

# ODIN: Protein Quality Assessment Tool

Gilbert A. El Khoury

Lebanese American University

**Contents**

Abstract	3
ODIN	4
Materials and Methods	6
Materials	6
Methods	8
Conclusion	11
References	12
Appendix	13
Tables	13
Images	1

### **Abstract**

ODIN is web tool that allows the user to assess the quality of proteins' structures. With the help of a wide variety of integrated tools and web services, ODIN can evaluate the protein structure in question with high accuracy and speed while providing vital information such as number of disallowed residues, chemical shift scores as well as many others. Not only does ODIN combines the various set of tools that measure the protein's quality, but also it provides a consensus score that reflect the overall quality of the protein depending on the set of tools selected by the user. Of the many online tools available that calculate the quality of proteins', ODIN integrated four tools to its server. These tools include dDFIRE, QMEAN, PROCHECK, and SHIFTX. This paper introduces ODIN as a web server while describing all the scripts involved in the process. In addition, this paper will describe the framework of the server, by describing the back-end and front-end details, and the organization of the web tool as well. The paper also displays the results of the variety of tools selected by the user with both high accuracy and speed all while requiring minimal input from the user.

*Keywords: ODIN, Web service, Tools, Protein, Protein Structure, Accuracy, Speed, Score, Consensus, Result, Quality, Quality assessment, Disallowed residues, Chemical Shifts, Minimal input.*

## ODIN

Proteins and their structures entail all biological processes in each living organism. These highly complex molecules are the corner stone of many biological fields and disciplines. Due to that importance, many scientific fields have emerged to tackle the various features of these complexes, some of which are the structural biology and structural bioinformatics fields. The advancements and developments in molecular structure determination techniques have been clearly noted. This allows structural biologists to infer that the functions of macromolecules are highly correlated rather result directly from a protein's particular native confirmation or tertiary structure. On that notion, the determination of protein structure has been the focus of many fields specifically the fields of structural biology and structural bioinformatics, both of which aim to identify the tertiary structure of a protein from its primary sequence. While many research labs are able to successfully determine the native conformation of various complexes in question, they failed to experimentally generate the structures of others. Thus, a new approach was taken to compensate for this failure. This new approach is based on building models that predict the structure of a protein when it folds, which is in other words, the protein's native confirmation. Such predictions are mainly dependent on multiple variables such as secondary structure, thermodynamics, hydrogen bonding, disulfide bonds, chemical shifts, hydrophobic interactions, and many others. Hence, there is a powerful urge to use computational power in order to process and hence predict many proteins structures in question. Although structure prediction is far from perfect, many tools offer very accurate results and predications such as I-TASSER. Thus, scientists rely on such tools to complete their research. Nevertheless, the only problem with this approach is that many of the structures used today are predicted. With all the predicted protein structures, the question turned from being "what is the structure of the protein?" to "how good is

this predicted structure of the protein?”. This highlights the need to build tools that study the stability and the properties of the predicted and the experimental protein structures as well, which is referred to as protein quality assessment. This report introduces ODIN, a new web-service that is able to perform such a task with high precision and swiftness. Having the capacity to integrate and correlate a wide set of quality assessment tools and web services, ODIN promises to deliver a quality assessment that feature the latest updated tools and conclusive scoring of proteins with ease, efficient speed and accuracy. The newly built web service includes many features that facilitate the user experience and user interface with the tool at hand while keeping the user’s interference at minimal levels. This paper discusses the tools features, compositions, as well as its workflow.

## **Materials and Methods**

To get familiar with ODIN, this paper will start by introducing the tiny cogs that lead to the formation of this tool. This section aims to introduce the reader to the components of the tool starting by the basal composition followed by a detailed description of these components and to conclude with a summary of how these pieces are linked and function coherently. ODIN portrays a well understood and compartmentalized architecture, such feature consent the tool to follow a faster, smoother, and error-free flow. The tool's run is completely autonomous allowing the tool to be used even when major updates are occurring. In addition, ODIN implements multiple programming languages, each for specific purposes described in table 3, ODIN make sure to use the specific specialties of each programming language to perform more efficiently. Hence, one can surely say that ODIN is able to provide accurate results in a fast, consistent manner due to the continuous updates to the tools it integrates, the possibility to continuously add new tools independently of old ones, and of course the optimal use of each programming language.

