Protein identification with cryoID

Nov 25, 2019

This tutorial provides an introduction to the use of the Linux-based program cryoID for the identification of proteins in near-atomic resolution (< 4 Å) cryoEM density maps. It covers the cryoID graphical user interface (GUI), as well as the two subprograms: get_queries and search_pool. A test dataset is provided with the cryoID package, to be used with this tutorial. CryoID was developed and tested on Linux OS (CentOS 6 & 7).

1 Getting started

1.1 Recommended reading

The rationale, framework and workflow of cryoID are described in detail in Ho *et al*. Bottom-up structural proteomics: cryoEM of protein complexes enriched from the cellular milieu (*Nature Methods*, 2019, doi:10.1038/s41592-019-0637-y).

1.2 Install cryoID

Please refer to our online instructions (https://github.com/EICN-UCLA/cryoID) for the installation of cryoID and related software packages.

1.3 Get your test data

The test dataset includes a cryoEM density map named UnkwnPro.mrc and a file named MS_candidates.fasta containing a list of 760 candidate protein sequences derived from our *Plasmodium falciparum* sample. We also provided our pre-generated results in the dataset. These files are available for download at https://github.com/EICN-UCLA/cryoID_SM.

2 General description

CryoID performs protein identification from near-atomic resolution (< 4 Å) cryoEM density maps by identifying high resolution segments in the map, building short sequences into these segments, and then using these short sequences as a set of query sequences in a customized Blastp search against a pool of candidate proteins supplied by the user. The candidate pool can either be the entire proteome(s) of the relevant organism(s), or a more limited list of proteins candidates identified by mass spectrometry analysis of the sample from which the cryoEM density map was generated.

The following information may be useful for obtaining optimal performance from cryoID.

2.1 Degenerate groups

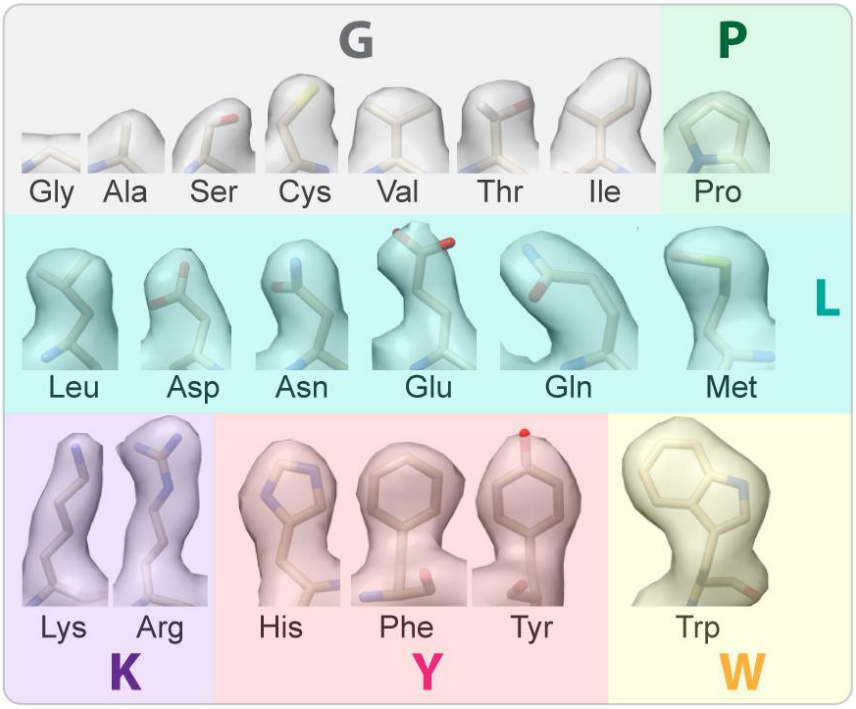

Figure 1 | CryoID Simplified Six-Letter Code

CryoID uses a simplified, "degenerate" six-letter amino acid table (**Fig. 1**) that groups the 20 amino acid residues into six simplified groups based on the similarity of their side chain densities in typical cryoEM density maps. For example, the

segment of the map shown in **Figure 2**, DKKAREYANDALKF, is translated into the following simplified sequence: LKKGKLYGLLGLKY.

