pre-processing 3. put target prediction tool queue into main queue after miRNA discovery 4. execute all tool in main tool queue sequentially		

Figure 3.10: Perform target prediction description

Use case	Add Tool		
Primary Actor	User		
Pre-Condition			
Post-Condition	1. Tool block will be added to the current step tab		
Flow of Event		Action Input	System response
	1	Add tool button pressed	
	2		Add tool block to current step tab

Figure 3.11: Add tool description

Use case	Remove Tool		
Primary Actor	User		
Pre-Condition	1. There are at least one tool in the current step tab		
Post-Condition			
Flow of Event		Action Input	System response
	1	Remove tool button pressed	
	2		Tool block removed from the tab

Figure 3.12: Remove tool description

Use case	Create setting profile		
Primary Actor	User		
Pre-Condition	1. There are at least one parameter given in the tool block		
Post-Condition	1. Profile will be saved to the profile list		
Flow of Event		Action Input	System response
	1	Save profile button pressed	
	2		Show profile name dialog
	3	Profile name given	
	4		Open profile list file
	5		Write profile name and parameter to file
	6		Close file

Figure 3.13: Create setting profile description

Use case	Load profile		
Primary Actor	User		
Pre-Condition	1. There are at least one profile for the current tool in a profile list		
Post-Condition	1. Parameter that record in the file will be set to its place		
Flow of Event		Action Input	System response
	1	Select profile	
	2		Read parameter of given profile name
	3		Set the parameters

Figure 3.14: Load profile description

Chapter 4

System Analysis and Design

This chapter of a thesis will show the design of the software. It started with user requirement follow by MicroRNA Data Analysis Workflow, Use case follow by the interfaces prototype, system requirement and class diagram.

4.1 System Requirements

What is shown in user requirement is the things that user expected the program to be able to do. The user requirement is design based on the principle of FURPS+. FURPS+ can be break down in to six smaller categories, Functional, Usability, Reliability, Performance, Security and other (+). The requirement is shown in Table 4.1.

Requirement	FURPS+
A program allow user to create project.	Functional
A program allow user to close project.	Functional
A program allow user to open a created project.	Functional
A program allow user to add input read.	Functional
A program allow user to remove input read.	Functional
A program allow user to add input genome.	Functional
A program allow user to remove input genome.	Functional
A program must be able to perform all task in miRNA data analysis workflow	Functional
A program must be able to add result file into the project automatically	Functional
A program must be able to trim adapter in target reads	Functional
A program must be able convert from fastq file format to fasta file format	Functional
A program must be able to generated the report after perform algorithms comparison	Functional
Graphic User Interface must be easy to understand.	Usability
A program must be able to run on Linux base system	Usability
A program must be able to run for month continuously.	Reliability

Table 4.1: System Requirement

4.2 Class Diagram

Class diagram show the structure of the program. The box is represent the class. The box has three important sections. The first section on the top of the box is the class name. The second section lie in the middle of the box. This section contain all attribute of the class. The pattern used to represent the attribute is [access modifier] [attribute name]:[data type of an attribute]. The access modifier tell the compiler that this attribute can be access by other class or not. There are mainly three type of access modifier. First public or a (+) sign. The attribute with public access modifier can be access freely by other class. On the other hand if the attribute has private access modifier or a (−) sign. Other class cannot access this attribute. The last type of access modifier is protected(#). Protected attribute can be access by a child of this class but cannot access by other class. After the access modifier is an attribute name. The last one is data type of an attribute. The data type of an attribute and attribute name are seperated by colon sign(:). The last section of the box is the method section which locate on the button of the box. The representation pattern of method is quite similar to the attribute. But method has parameter name and data type of a parameter after the method name.

The relationship between the classes are represent by line with different arrow head. Each arrow head has different meaning. The line with no arrow is called Association. The association is represent as a line without arrow head. The line can be named. the next relationship is Aggregation. Aggregation is a has-a relationship. It happen when one class contained the other class. But container class doesn't responsible for the creation of the contained class. Aggregation is represent as an white diamond arrow. The black diamond arrow head represent the Composition relationship. The composition mean the class that has been pointed are the container of the pointer class. The container create the contained class. If the container die, contained class also die. The line with white arrow head represent the Generalization relationshipe. Parent class use this arrow to point to a child class.

From figure 4.1 the structure of this program are divided into 2 packages. The first package is the UI package. The second package is the Utility. The UI package is responsible for everything about user interface. UI package also contain the class that responsible for creating an algorithm tool block, storing the parameter from user and generating the command from the input parameters. The Utility provide support for all other package but mainly on UI package. The Workflow package responsible for generating command and control the workflow.

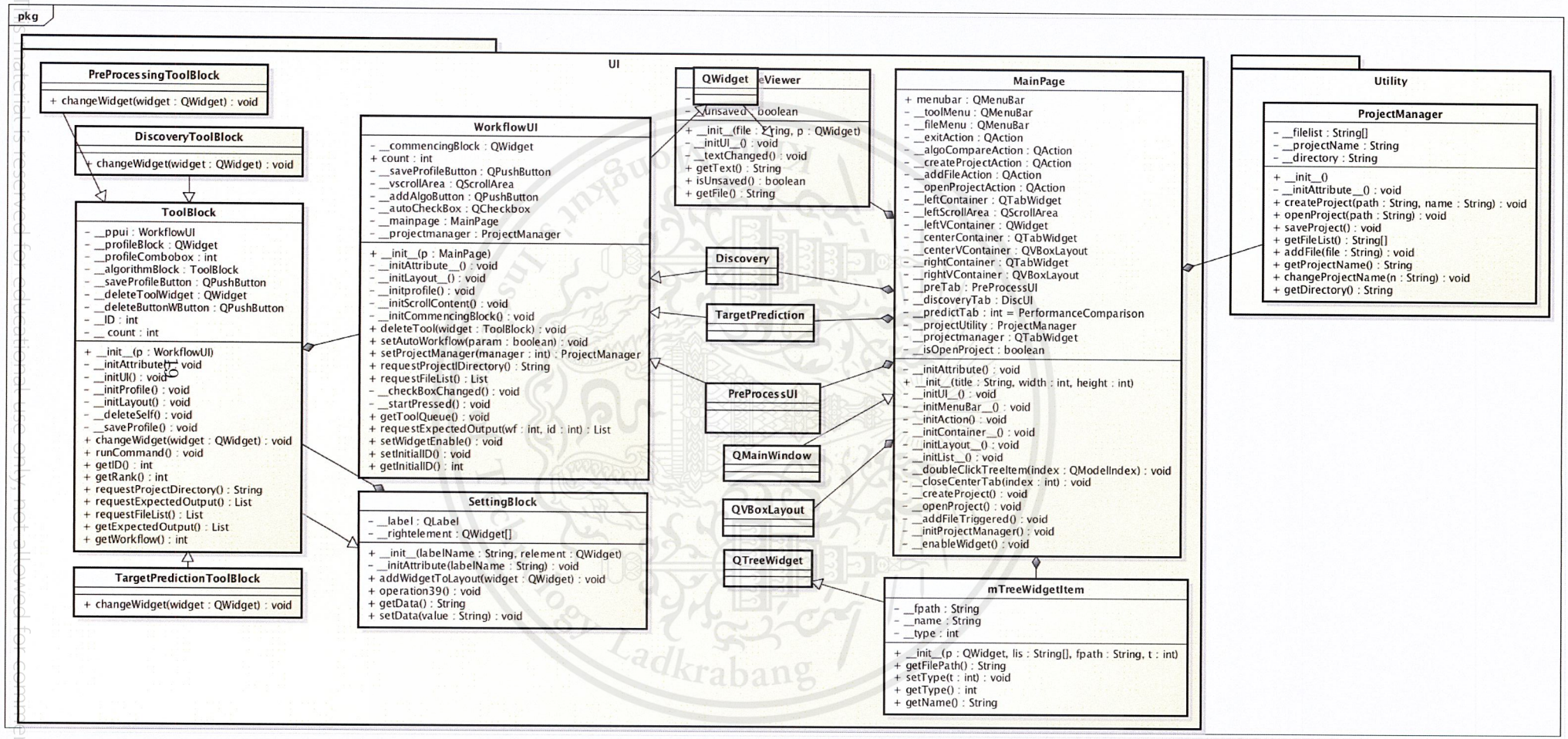

Figure 4.1: Class Diagram

4.3 Sequence Diagram

Sequence diagram is a diagram that show the commmunication between object in order. An object is represent by rectangle box. A dash line below the box is the lifeline of an object. Horizontal line with arrow on one end represent the message sent from one object to another object. A dash line with arrow represent the return point. Stereotype is used to represent the action of the message. Create stereotype is used when one object create another. Stereotype is represent by two less than signs follow by stereotype, enclose by two greather than signs. From this point we will split the sequence diagram into two group. First group is Project management and Tool configuration and execution.