### **Materials**

The tool is composed of five primary folders and six primary scripts represented in table 1.

These six primary scripts file include the drop page or the home page of the website (Index.php), user account page (User2.php) as well as four other error handling .php files. As for the five primary folders, each takes on a particular function ODcss folder, for example, is concerned with the styling of ODIN webpages elements. It contains pure CSS snippets that are able to showcase the website in its shape. ODimages is the folder containing all the images used by the tool including the tool logo. As for the /php folder, it contains .php scripts that allow the user to interact with the tool: submitting jobs, logging in, logging out, signing up, accessing the database to see results history, and many others such scripts that are described comprehensively in table 2

due to their significance. The folder results include all the results generated by the jobs submitted and is used as a depository for the tool to access previous user's result. As for uploads and processed folder, they are the containers which will hold the uploaded and downloaded PDB files for further processing or backup purposes. A successful job submission will direct PDB files to the tools folder where multiple tool scripts would access such files and call specific tools to perform on that PDB file.

The composition of the /tools folder is displayed in table 4 with description of its content. The compartmentalization process followed by ODIN allows it to keep track of each job on its own while reducing clashes between different jobs submitted by various users which additionally helps in debugging and testing purposes. In other words, the tool can still be used while its being updated. ODIN components show high variability since each script or code is optimized for its specific purposes. In summary, all scripts dealing with client side-server side cross talk are written in PHP file, all database queries are written using SQL, particular file handling and extraction as well as communicating with APIs are written in Python, and last but not least the call for particular tools and the localization, creation of directories is made using BASH (shell scripts). A detailed description of languages, framework, tools used and folders components can be found in tables 1,2,3,4.

### Methods

After launching the website, the user would get access to the index.php file which is the main homepage of the tool. The function of the homepage is to introduce the user to ODIN web tool, display certain differential advantage over other pre-existing tools, showcase the makers of the tool and finally allow the user to either signup for a new account or sign in to his old account (see figure 2). In case of sign in or sign up, the appropriate script would be called from the folder /php (see table 2). In both cases, a successful sign up or sign in, the user is finally redirected to his account page which is User2.php (see table 1) thus the information given at the level of homepage are either introduced at the database ODIN after careful security checks (on both front-end and back end levels) in case of sign up using the script ODsignup.php (see table 2) or assessed from the same ODIN database and, then, if the authentication is valid, the user would be redirected to his account page using the script ODsignin.php. In case of failure in signing in, the user is redirected to the homepage with a specific error message. Once the user has access to his account page, he/she would have a set of information to guide him/her on how to submit a job and access to his/her job history as well access to a form which allow him/her to submit new jobs (see figure 3). The user2.php has all the information the user needs in order to start a job as well as a set of help button to guide him/her when selecting the tools. In case of error submission, the user would be prompted of the specific error and the job won't be submitted (front-end security checks). If the user was able to bypass the front end checks, the PHP scripts called would handle these errors as well (back end checks). When submitting a job with ODIN, the user has a possibility of two options (see figure 3): either to submit a PDB file using an upload button or to submit the job using a PDB ID which will download the PDB file from the PDB database if found. Otherwise, the user is notified that the PDB ID used is invalid. In any of these cases, the



user is again presented with three choices: either to go for all the chains in the appropriate PDB file (either downloaded or uploaded), or to select one chain from the PDB file, or even to select multiple chains all at once. PDB Chain selection is handled in the ODuploadfile.php and ODdownloadfile.php scripts which will check if a chosen chain for a particular PDB file exists. If such file does not exist, the job won't be added to the ODIN database and hence won't be executed. In the case where the PDB chain ID provided was only one and it was erroneous, the user will be notified about the invalid chain ID used. Otherwise, the scripts will simply ignore any invalid PDB chain ID. The tool proceeds to check if any of the tools are selected, the tools available are PROCHECK which generate a Ramachandran plot as well as provide an insight about the allowed and disallowed residues, QMEAN which offer a normalized score of how good the quality of the protein is, assigning its score with 4 and 6 different variables including solvation, torsion, etc. The tool incorporates two tools which are dDFIRE and SHIFTX where the first calculates an RMSD score calculated from ab initio folding of misfolded parts of the proteins, while the second one calculates the chemical shifts with respect to various atoms (see table 4). At least one of these tools should be selected; multiple tools could be selected at once, however, if none of these tools are selected, ODIN notifies the user about the error and stops the submission. Hence, for a successful submission of a job, the user must validate the PDB ID in case he/she is using the download option or should submit only PDB file in case of using the upload button. The user has the option to either use all the PDB file, a specific chain (this chain should be valid) or multiple chains (invalid chains are ignored), then the user has to select desired tools. At least one tool must be selected, failure to meet any of the above conditions will halt the submission and warn the user about the specific error until its fixed. Otherwise, the submission is successful and at this stage ODuploadfile.php or ODdownloadfile.php will add to