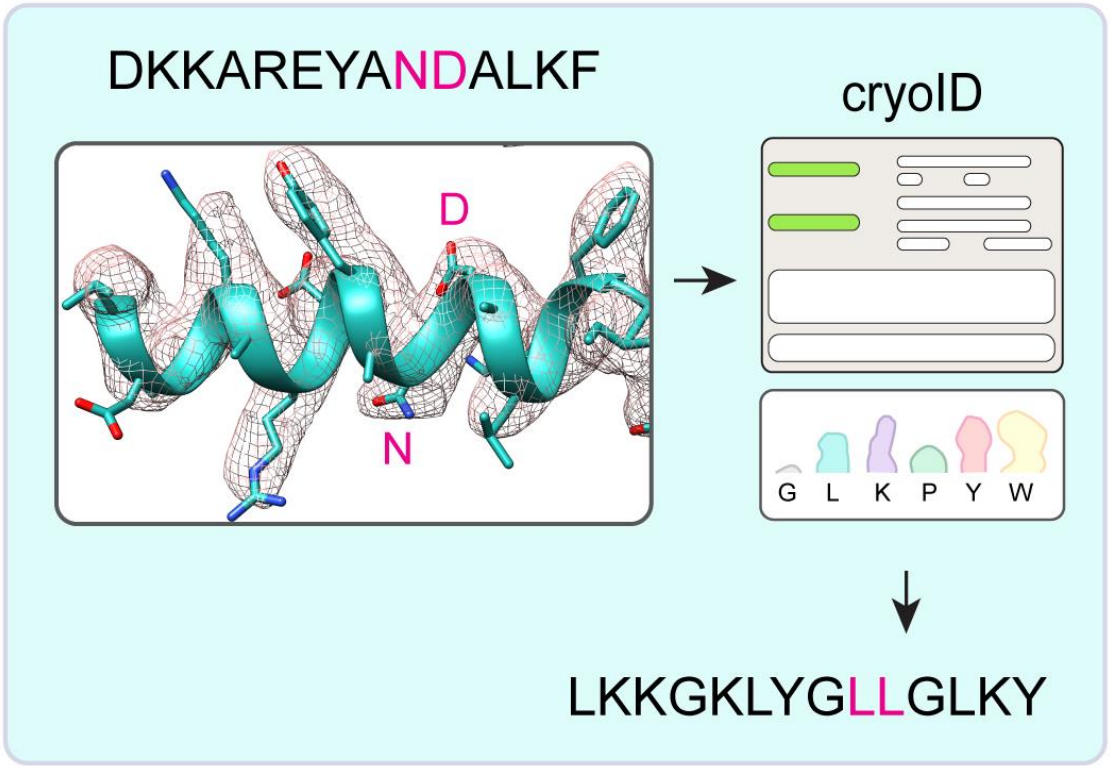

Figure 2 | CryoID Sequence Prediction and Simplification

By using the simplified code, a redundancy was introduced that imparts a certain amount of tolerance for errors made by cryoID during the sequence prediction step. In the example shown above, the densities for N and D look very similar; however, there is no need to differentiate between them in cryoID, since they both fall within the "L" group and will both be translated into the same letter in the simplified code.

Keep in mind that all input sequences are simplified in cryoID based on this table.

2.2 Customized scoring matrix for Blastp search

We created a customized alignment scoring matrix for cryoID by adapting the BLASTP PAM30 scoring matrix so that the substitution scoring matrix used has higher bonus scores for matches to P-like/K-like/Y-like/W-like categories and appropriate penalty scores for mismatches depending on the severity of side chain shape dissimilarity between categories. The table may be useful when the user optimizes the queries.

	G	P	L	K	Y	W	X
G	6	-6	-5	-8	-14	-15	-1
P	-6	8	-4	-8	-13	-14	-1
L	-5	-4	6	-5	-9	-14	-1
K	-8	-8	-5	11	-11	-13	-1
Y	-14	-13	-9	-11	10	-5	-1
W	-15	-14	-14	-13	-5	13	-1
X	-1	-1	-1	-1	-1	-1	-1

Figure 3 | Scoring matrix for blastp search used in cryoID

2.3 Query set reference table

Number of queries	Query Sequence Length			
1	30	45	75	>100
2	20	30	55	100
3	16	23	32	75
4	15	20	30	65
5	13	19	28	55
6	12	18	27	50
7	12	16	25	45
8	11	16	24	40
9	11	15	22	40
10	11	15	21	40
Tolerable percentage errors	**10%**	**20%**	**30%**	**40%**

Table 1 | Reference Table for Optimal Query Set Generation

To guide successful query set generation, we have provided a reference table (**Table 1**), which is derived from simulation results of a candidate protein pool containing 880 proteins. You may refer to this table to find the tolerable percentage errors for certain number/length of queries. If your query set satisfies the table, cryoID should be able to identify the protein(s) in your cryoEM density map with high confidence!

3 Running your job

3.1 The cryoID GUI

The most convenient way to run jobs is *via* the cryoID GUI. The upper half of the GUI contains two separate panels for the cryoID subprograms get_queries and search_pool. Each subprogram panel contains several fields for user inputs. Outputs and errors from both subprograms will appear in the output/error windows located on the lower half of the GUI. An abort button and a job status bar are located in the middle, between the input fields above and the output/error windows below.