4.3.1 Project management

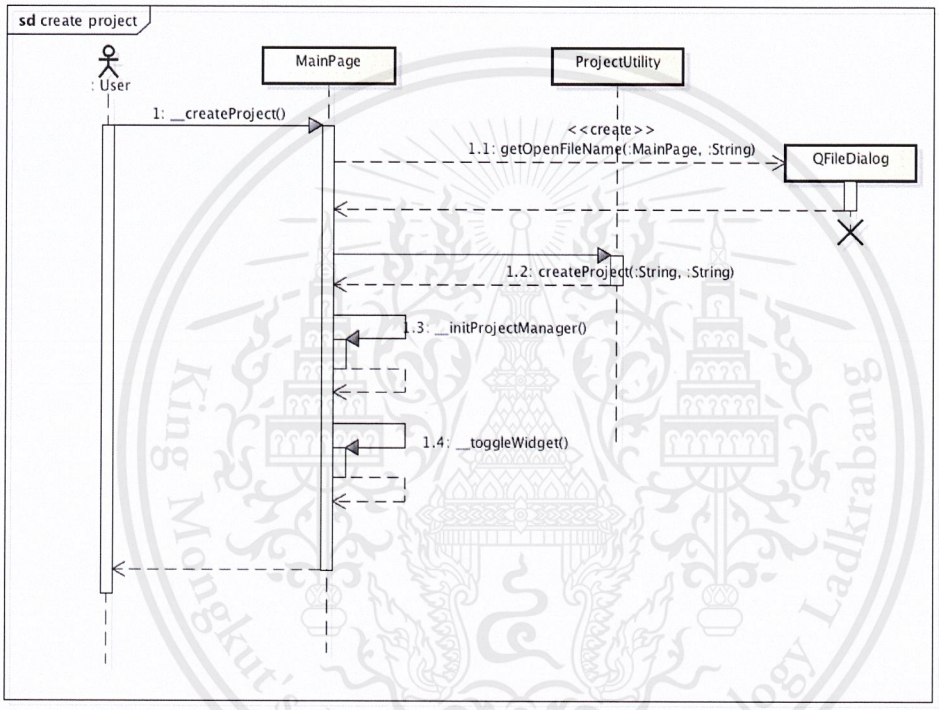

Figure 4.2: create project sequence diagram

The first sequence diagram that we are going to explain is create project sequence diagram. Figure(4.2) The sequence begin when the user press create project button. Create project button then invoke the createProject method of the MainPage object. MainPage object then create QFileDialog object to let user select directory and enter the project name. After the user select directory and enter the project name. QFileDialog will return the result to the MainPage object and destroyed. After validate the return data received from QFileDialog object. MainPage sent the data to a ProjectUtility with createProject method. ProjectUtility will create a everythings neccessary for the project to work then return to MainPage object. MainPage object then call initProjectManager and toggleWidget method and return to user.

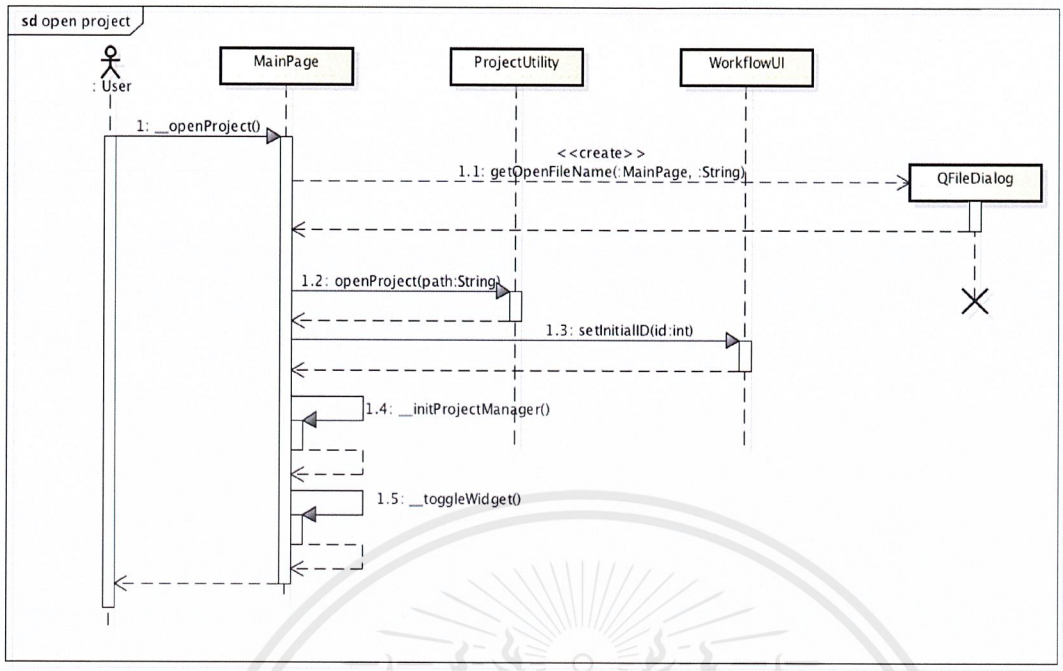

Figure 4.3: open project sequence diagram

The second sequence diagram is open project. The sequence start when user press open project button. Figure(4.3) The button then call openProject method. MainPage then create QFileDialog, so that user can select project to open. To prevent mistake, the dialog only allow user to pick a file with pmi extension. After user pick a file. openProject will validate the input. MainPage will use ProjectUtility to read a data from the project file. MainPage then call method setInitialID of WorkflowUI and pass ID that read from the project file as a parameter. After finish invoking initProjectManager and toggleWidget method. MainPage will return to the user.