the ODIN database (see figure 4) the job specifying a specific ID for the job, the id of the user who submitted the job, the tool selected for the job, the date of submission and chain specified, in case no chain is specified a dummy “ALL” string is used to denote such cases. Upon successful addition to ODIN database, the tools redirect to the script ODexecute.php in the background while showing a loader in the front end. Once executing the ODexecute.php script, the tool will start by extracting the specified PDB chains from the PDB file, if specified, using the script extract.py (see table 3), then, it will read from the ODIN database all the incomplete user submitted jobs that are active and based on the tool specified for each job the script is called upon. If PROCHECK is the tool specified for this job, the ODexecute.php will call run\_procheck.sh. As for QMEAN, the script will call for Qmean.py script. Similarly, in cases of dDFIRE, run\_dDFIRE.sh is executed or in case of shiftX, run\_shiftx is called. Once the specified tool is executed, the ODIN database is updated to set these jobs status as completed. After completion of all the jobs, getresults.py (see table 4) is executed to fetch the results of each tool and build the specified result page then send it to /results folder and redirects the user to it. At the result page, the user has the option to check other job’s results or run another job or download current job. In case the user selects one of the first two options, he/she will be redirected back to user2.php, where in option one, he/she would have access to all his/her previous results to select from. However, note that user2.php would automatically remove result pages from the database after 2 months from time of submission in addition to all their related job information. To allow full disclosure, ODIN users can track job by job status featuring the file name, chain selected, tool selected, status of the job, date of submission, and the date by which the job was complete as well the ability to cancel, delete a job from the history or even deleting the whole job runs history. To wrap it up, the process of using ODIN is simplified and represented in figure 1.

## Conclusion

Since the central dogma of molecular biology has been established, sequence dictates structure, structure dictates function, both structural biology and structural bioinformatics disciplines are set on the race to develop the best tools and algorithms to predict protein structures and since predicting proteins tertiary structures is usually associated with a huge amount of possible errors simply because the mechanism of protein folding is still not fully explored and documented. Structure prediction allowed the establishment of a subfield in structural biology which is protein quality assessment. Although during the past few years both biology and computer science fields witnessed huge advancements and improvements, many protein quality assessment tools were either not regularly updated or not maintained anymore. Therefore, the need for a new tool submerged, a tool that integrates a wide variety of quality assessment programs and websites under one website which allows the user to submit multiple queries to multiple programs using only one file submission. ODIN is then able to generate a result page which includes the results of each individual tool while providing a consensus score about the overall protein's quality. ODIN differs from other tools in many ways, either by being able to integrate all these tools together under one website, or by being able to maintain, update and integrate a wide set of QA tools, or by showing a faster result generation. Not to forget also that ODIN offers multiple modes of usage (File Upload, PDB File download, specific chain study, multiple chain study, whole protein study). ODIN demonstrates ease in both user interface and user experience all while maintaining a fast, reliable protein structure quality assessment by implementing multiple programming languages and a reliable organizational structure.

## References

Benkert, P., Tosatto, S.C.E. and Schomburg, D. (2008). "QMEAN: A comprehensive scoring function for model quality assessment." *Proteins: Structure, Function, and Bioinformatics*, 71(1):261-277.

Laskowski R A, MacArthur M W, Moss D S, Thornton J M (1993). PROCHECK - a program to check the stereochemical quality of protein structures. *J. App. Cryst.*, 26, 283-291.