To start the GUI, type cryoID & in the terminal command line. The GUI will appear, as pictured in **Figure 3**:

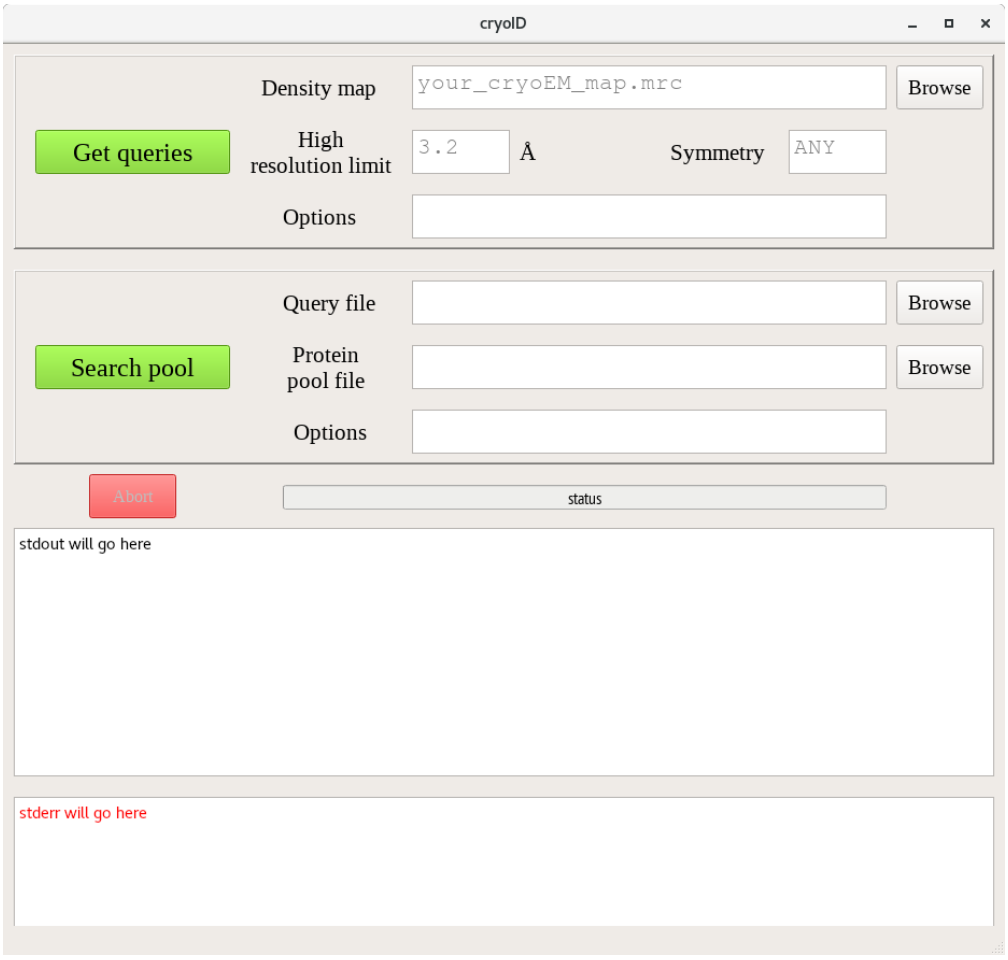

Figure 3 | The cryoID GUI

For help or additional information or options, the alt text for each field can be activated by hovering the mouse over the field you are interested in learning more about, as pictured in **Figure 4**.