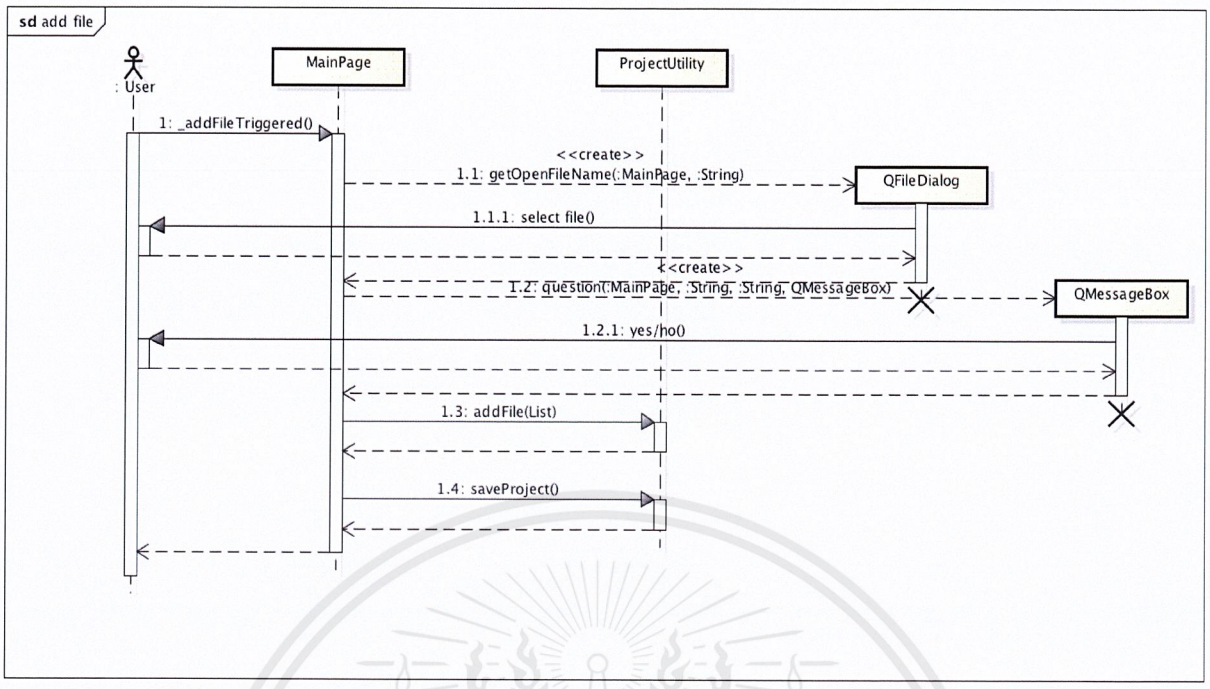

Figure 4.4: add file to project sequence diagram

For add file sequence diagram. Figure(4.4) User press the add file button. Add file button will called addFileTriggered method of the MainPage. Then MainPage create QFileDialog. QFileDialog will be used to show the file and receive the file that user wanted to add. After user select a file. MainPage will create a QMessageBox to ask the user that he/she want to copy the file to the project directory or not. After user answer, MainPage will add file to the project using addFile method. addFile method required one parameters which is the file path. After finish adding project. MainPage will save the project and return to the user.

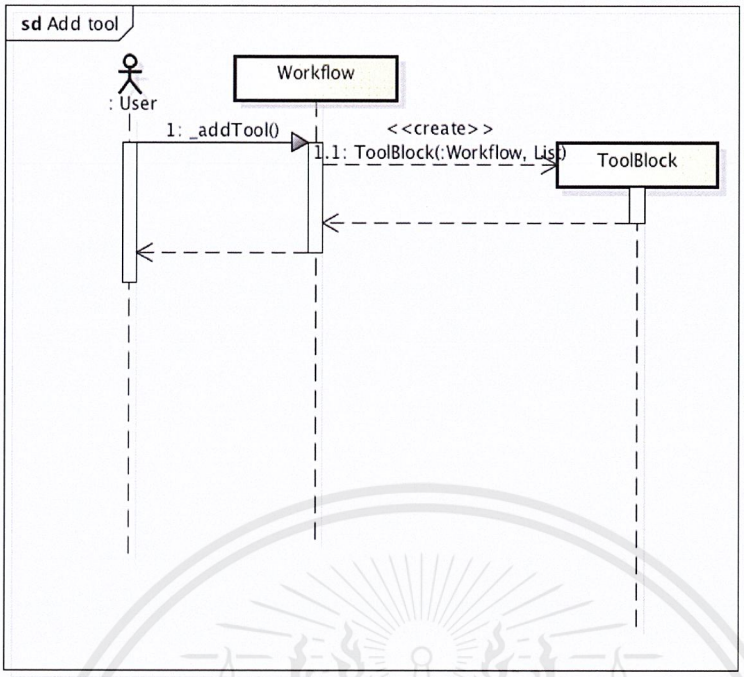

Figure 4.5: add one more tool to the current workflow sequence diagram

The next sequence diagram is add tool. Figure(4.5) When user press add tool button. The button invoke addTool method. addTool method will create ToolBlock and add it to the tab.

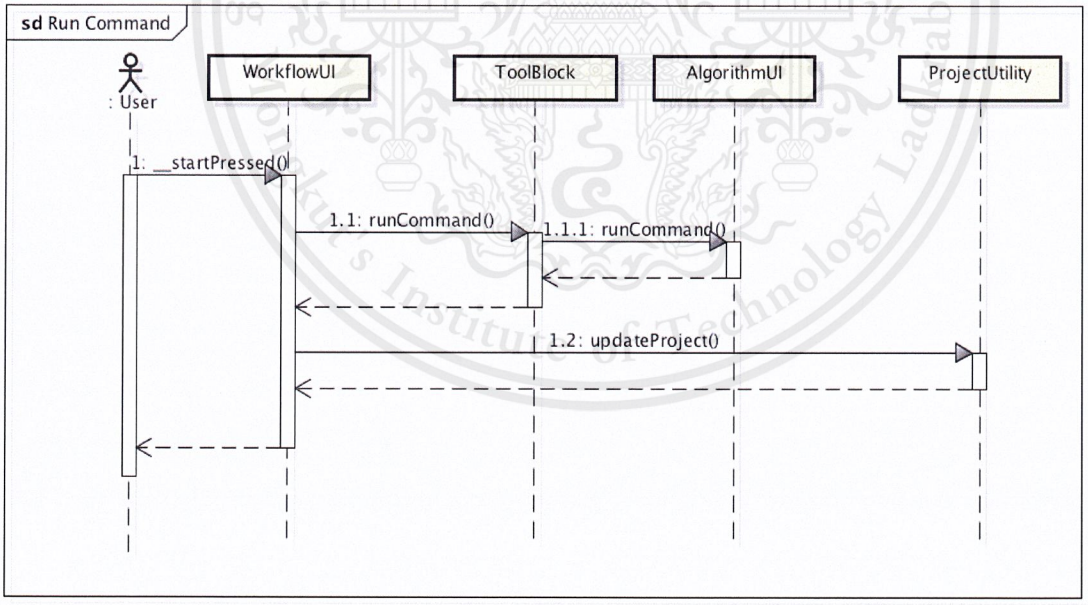

Figure 4.6: Run all tools

4.3.2 Tool configuration and execution

To run the command figure(4.6) user have to add tool to the workflow and enter required paramter require by the tools. If user want to execute only one workflow. User must leave the radio button and the bottom of the window uncheck. On the other hand if the user want to execute whole workflow. User have to check the radiobutton. Program will start execute when user press startPressed button of the WorkflowUI. WorkflowUI will call runCommand method of ToolBlock object. ToolBlock object then call runCommand method of the AlgorithmUI. After finish running the algorithm. WorkflowUI will update the project by calling updateProject method from ProjectUtility object.