Stephen Neal, Alex M. Nip, Haiyan Zhang, David S. Wishart (2003) "Rapid and accurate calculation of protein <sup>1</sup>H, <sup>13</sup>C and <sup>15</sup>N chemical shifts" *Journal of Biomolecular NMR*, 26:215-240.

Yuedong Yang and Yaoqi Zhou, Specific interactions for ab initio folding of protein terminal regions with secondary structures. *Proteins* 72, 793-803 (2008)

Yuedong Yang and Yaoqi Zhou, Ab initio folding of terminal segments with secondary structures reveals the fine difference between two closely-related all-atom statistical energy functions. *Protein Science* 17 1212-1219, (2008).

## Appendix

### Tables

Table 1

Main Composition of ODIN

Folder+Files	Main Role	Content
ODcss	This folder is designed to control the styling of the website	1 file
ODimages	This folder contains all the images displayed for design purposes (including logo)	1 folder + 8 files
php	This folder is mainly to handle all the server side client side interaction and connect to the database	9 files
uploads	This is the folder where all the submitted PDBs will exist	NA
processed	This is a backup folder of all the submitted PDBs	NA
tools	This folder contains all the tools and scripts that assess the PDB files and generate the results	4 folders + 8 scripts
results	This folder will hold the results generated by the tool	NA
fde.php	This is the page to be redirected to in case the user input a wrong PDB ID	NA
fue.php	This is the page to be redirected to in case the file uploaded by the user is invalid	NA
fue2.php	This is the page to be redirected to in case the user did not select a tool	NA
fe3.php	This is the page to be redirected to in case the user input a wrong PDB Chain	NA
user2.php	This is the main login page of the user (user account page)	NA
index.php	This is the drop page (before any signing up / signing in)	NA

*Note: as seen in this table, ODIN integrate a very well organized structure and compartment allowing it to carry its function both fast and with accuracy and avoid mistakes. (the files fde.php, fue.php, fue2.php are kept for backend validation purposes only, in other words they would never be visited unless the front-end checks fails)*

Table 2

Main Composition of /php

Folder+Files	Main Role	Language
ODsignup	Script that validate the user's location and information and adds the user to ODIN database (while hashing his information using PHP hash function)	PHP
ODsignin	Script that validates user credentials and allow the user to login to his account (in case of failure the user will remain at the login page)	PHP
ODuploadfile	Script that allow the upload a file from the user to the /uploads, /tools, /processed folder after security checks on file extension, file size, tools selected and chains ID, if everything is safe adds the job to the ODIN database in order to be executed	PHP
ODdownloadfile	Script that validate the existence of the submitted pdb ID, if exists it will download it in the /tools folder then in case chains are specified it will extract all chains and then check selected tools to add them to the ODIN database to be later executed	PHP
ODexecute	Script that gather from the database the active successful job submission (File name, tool selected , chains, user ) and runs the specified job	PHP
ODsignout	Script that deletes all the cookies of the user and redirect to the homepage hence ending the user's session	PHP
ODcanceljob	Script that access a specific job submitted in the ODIN database and set it as inactive preventing its execution while canceling its execution only	PHP
ODdeletejob	Script that access a specific job submitted in the ODIN database and set it as inactive preventing its execution. This job will be then set as inactive and hence won't appear in the user's history	PHP
ODdeletealljob	Script that access the ODIN database changing all job submitted jobs by this user to inactive and hence they won't appear in the user's history and they won't be executed	PHP

Table 3

Languages and their uses

Language	Main Role
PHP	Used to Display the front-end UX and UI to the user as well to communicate with both databases and server side, used as well for back-end checking and some website to website communication
SQL	Used by PHP scripts to update the ODIN database Information and hence control job submission, execution, allowing the website to acquire updated information
BASH	Used to create directories, copy files, change directories facilitating file transfer and localization, in addition used to call some of the tools binary
Python	Used to handle file reading, extraction, as well calling some tools and communicating with REST APIs and Web servers
JAVASCRIPT	Used mainly to animate the website and provide smooth UX and UI as well as integrate some error handling and front-end checks
CSS	Used to maintain the style of elements in the web page its use is primarily to add beauty to the website
BOOTSTRAP	Used as a framework to build the website on allowing the website to maintain a smooth coherent structure while offering the ability for the website to be accessible from mobile phone.