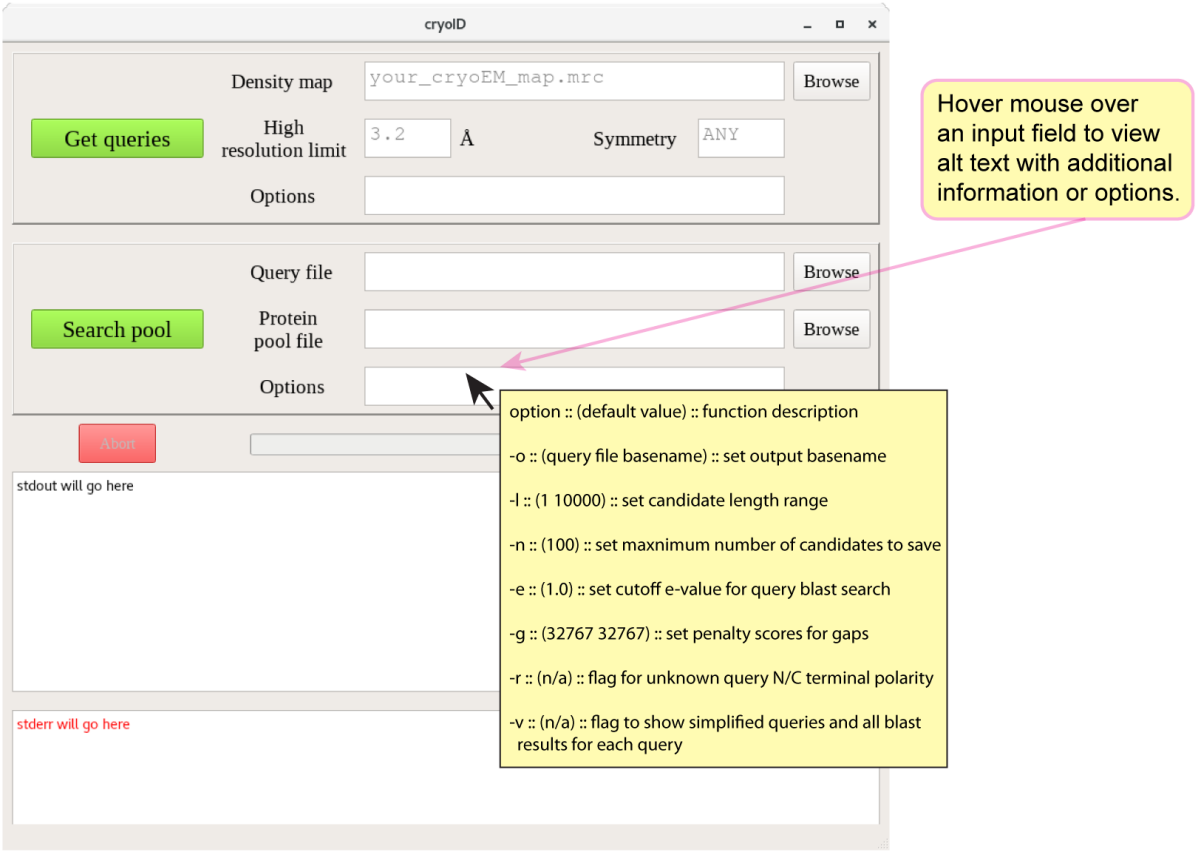

Figure 4 | CryoID Tooltips

3.2 Get queries

The subprogram get_queries identifies multiple promising segments from the cryoEM density map and predicts the primary query sequences for these segments. To run this job for the test data, input your parameters in the input fields on the GUI Get queries panel:

Density map:	UnkwnPro.mrc	*Your cryoEM density map file
High resolution limit:	3.2	*High resolution limit for map analysis (in Å)
Symmetry:	T	*Symmetry of your cryoEM map
Options:		*Input additional options here