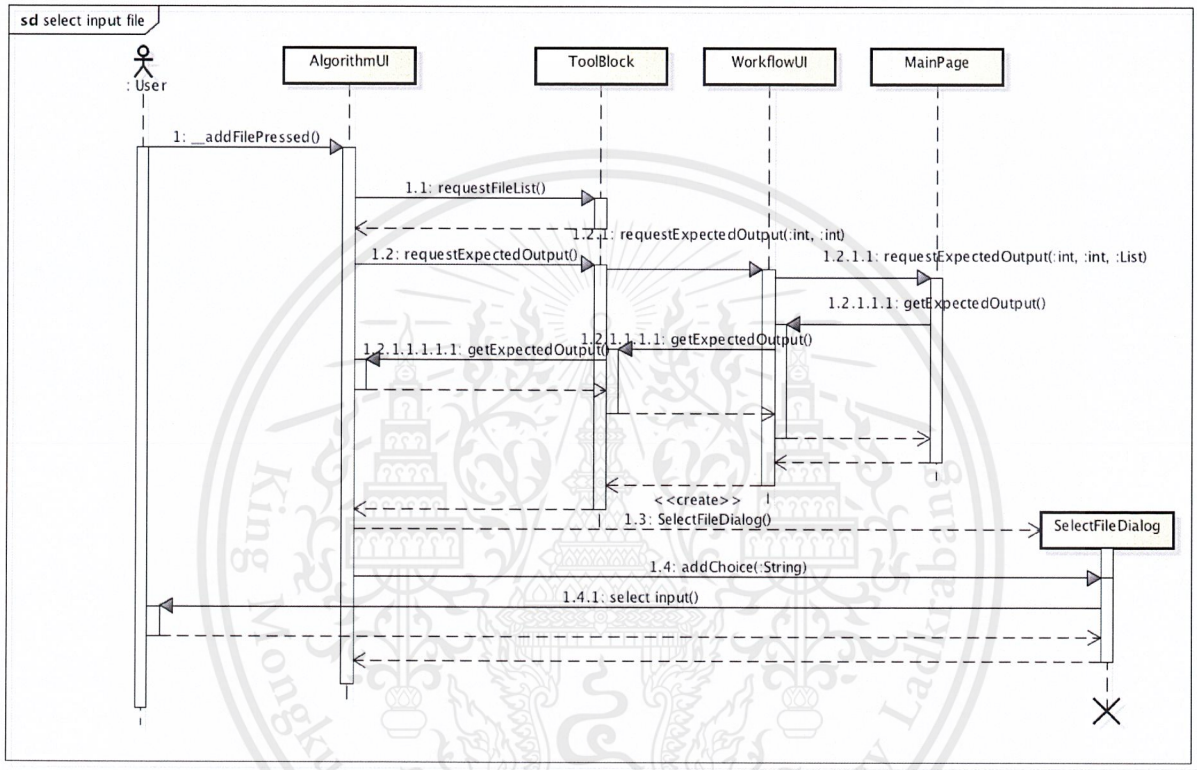

Figure 4.7: Select input file dialog

When user want to add the input file to the tool. User have to press select input button in the tool block. figure(4.7) The button will invoke addFilePressed method of AlgorithmUI. AlgorithmUI will request file list from the ToolBlock object using requestFileList.ToolBlock then request File list from the WorkflowUI then WorkflowUI request file list from MainPage. MainPage will collect all file from all tools of each workflow. MainPage then add all file in the project to the list. Then MainPage will request all expected output from every tools of each workflow that run before the algorithm. Then MainPage return the list to the algorithm that request. After receive file list, AlgorithmUI add all file list to the SelectFileDialog using addChoice. SelectFileDialog will pop up and wait for the user to select files. After user select a files. SelectFileDialog return the selected files to the AlgorithmUI and destroyed

4.4 Workflow Engine

To provide a software that support miRNA data analysis and customizable workflow. The program is designed to cover every step in the workflow(Preprocessing, miRNA Discovery and Target Prediction). To make it more flexible to the point that the workflow is customizable. The program allow researcher to add tool freely to each step in the workflow. But the problem now occur. If the program let researcher add tool manually. Data structure for storing the tool is required.

4.4.1 Queue

It's exactly like it sound. The general idea of queue is first come first serve. The first object which get in the queue has right to do what it came for first. In this case, the tools is store within the queue. The first tool in the queue will execute first. The second tool will execute once the first finished. Each step has its own queue. Researcher can also add the same tool into the queue in case if reseachers wanted to use the same data but different parameters.

4.4.2 Execution

There are two procedures to execute the workflow. The first one is to run only the current step. Another one is to run every step in the workflow. This two procedures are slighly different. For the first procedure, when researcher start execute. The program will retrieve the queue of the current step and run it. For the second procedure, first the program will retrieve tool queue from preprocessing step and add it to the main queue. Then the program will request tool queue from miRNA discovery step and add it to the main queue after the preprocessing queue. Then the program will add the target prediction tool queue to the main queue Figure(4.8). Now after the main queue has every queue from every step in the workflow. The program will start execute the tool in queue order. To prevent the program from irresponding. Program created a new thread to run the tools. The first tool in main queue is send to the new process to run. But how do the main queue know when to send another tool to a new process. The lock and release machanism is implemented to lock the main queue from sending the tool to new process unless the new process has finished running the current tool. The new process lock the main process when the main process give the new process a tool to run. The lock will be release when a new process finished running. If the execution of tool end up error. The new process will notify a main process then release a lock and kill itself. The state diagram of new process is shown is Figure(4.9).

Figure 4.8: Tool queue contain all three queue from different step in the workflow. Red for Pre-processing, green for miRNA Discovery and black for Target Prediction. The head of the queue is on the left side.

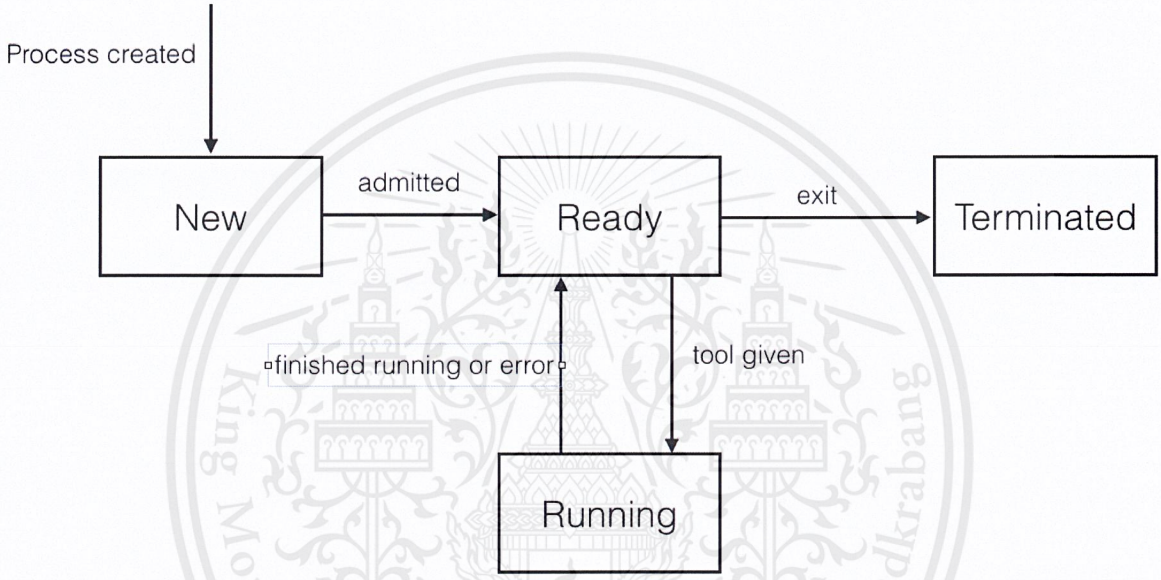

Figure 4.9: State diagram of the new process

Chapter 5

Result

5.1 Experiments Project

File management is a problem that every researchers have. This is why MirWorkbench was implemented file management feature within the program. To use MirWorkbench researcher has to create or open a project first. Researchers can do so by following these step. First researcher has to press File menu on top of the program and press create project (Figure 5.1).