Table 4

Main Composition of /tools

Folder+Files	Main Role	Languages
dDFIRE	This folder contains dDFIRE executables and each dDFIRE job results in a folder	NA
PROCHECK	This folder contains PROCHECK executables (procheck.sh) and each PROCHECK job results in a folder	NA
Qmean	This folder contains a folder that gathers all the results of submitted QMEAN jobs	NA
shiftx	This folder contains ShiftX executables and each ShiftX job results in a folder	NA
download.py	This is the script that download PDB files from PDB if the use PDB ID was chosen instead of Uploading a file (arguments: PDB ID)	Python
extract.py	This script is called to extract all possible chains from a PDB files and generate new PDB files (arguments: PDB ID, CHAIN ID) none existing chain are not extracted	Python
getresults.py	This script assembles the various results generated by each selected tool and calculate a consensus scores about the protein quality and generate Result page (arguments: PDB ID, Tools selected, User ID, URL)	Python
Qmean.py	This script communicates with the QMEAN server send job request and return results, results returned are then extracted and set in a subfolder with the name of the pdb file submitted under the Qmean folder	Python + REST API
run_procheck.sh	This script sends a copy of the PDB file to be processed to the PROCHECK directory and calls upon the executable procheck.sh which will generate the procheck results, these results are then transferred to a subdirectory in the procheck folder with the name of the submitted job file.	BASH
run_dDFIRE.sh	This script sends a copy of the PDB file to be processed to the dDFIRE_bin directory and calls upon the executable dDFIRE which will generate the dDFIRE results, these results are then transferred to a subdirectory in the dDFIRE_bin folder with the name of the submitted job file.	BASH
run_shiftx.sh	This script sends a copy of the PDB file to be processed to the shiftx directory and calls upon the executable SHIFTX which will generate the SHIFTX results these results are then transferred to a subdirectory in the SHIFTX folder with the name of the submitted job file.	BASH

*Note: details about how each script function can be read in the script files themselves.*