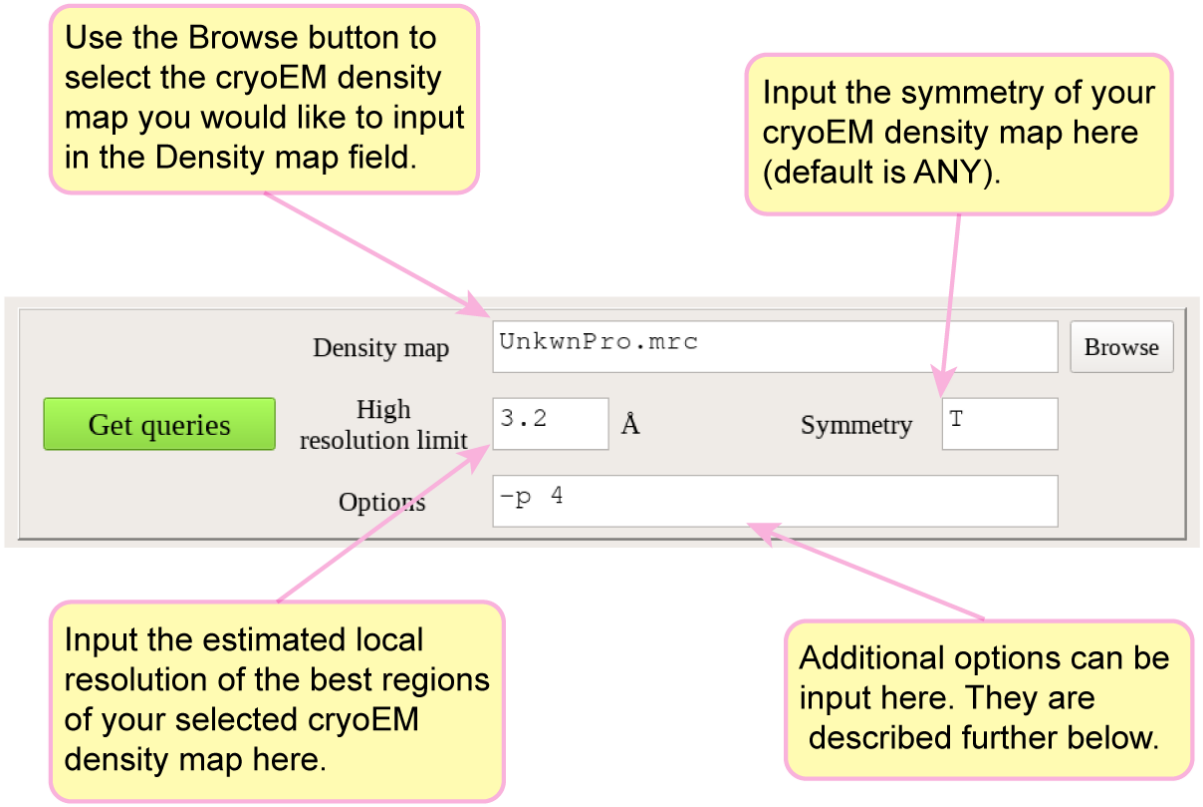

Figure 5 | The cryoID Get queries Panel

Using the **Browse** button, select your Density map (mrc format) first. CryoID performs best with maps that have an overall resolution better than 4.0 Å, but still works for maps at poorer overall resolutions, as long as the map contains a few regions with good enough local resolution and map quality.

The Resolution limit parameter may affect the location/length of queries cryoID can generate from your density map. The overall resolution reported for your cryoEM density map during 3D reconstruction is a good starting point. You may slightly tune this parameter according to the local resolution estimation until you're satisfied with the segments.

Finally, provide the symmetry information, such as T (tetrahedral) or C6 (cyclic 6-fold). The default value is ANY, for which get_queries will try all symmetries and use the highest symmetry found.

There are three additional options for the get_queries subprogram:

Option	Default value	* Description
-o	Input map basename	*Set output file basename
-n	10	*Set maximum number of segments to keep
-p	1	*Specify number of processors you want to use

Click on the Get queries button to run your job. Once the job starts, you will notice that the subprogram panels will become inactive, while the Abort button and Status bar become active. Click on the Abort button if you want to abort the running job. Real-time output will be displayed in the output/error windows in the GUI.

It may take tens of minutes to build query model for a density map with a single processor, depending on the density map and the system the job is run on. Get_queries will generate a fasta file containing the resulting query sequences, as well as a pdb file containing the atomic coordinates for each of the query sequences.

Next, the GUI will automatically open the density map and the pdb file of your query sequences, in a COOT window for your inspection (**Fig. 6)**.