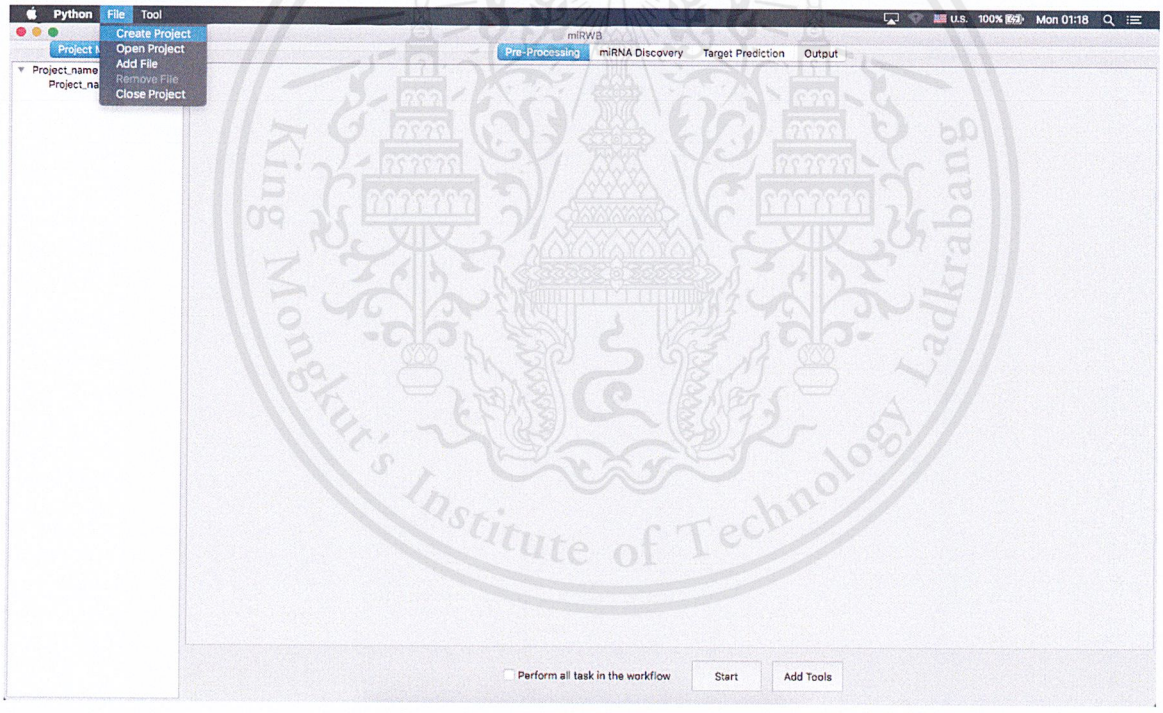

Figure 5.1: File menu then create project

After researcher press create project a dialog will appear. Researcher has to create a project directory then select that directory, enter the project name and press ok (Figure 5.2).

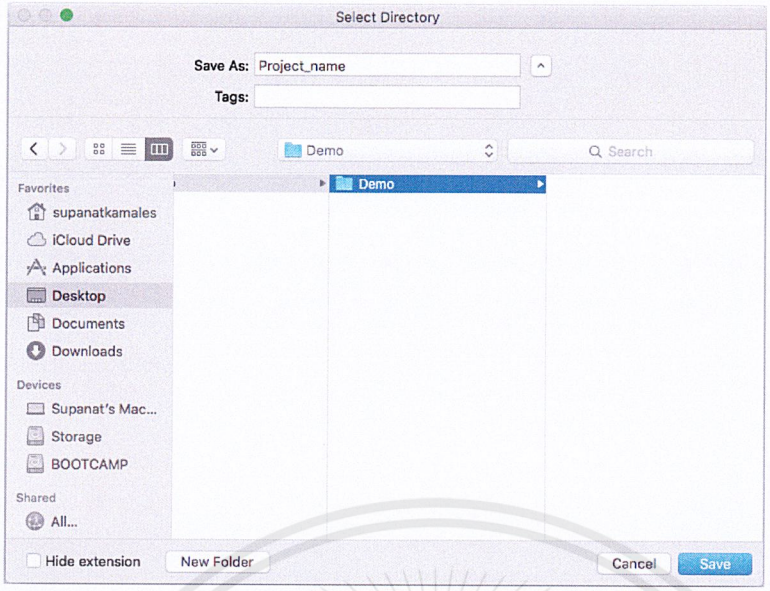

Figure 5.2: create a project directory then enter a project name

If researcher has created a project already and wanted to continue working. Researcher can do so by press File menu and press open project instead (Figure 5.3).

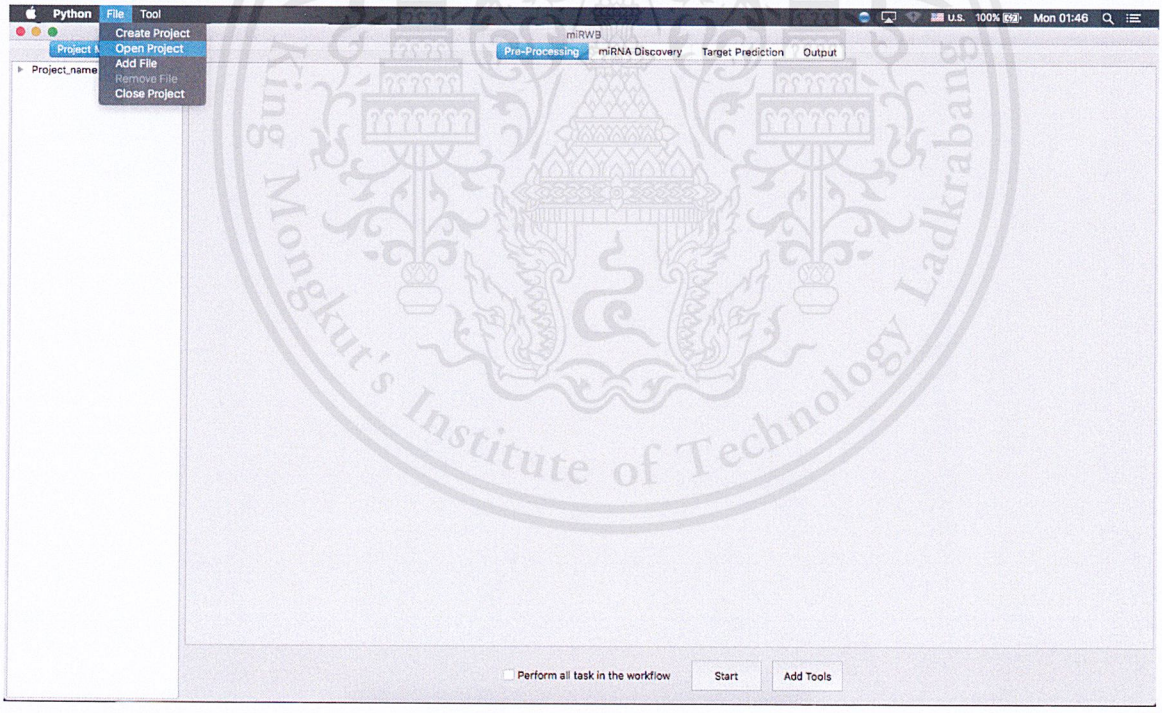

Figure 5.3: File menu then open project

The project management is show on the left side of the program. From this point on the project has been open. Researcher can start adding file into a project. Researcher can add file into the project by press File menu then add file menu (Figure 5.4).