# ODIN: Protein Quality Assessment Tool

## Images

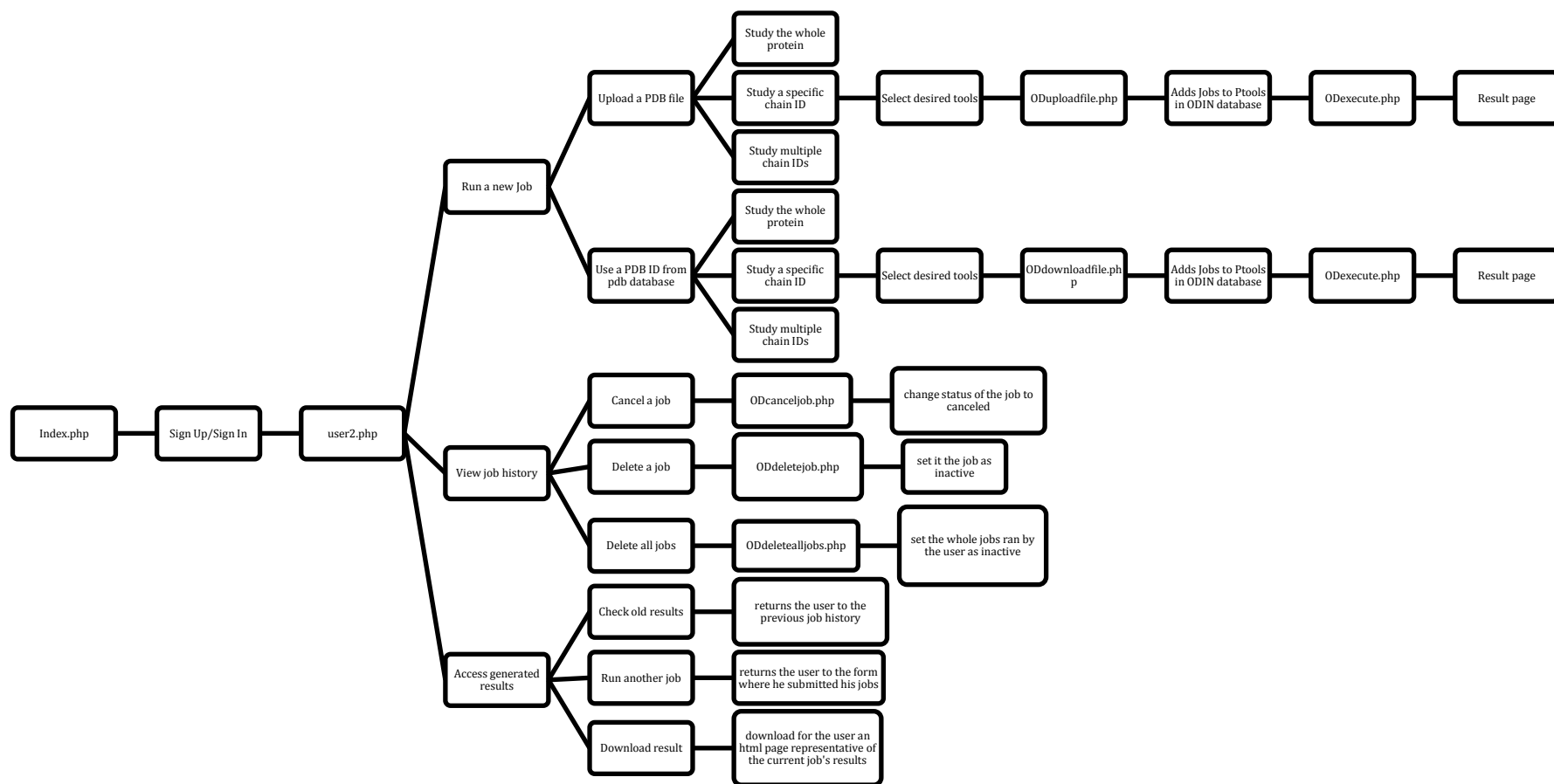


Figure 1: graph showing the workflow of ODIN tool

# ODIN: Protein Quality Assessment Tool

ODIN

[About](#) [Characteristics](#) [Makers](#) [Sign Up / Login](#)

START

Do You Have A Protein Structure File

YES

NO

Upload Your PDB File

Pick The Tools You Wish To Use

Read their descriptions

Submit The Job

Insert a 4 letter PDB ID

DONE!

Wait For A Result Email

ABOUT ODIN

ODIN is a web server designed to provide you the ability to assess the quality of your protein tertiary structure at ease and with minimal interference. In other words, having integrated a large set of widely used and known protein quality assessment programs, ODIN is capable of fast delivery of a consensus protein quality score representing an overall quality of the protein module submitted.

MORE

Characteristics

Friendly

ODIN is your web tool of choice simply because of its ease to use and efficiency. Rather than navigating multiple website submitting same job at different web servers, ODIN allows you to submit the same job for different web servers allowing you to concentrate on more important tasks for the results will reach you later on by email


Powerful

ODIN could be identified as a better quality assessment webtool not only by being user friendly rather by being ultra fast in addition to being largely updated in order to deliver the fastest quality assessment possible while maintaining a high rate of accuracy by updating and adding new tools making ODIN more precise and more accurate.

Immediate

ODIN takes protein quality assessment a further step forward making sure you are saving time allowing you to choose between a set of the best quality assessment programs and web servers. the results will reach you by email containing the information generated by these chosen web server using only one web tool and one website

The Makers



Mr. Gilbert El Khoury

Mr Gilbert El Khoury is an undergraduate student at the lebanese american university majoring in the field of Bioinformatics while minoring in both chemistry and business. he joined LAU in 2014 and this web server is his capstone project built in 2018

Contact Mr El Khoury

Sign Up

Username

First Name

Last Name

Enter Your Country

Email Address

Password

Retype your Password

Submit

Sign In

Username

Password

Submit

Figure 2:

*Index.php or the homepage of ODIN*

# ODIN: Protein Quality Assessment Tool

ODIN

Run A Job

Check Jobs

Sign out Gix

Step One:

Upload the PDB structure of interest

Step two:

Read description of each of the available tools

Step three:

Select the tools needed to assess your protein structure

Step four:

Wait no more. The result will be sent to you directly by email

Upload the pdb File

Select pdb file to upload:  

Choose File

 No file chosen

Enter PDB ID to use:  

Insert 4 letter pdb ID 4

Enter Chain ID or Leave blank for all chains:  

Insert 1 letter chain or

Seperate the Chain IDs by a comma

Example: AB...

