Usage Notes

https://gitlab.ic.unicamp.br/ra135368/np3_ligand/-/tree/master/np3_blob_label

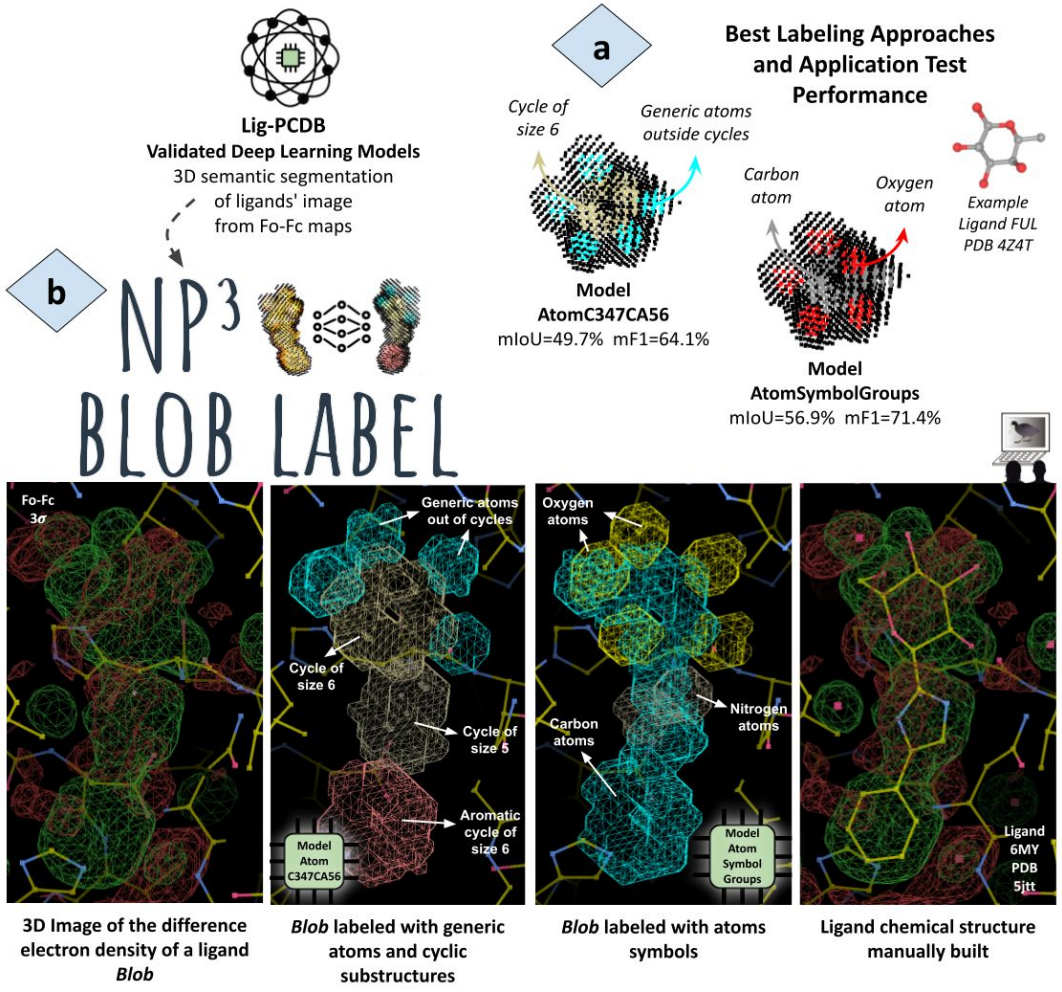
Cristina Freitas Bazzano

Summary

The NP³ Blob Label is a deep learning semi-automatic solution to auxiliate in ligand building tasks.

It is capable of finding new ligands sites in difference electron density maps, called blobs, and for each found blob the application will use the validated DL models from Lig-PCDB (a) to predict chemical substructures that fill and explain each part of the blob.

The predictions serve as an initial proposal to help in the complete manual reconstructions of the ligand chemical structure, as illustrated in (b).



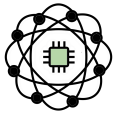
How

The deep learning image segmentation models were trained and validated with a dataset of **78902** ligands images from difference electron density of **26976** PDB entries and **11925** unique ligands in a resolution range between **1.5 Å** and **2.2 Å**. Data outside this resolution range may still be used in the NP³ Blob Label application, but the models accuracy (presented next) may not be reliable.