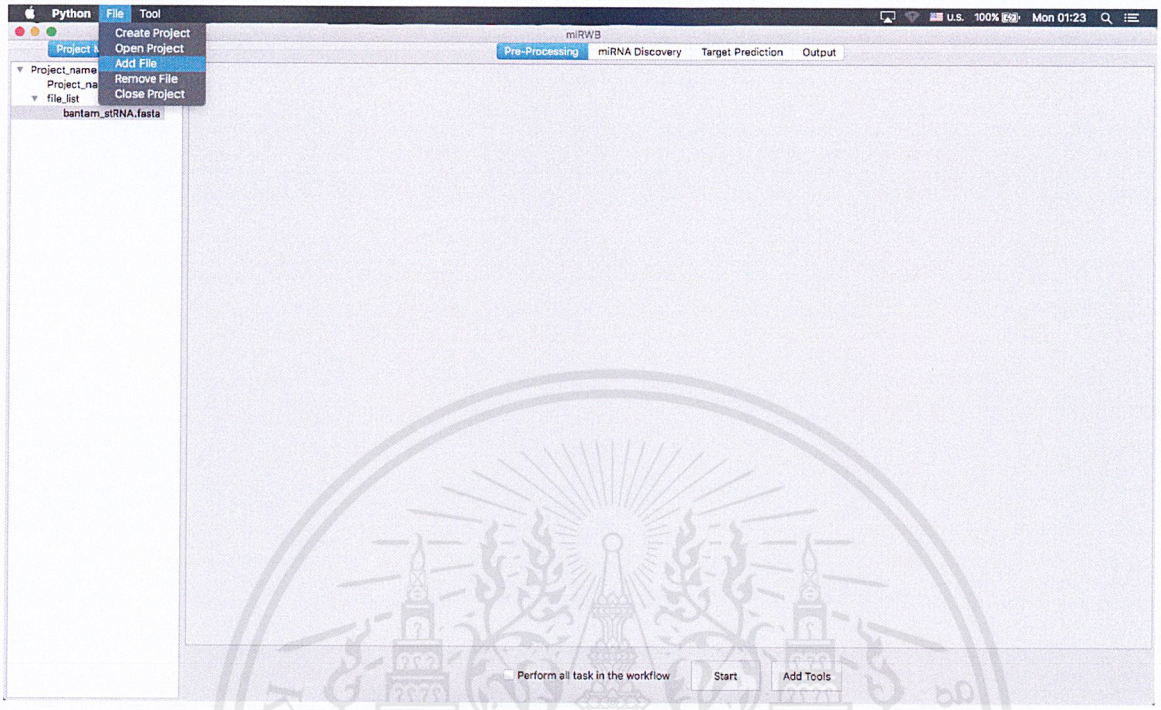

Figure 5.4: File menu then add file

The file selection dialog will open for researcher to pick a file (Figure 5.5).

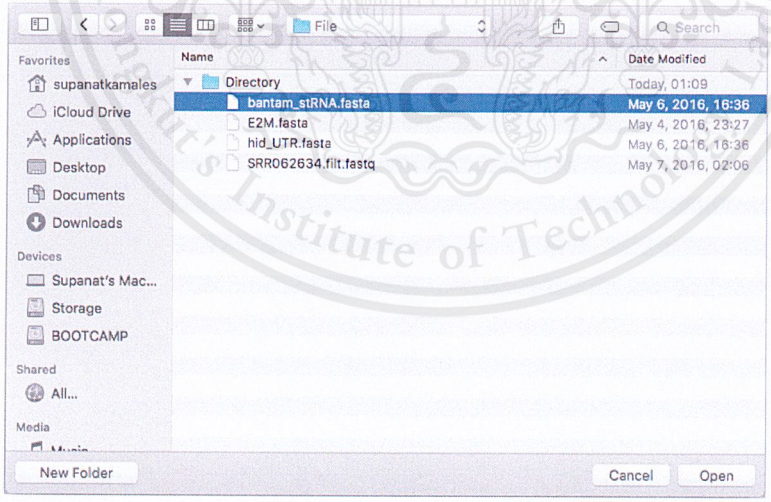

Figure 5.5: press ok to copy the selected file to the project

After researcher pick a file. The file selection dialog will disappear and question dialog will appear instead. The question dialog will ask researcher if researcher wanted to copy the file to the project directory or not. Researcher can copy the file to project directory by press ok. Researcher can remove the file from the project by select the file in the project directory. Then press File menu and remove file.

5.2 Customizable Workflows

After creating a project earlier. Now researcher can use every features in this program. MirWorkbench can improve the working experience of the researchers by typing the command for the researcher. All researchers have to do are to enter the correct parameters for each selected tool in the program. MirWorkbench allow researchers to create their own workflow by add the tools that researchers wanted to use. Researcher can do so by press add tools button on the bottom of the windows. Now tool block will appear on the center of the windows (Figure 5.6).

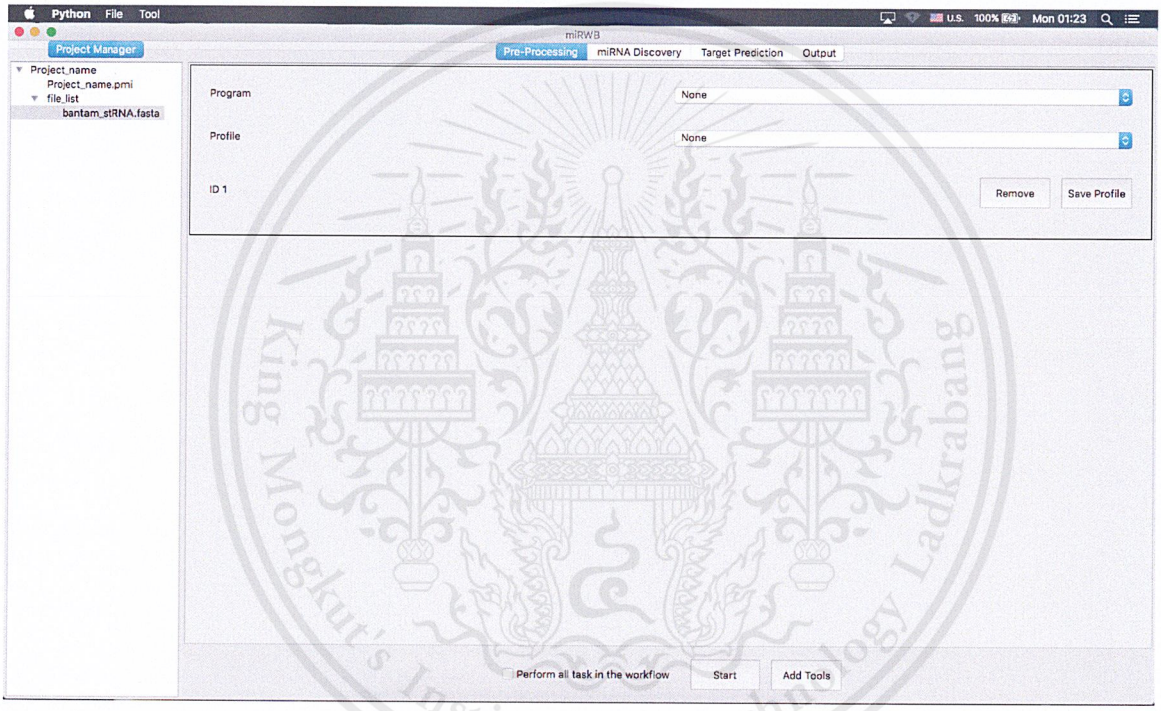

Figure 5.6: add tool

There are two combobox in the tools block. One for select the tools. Another for select the profile. After select the tool, researcher have to enter all required parameter. The required parameters are marked with the * sign on its name (Figure 5.7).