Select the tools

■ Procheck 1

■ Qmean 1

■ dDFIRE 1

■ SHIFTX 1

Start

Description:

1- Please check the chains of your PDB file (The Tool will ignore any invalid chain)

2- It is Important to check that your PDB file do not contain any nucleotide. Otherwise the tool wont show promised results.

3- Once you press start your job is submitted.

Average Waiting Time is 3 minutes

Job History

Your Most Used Tools Are:  
Procheck, Qmean, dDFIRE, SHIFTX

Number Of Completed Jobs is:  
8

Number Of Running Jobs is:  
0

Your Results:

1d1l Time Remaining: 60 Days

4ull:A Time Remaining: 60 Days

History [+]

Number Of Submitted Jobs

8

1d1l	Qmean	2018-12-04 21:05:13	2018-12-04 21:05:20	Completed	Cancel	Delete
1d1l	shifTX	2018-12-04 21:05:13	2018-12-04 21:05:20	Completed	Cancel	Delete
1d1l	dDFIRE	2018-12-04 21:05:13	2018-12-04 21:05:20	Completed	Cancel	Delete
1d1l	Procheck	2018-12-04 21:05:13	2018-12-04 21:05:18	Completed	Cancel	Delete

Figure 3:

User2.php or the user's account page

2 | Page

## ODIN Database Tables

Table	Action	Rows	Type	Collation	Size	Overhead
Ptools	Browse  Structure  Search  Insert  Empty  Drop	12	InnoDB	latin1_swedish_ci	16 KiB	-
Results	Browse  Structure  Search  Insert  Empty  Drop	5	InnoDB	latin1_swedish_ci	16 KiB	-
User	Browse  Structure  Search  Insert  Empty  Drop	6	InnoDB	latin1_swedish_ci	16 KiB	-

## ODIN Ptools Table

1	ID	int(11)	No	None	AUTO_INCREMENT	Change  Drop  Primary  More
2	User_ID	int(11)	No	None		Change  Drop  Primary  More
3	Filename	varchar(10) latin1_swedish_ci	No	None		Change  Drop  Primary  More
4	Chain	varchar(1000) latin1_swedish_ci	No	ALL		Change  Drop  Primary  More
5	Tool_name	varchar(150) latin1_swedish_ci	No	None		Change  Drop  Primary  More
6	Status	int(11)	No	0		Change  Drop  Primary  More
7	Start	datetime	No	CURRENT_TIMESTAMP		Change  Drop  Primary  More
8	End	datetime	No	CURRENT_TIMESTAMP		Change  Drop  Primary  More
9	Active	int(11)	No	1		Change  Drop  Primary  More

## ODIN Results Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	ID	int(11)			No	None		AUTO_INCREMENT	Change  Drop  Primary  Unique  Index  More
2	User_ID	int(11)			No	None			Change  Drop  Primary  Unique  Index  More
3	Filename	varchar(150)	latin1_swedish_ci		No	None			Change  Drop  Primary  Unique  Index  More
4	Chain	varchar(1000)	latin1_swedish_ci		No	ALL			Change  Drop  Primary  Unique  Index  More
5	Completed	date			No	None			Change  Drop  Primary  Unique  Index  More
6	Remove	date			No	None			Change  Drop  Primary  Unique  Index  More

## ODIN User Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	ID	int(11)			No	None		AUTO_INCREMENT	Change  Drop  Primary  Unique  Index  More
2	Username	varchar(150)	latin1_swedish_ci		No	None			Change  Drop  Primary  Unique  Index  More
3	Email	varchar(150)	latin1_swedish_ci		No	None			Change  Drop  Primary  Unique  Index  More
4	Password	varchar(1000)	latin1_swedish_ci		No	None			Change  Drop  Primary  Unique  Index  More
5	Country	varchar(1000)	latin1_swedish_ci		No	None			Change  Drop  Primary  Unique  Index  More
6	Firstname	varchar(1000)	latin1_swedish_ci		No	None			Change  Drop  Primary  Unique  Index  More
7	Lastname	varchar(1000)	latin1_swedish_ci		No	None			Change  Drop  Primary  Unique  Index  More
8	Active	int(11)			No	0			Change  Drop  Primary  Unique  Index  More

Figure 4: ODIN Database Tables and their respective structures