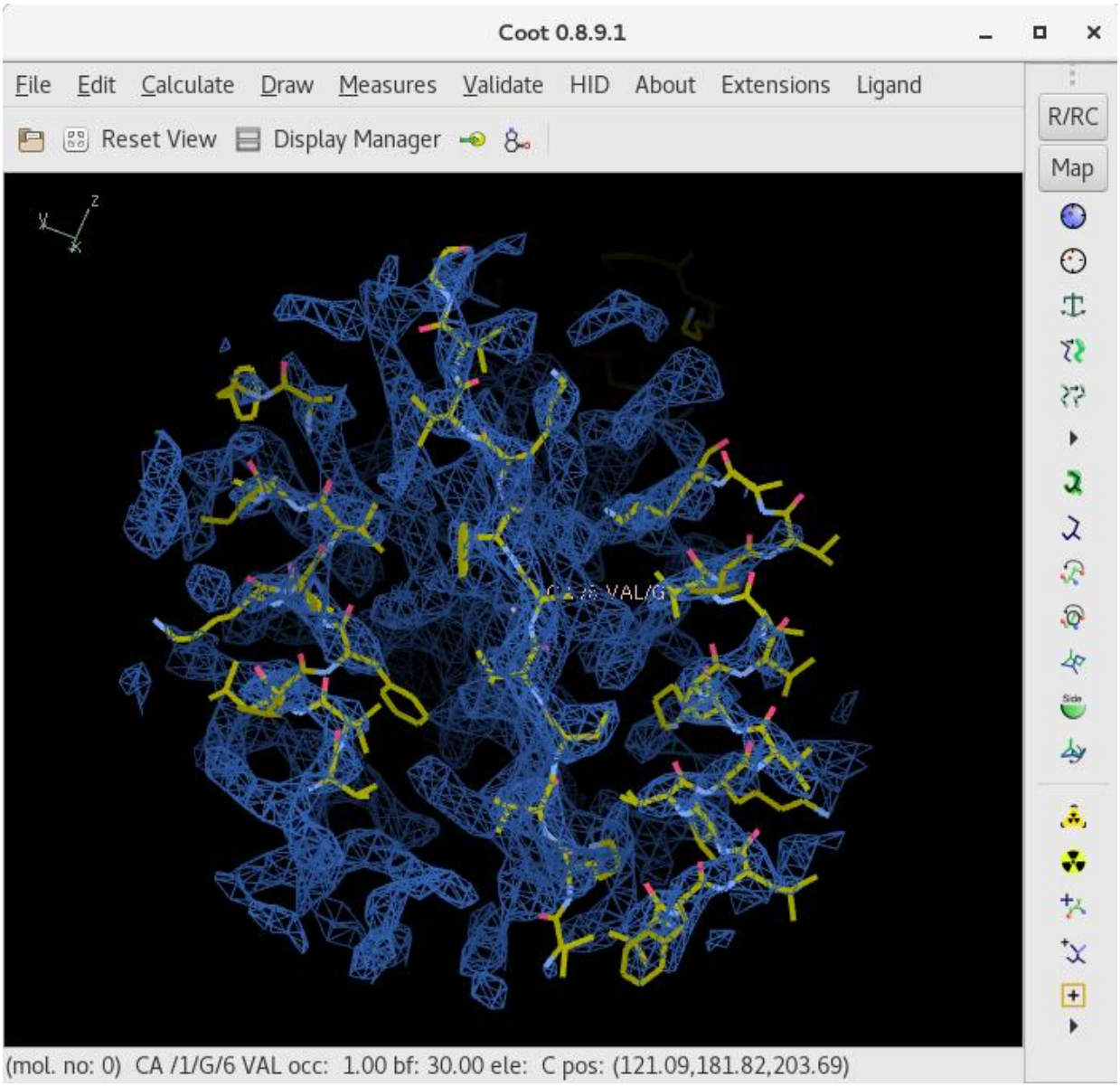

Figure 6 | COOT Pop-up Window From cryoID

Critical step: Inspect the generated queries and the corresponding segments, and make sure they fit well. You need to manually correct the obviously wrongly assigned residues (which falls to a different degenerate group) to the correct group with “mutate” tool in COOT. Be sure to mutate any residues that cannot be confidently predicted (i.e. due to broken side chain density) to the MSE residue-type with the “simple mutate” tool in COOT. CryoID search_pool will convert residues designated as MSE into X, using the PHENIX print_sequence function in the next step. Discard queries that contain excessive wrong residues or

unpredictable residues and use the best ones for the searching step. Based on our experience, a set of 2~4 accurate queries is sufficient for successful identification of the target protein.

For improved results, you can extend the queries on either end if the cryoEM density is of sufficient quality. For each query, we recommend a minimum length of 15 amino acids. You can also directly modify the query sequences in the fasta file instead of mutating residues in pdb file *via* the COOT interface, if this is easier for you. In the case of hetero-complex (if you suspect), it's a good idea that you work on one protein at a time: try to group queries that apparently from the same protein to the same query set and search the best query set you get first.

The optimized query set files for UnkwnPro.mrc are provided in the test dataset with this tutorial.

3.3 Search pool

The program search_pool translates both the query and candidate sequences into the simplified degenerate code, and performs a customized blastp search of the query sequences against each of the candidate protein sequences. To run the job with test data, use the **Browse** buttons in the GUI to locate and select your query set and candidate pool files:

Query file:	UnkwnPro_queries_optimized.pdb	*Input the latest user-updated query sequence file (.pdb or fasta file)
Protein pool:	MS_candidates.fasta	*Input your candidate protein sequences (.fasta file)
Options:		*Additional options described below

Input your modified query pdb or fasta file from get_queries as the query file. If you don't know the N/C terminal polarity of your queries, add the –r flag in the Options field (see below). For the protein pool, you can provide a standard fasta file which contains all the candidate protein sequences.