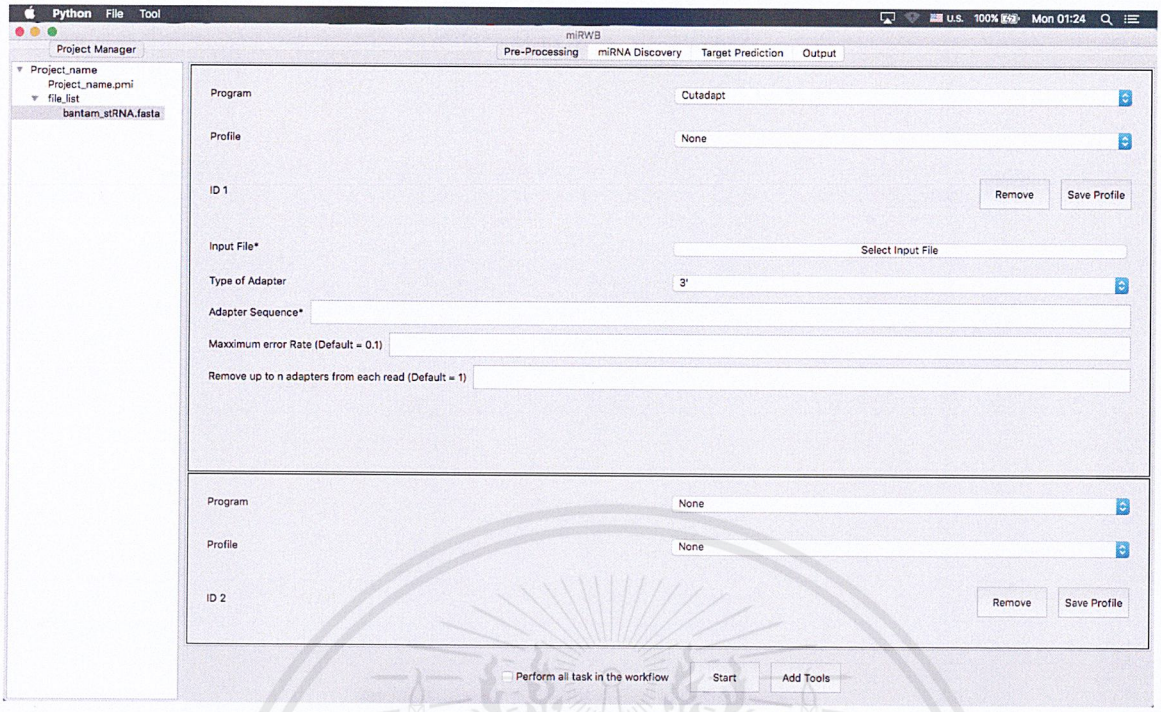

Figure 5.7: Enter parameter

When user press select input button. The file selection dialog will appear. The file mark with * sign is an expected output that generated by tools that execute before current tool. Researcher can record the parameters that researcher often used in to the profile. So next time, researcher don't have to retype it again. User can add as many tools as user wanted. After researcher done adding the tools and enter all parameters. Researcher can execute the workflow by press start. Researcher can execute all task by check the perform all task checkbox on the bottom of the windows. If researcher want to execute current step only. Researcher has to leave this checkbox uncheck.

5.3 Results Visualization

This section will show some result of the tools execute by this program.

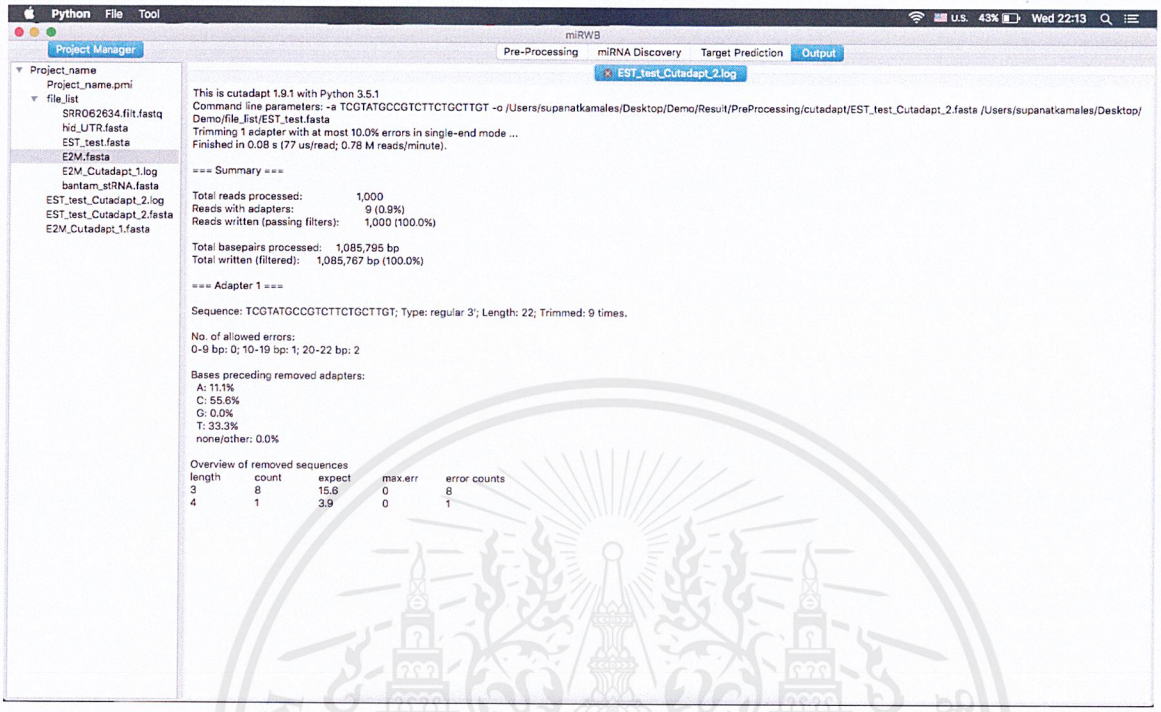

Figure 5.8: Log file generated by Cutadapt tool

Figure 5.8 is the report file generated by cutadapt algorithm. The first line show the version of the cutadapt and Python. The second line show the cutadapt's command along with parameters. The last line before summary show the time the tool used to process. The summary part show the total reads that have been process. In this case is 1,000 reads. It then show the total number of reads that contain adapters. Cutadapt also remember the number of base pairs that have been process. The adapter that has been pass to the command also written within the report file.

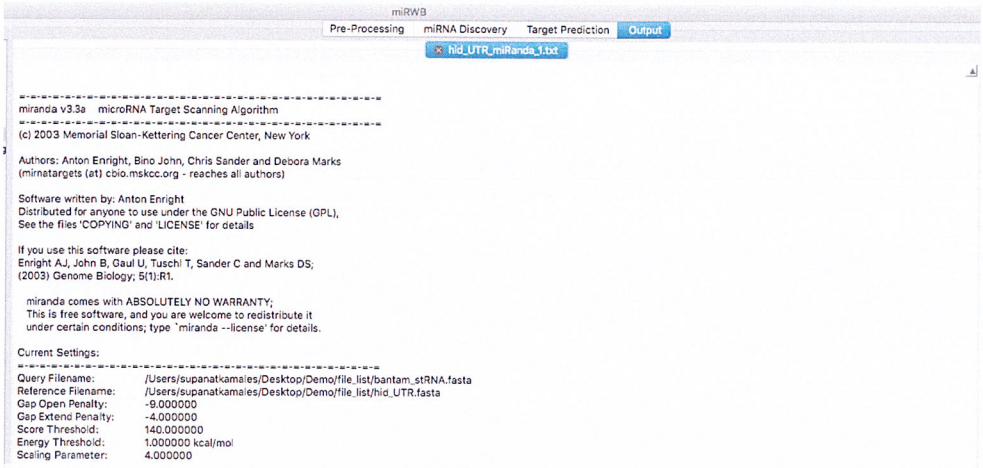

Figure 5.9: First half of the result file genereted by miRanda

Now we move on to the next example. Figure 5.9 show the first half of the report file generated by tool name miRanda version 3.3a. miRanda is a target binding site scanning tool that use in the last step of the workflow. The report file show the information about itself and the parameters that used to run this command. The second half of the report file is shown in figure 5.10. The result file contain an information about the potential binding site for miRNA that miRanda found. It show the score of the hit and other information necessary for the miRNA binding site analysis.