The application will create an image of the blob from its difference electron density map and label this image. The blob image will be slight affected by the σ contour level used in the search and the labels depends on the model used for the predictions. A schema of the application pipeline is illustrated next.

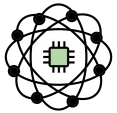
The following models were validated by Lig-PCDB and gave the best result. They are available in the NP³ Blob Label repository (*models* folder):

Vocabulary
Background
Atom
C5
CA5
C6
CA6
C3
C4
C7



Model
AtomC347CA56

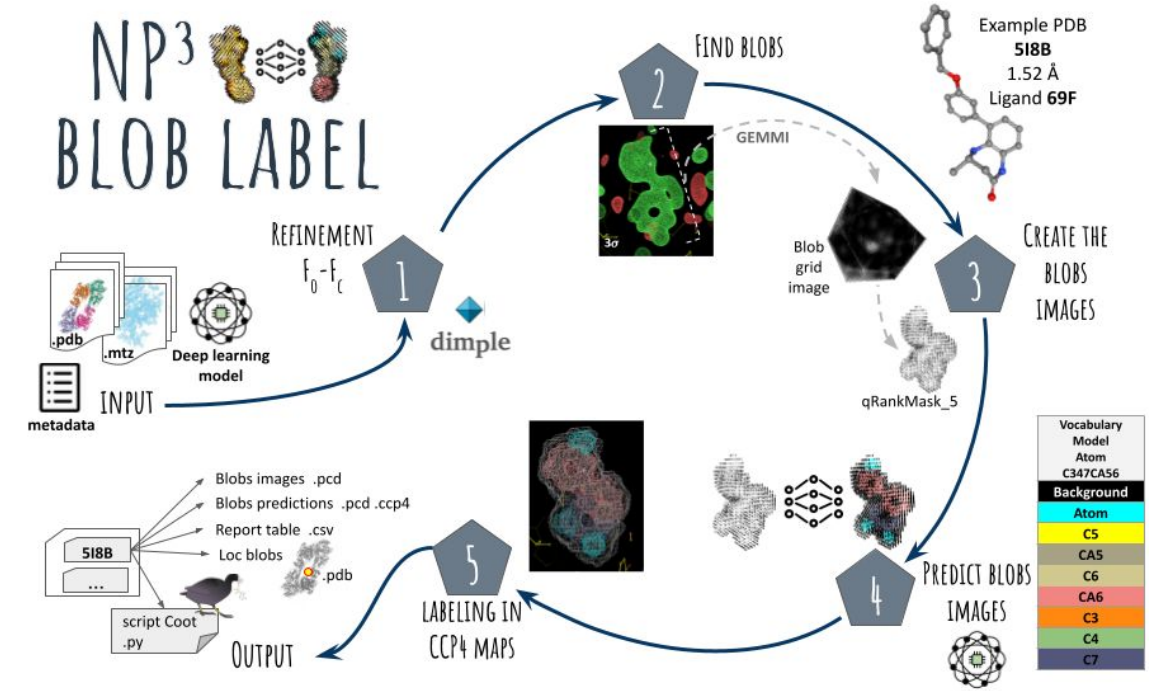
Predicts generic atoms out of cycles, cycles with sizes from 3 to 7 and aromatic cycles with sizes 5 and 6



Model
AtomSymbolGroups

Predicts the atoms symbols with groupings. The halo group with the halogen atoms (Cl, Br, I, F) and the PSe group with the minority atoms (P, S, Se)

Vocabulary
Background
C
O
N
PSe
Halo



At the end of the workflow the user may easily visualize the result of each entry with the Python script created for Coot, which automatically opens the inputs (.mtz and .pdb) of an entry along with the synthetic electron density maps of each segmented class of the found blobs. The user may **browse the found blobs** and visualize their predictions using Coot's atom navigation tool (using the .pdb file of the protein entry with dummy atoms centered at the position of each found blob). The application result also contains a **report table** with all found blobs, their information (intensity, volume, score and position) and their predicted classes by size (number of labeled points in each class), which may help the user summarize the findings and prioritize further analysis.