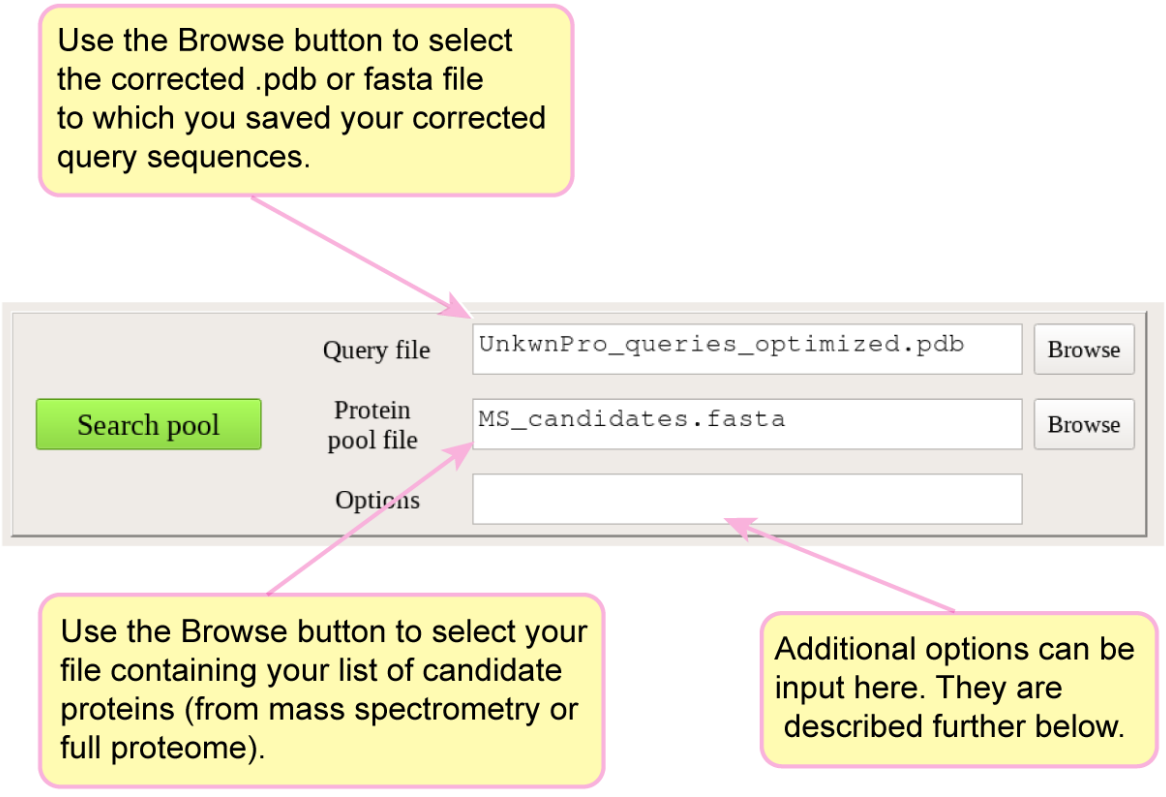

Figure 7 | The cryoID Search pool Panel

There are a number of additional options available for the search_pool subprogram:

Option	Default value	Description
-o	Input query file basename	Set output file basename
-l	1 10000	Set candidate sequence length range
-n	20	Set maximum number of candidates to keep
-e	1.0	Set cutoff E-value for Blastp search
-g	32767 32767	Set gap penalty scores for Blastp search
-r	n/a	Flag to indicate unknown query N/C terminal polarity
-v	n/a	Flag to increase output verbosity

By default, we set very large gap penalty scores such that no gaps are allowed in the alignments of query and candidate sequences. To allow gaps during blastp search, change the gap penalty scores to “15 3” (“gap open”/“gap extension” parameter in Blastp).