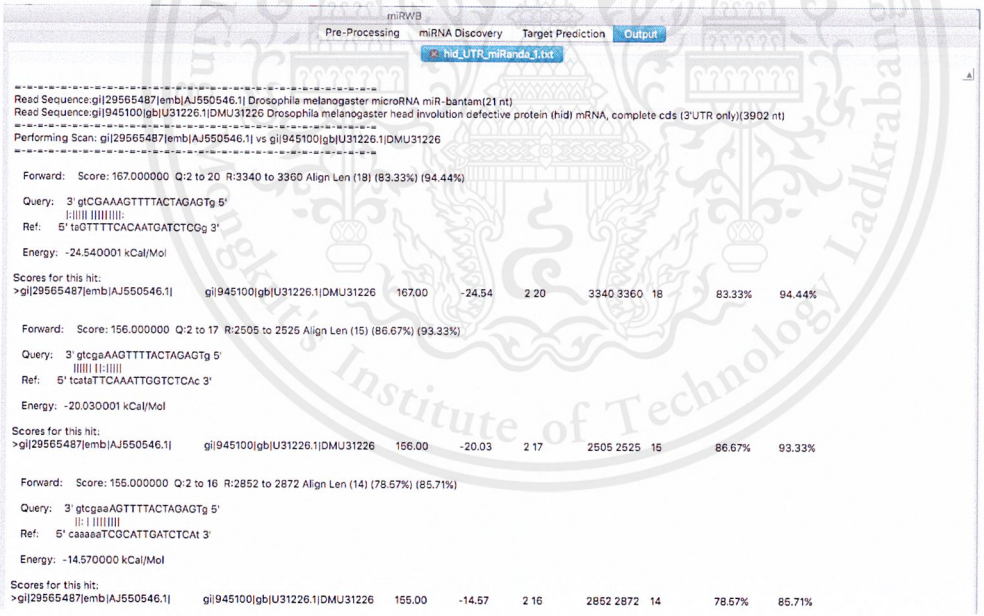

Figure 5.10: Second half of the result file generated by miRanda

Chapter 6

Conclusion

There are four step in miRNA data analysis workflow the first step is Pre-processing. In pre-processing the input data will be cleanse, all contamination will be remove, adapter will be trimmed out of the sequences and nucleotide with low quality will also be remove from the sequences. The next step in the workflow is miRNA discovery. After pre-processing, the known and novel miRNA will be indentified. The third step of the workflow is target prediction. This step will identify possible miRNA binding site on the target genes. The last step is visualization of target prediction result. To cover all of this, a lot of tools is needed. So researchers have to remember and type a lot of command. Researchers have to type the command again if there is a typo in the command. To make things more convinient. A program with graphic user interface that have tools require for miRNA data analysis and file management is needed.

To solve this problem, MirWorkbench is created. By providing a project management tool. Researcher can create project and put all relevent files in one place. MirWorkbench also support workflow management module. MirWorkbench combine several tools needed for miRNA data analysis in one program. Researcher can add tools to the workflow, enter the parameter and execute them easily. After the tools finish, the program will put the result into the project automatically.

For the future work. There are two problems that might interesting. The first one is the thread synchronization algorithm. Right now the current version of miRworkbench use a very simple lock release synchonization algorithm. Becuause the program doesn't require the two tools to execute or access the shared resource at the same time . So lock release which simple to implement but solve the problem nicely is the most suitable algorithm at time. However there might be more suitable synchronization algorithm in the future. The second problem is the visualization. Right now the program cannot render the pdf file and cannot generated the graph that summarise the result of the execution. MiRworkbench will be a lot more flexible if it support pdf render becuase pdf can render so many kind of output such as images, text, equation, graph, table etc.

Appendix A

Software Development Tools

In order to develop some software, a lot of external tools are needed. There are 3 main external tool that were used to develop this program.

A.1 Python

Python is a programming language that is very popular amoung the researcher that research in bioinformatic field. The syntax of python is very easy and python also support a lot of external tools.

A.2 PyCharm IDE

Pycharm is a Integrated development environment. The different between native python editor and Pycharm is Pycharm has a lot of features that native python editor don't have such as debug mode, generate diagram, full support version control (git and mercury), etc. So compare to native python editor which provide only a editor and builder. Pycharm is superior in every way.

A.3 BitBucket

BitBucket is an online version control environment that allow user to create repository to store the code. Version control is used to store the code in version if current version is broken. Version control can reverse the code to some checkpoint. Developer can create those checkpoint by select a file to commit then commit. After commit the checkpoint is create in a local system. To make it global, developer has to push the code to the server.

Appendix B

Installing the MirWorkbench

B.1 Software Requirements

- Operating System: Ubuntu 12 or higher, Mac os X Maverick or higher
- Python v.3
- PyQt v.5
- Sip v.4

B.2 Installation Procedure

to install miRworkbench, researcher has to type

- python setup.py build
- python setup.py test
- python setup.py install −−record installedFiles.txt

or quick setup

- pip install pymirna

After the installation is finished. The researcher can run the program by typing

- import pymirna
- pymirna.runApp()

Note that the program required python version 3.0 or higher, sip and PyQt5. The program also required Unix based operating system since some of the tools are unix based

Bibliography

[1] M. R. Friedländer, S. D. Mackowiak, N. Li, W. Chen, and N. Rajewsky, "mirdeep2 accurately identifies known and hundreds of novel microrna genes in seven animal clades," *Nucleic acids research*, vol. 40, no. 1, pp. 37–52, 2012.

[2] M. Hackenberg, M. Sturm, D. Langenberger, J. M. Falcon-Perez, and A. M. Aransay, "miranalyzer: a microrna detection and analysis tool for next-generation sequencing experiments," *Nucleic acids research*, vol. 37, no. suppl 2, pp. W68–W76, 2009.

[3] W.-C. Wang, F.-M. Lin, W. C. Chang, K. Y. Lin, H.-D. Huang, and N.-S. Lin, "mirexpress: analyzing high-throughput sequencing data for profiling microrna expression," *BMC bioinformatics*, vol. 10, no. 1, p. 328, 2009.

[4] V. Buffalo, *Bioinformatic Data Skills: Reproduceable and Robust research with opensource tools.*

[5] Y. Li, Z. Zhang, F. Liu, Q. J. Wanwipa Vongsangnak, and B. Shen, "Performance comparison and evaluation of software tools for microrna deep-sequencing data analysis," *Nucleic acids research*, pp. 1–8, 2012.

[6] K. Somboonviwat, N. Kaewkascholkul, and K. Somboonwiwat, "mirna workbench: a computational toolkit for mirna identification and target prediction experiments," in *The 13th International Conference on Bioinformatics 2014*, 2014.

[7] E. Korpelainen, J. Tuimala, P. Somervuo, M. Huss, and G. Wong, *RNA-seq Data Analysis Practical Approach.*

[8] S. Griffiths-Jones, "The microrna registry nucleic acids research," *Nucleic acids research*, vol. 32, pp. D109–D111, 2006.

[9] A. Kozomara and S. Griffiths-Jones, "mirbase: annotating high confidence micrornas using deep sequencing data," *Nucleic acids research*, vol. 42, no. suppl 1, pp. D68–D73, 2014.

[10] A. Kozomara and S. Griffiths-Jones, "mirbase: integrating microrna annotation and deep-sequencing data," *Nucleic acids research*, vol. 29, no. suppl 1, pp. D152–D157, 2011.

[11] S. Griffiths-Jones, H. Kaur Saini, S. van Dongen, and A. J. Enright, "mirbase: tools for microrna genomics," *Nucleic acids research*, vol. 36, no. suppl 1, pp. D154–D158, 2006.

[12] S. Griffiths-Jones, R. J. Grocock, S. van Dongen, A. Bateman, and A. J. Enright, "mirbase: microrna sequences, targets and gene nomenclature," *Nucleic acids research*, vol. 34, no. suppl 1, pp. D140–D144, 2006.