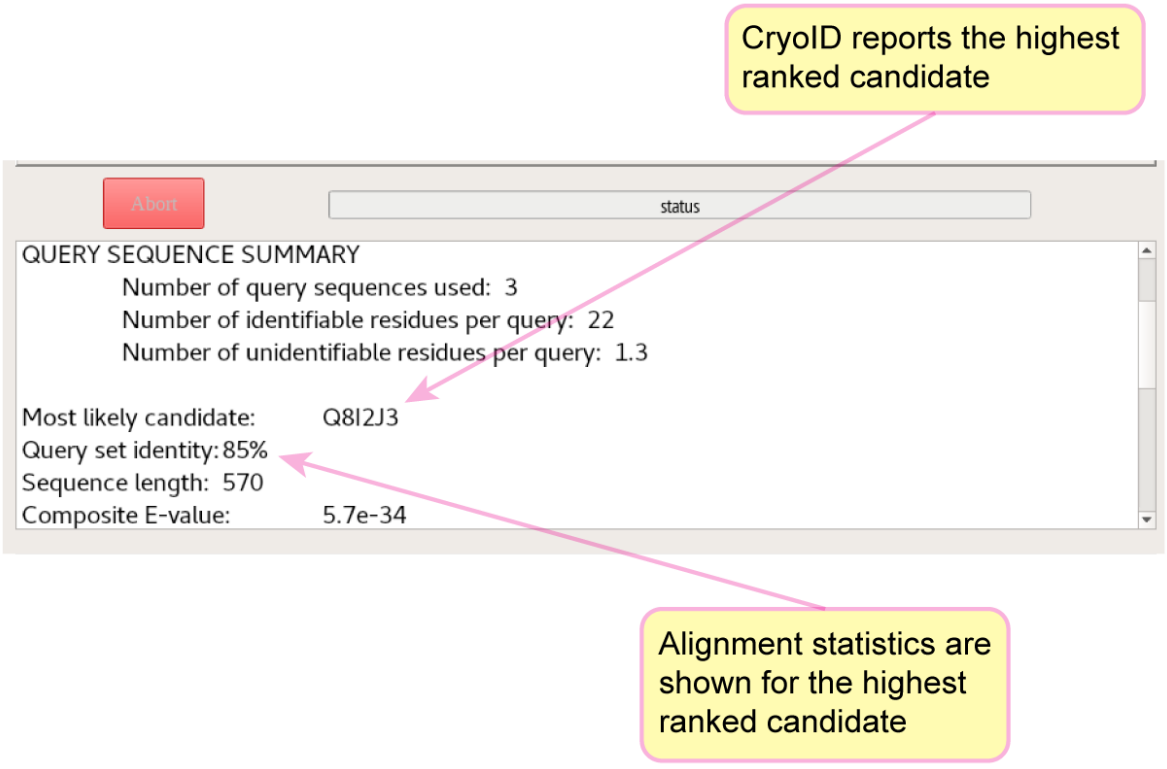

Figure 8 | The Best Candidate is Reported in the cryoID Output Window

Search_pool simplifies the candidate sequences and generates a local blastp database of simplified candidate sequences. It then analyses the query set and reports the number/length/X(spacers). A warning or error message will appear if the query sequences are too short or if there are not enough query sequences based on the simulation result. The search job should complete in a few seconds on most linux systems.

Once the job is complete, search_pool will rank the candidates by the composite E-value and report the most likely candidate, with the corresponding alignment statistics, in the output window in the cryoID GUI (**Fig. 8**).

CryoID may raise a warning message if the identity of the query set with all candidates is low, since these instances may lead to unreliable results.

CryoID will create two files. The first file, UnkwnPro_queries-optimized_summary.txt, contains the composite E-value and % identity reported from the alignment of top ranking candidate proteins against the query set. The second file, UnkwnPro_queries-optimized_alignment.txt, contains the alignment statistics of top ranking candidate proteins against the query set.

The GUI will also open a pop-up window (**Fig. 9**) displaying two graphs with the top candidates plotted against composite E-value and against % identity with the query set.

The presence of a statistically significant “gap” in composite E-values between the top candidate and all other candidates indicates high confidence identification. Two dashed lines are shown on each graph, corresponding to the composite E-values of the top candidate and the next most likely candidate. If there is a clear gap between the two dashed lines in both the E-value and % identity graphs, as seen in **Figure 9**, the highest ranked candidate has a very high likelihood of being the correct protein in your cryoEM density map. If not you may have to further optimize your queries, say to extend the length or polish the residue predictions.

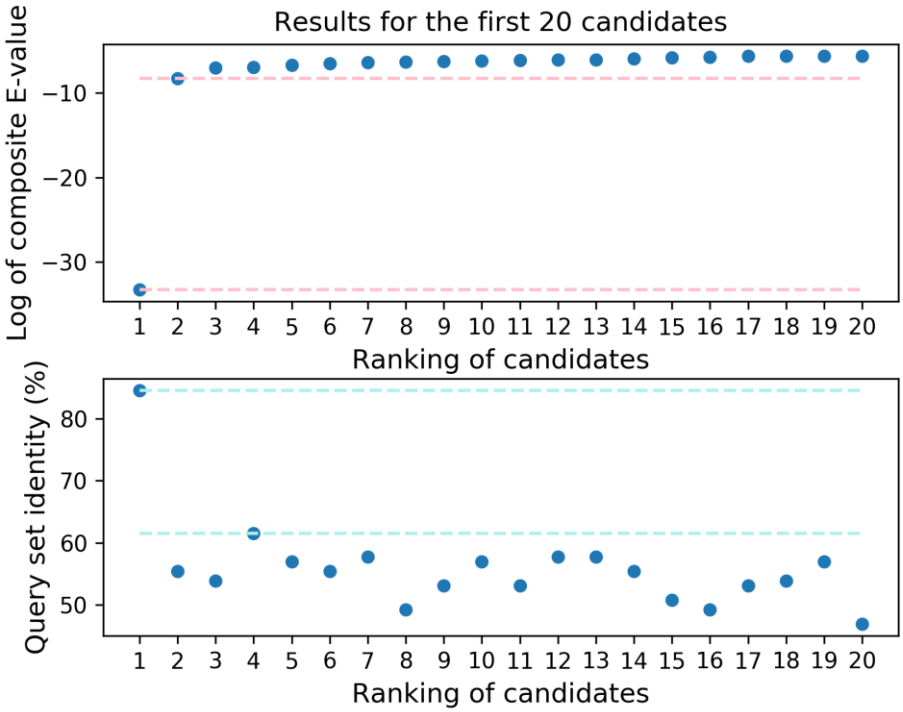

Figure 9 | The cryoID Final Candidate Ranking Pop-up Window

For the user's convenience, pre-generated results of the test data are also provided for the test dataset. We identified the cryoEM density map provided in the test dataset to be Q8I2J3 (M18 aspartyl aminopeptidase) from *P. falciparum*.