The **NP³ Blob Label** is fully automated in python scripts and its functionality was wrapped up in a command line interface with few inputs. The code is freely available in the github repository of the application<link>.

The NP³ Blob Label **pipeline workflow:**

(1) Refine the entries with Dimple;;

(2) Search for blobs in the entire calculated Fo-Fc map or in given specific positions

- Uses a σ contour level for the search and other parameters;

(3) Create the final images of the blobs in 3D point cloud (qRankMask_5);

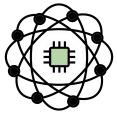
- Uses the quantile rank scale to transform the density values of the blob grid

(4) Label these images with a validated DL model prediction; and

(5) Convert the prediction result to synthetic electron density maps in CCP4 format, with each segmented class in a different map.

Models Accuracy

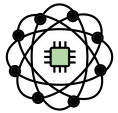
Vocabulary
Background
Atom
C5
CA5
C6
CA6
C3
C4
C7



Model AtomC347CA56

Three different values for the σ contour level parameter were used to evaluate the impact of this parameter on the accuracy of the application. Contours equal to 2, 2.5 and 3σ were used in the blobs search. For the other parameters, their default values were maintained (minimum volume set to 24 $Å^3$ and minimum score and peak intensity set to zero).

The results of these tests are presented below with their accuracies in terms of mIoU, mF1, Precision and Recall and the percentage of entries that were correctly imaged.



Model AtomSymbolGroups

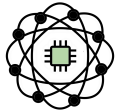
Vocabulary
Background
C
O
N
PSe
Halo

Results from the hold-out CV against 3036 ligands from the k=1 test subset of the stratified training dataset from Lig-PCDB for model AtomSymbolGroups and against 3035 from k=13 test subset for model AtomC347CA56. Best values by model are highlighted.

DL Model	Sigma Contour Level	Number of Blobs Imaged	Test mIoU	F1 score	Precision	Recall
AtomC347CA56	2	**2674 (88%)**	47.4	61.8	61	64.1
AtomC347CA56	2.5	2523 (83%)	48	62.2	60.7	65.2
AtomC347CA56	**3**	2362 (78%)	**49.7**	**64.1**	**61.3**	**69.4**
AtomSymbolGroups	2	**2687 (89%)**	54.4	69.2	67.3	72.9
AtomSymbolGroups	2.5	2540 (84%)	56	70.6	68.2	74.7
AtomSymbolGroups	**3**	2367 (78%)	**56.9**	**71.4**	**68.4**	**76.2**

Models Accuracy

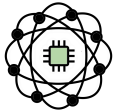
Vocabulary
Background
Atom
C5
CA5
C6
CA6
C3
C4
C7



Model AtomC347CA56

The diagonal of the confusion matrix below shows the accuracy by class in terms of Intersection over Union (IoU) from 0 to 100%.
And the mean IoU (mIoU) is the accuracy for the entire model. The best σ contour level was used.

The confusion matrix have the expected Class in the row and the predicted Class in the column.
They were normalized by row.



Model AtomSymbolGroups

Vocabulary
Background
C
O
N
PSe
Halo

Confusion Matrix

3 σ	Background	Atom	C5	CA5	C6	CA6	C3	C4	C7
Background	**86.5**	3.8	0.3	0.1	2.0	0.6	0.0	0.0	0.0
Atom	21.5	**54.3**	0.4	0.0	1.1	0.6	0.0	0.0	0.0
C5	13.7	2.8	**59.9**	1.2	2.0	1.3	0.0	0.0	0.0
CA5	14.0	2.5	2.7	**64.7**	0.8	6.1	0.0	0.0	0.0
C6	13.7	2.8	0.6	0.1	**36.1**	3.7	0.0	0.0	0.0
CA6	15.5	2.5	0.4	0.3	3.2	**60.7**	0.0	0.0	0.0
C3	20.3	22.1	0.0	0.0	1.1	0.3	**22.9**	1.5	0.0
C4	23.3	10.0	13.8	5.4	0.2	1.1	0.0	**30.3**	0.0
C7	19.7	4.3	0.2	1.5	26.7	0.3	0.0	0.0	**31.6**

mIoU = 49.7%

Expected Class

Predicted Class

3 σ	Background	C	O	N	PSe	Halo
Background	**86.7**	4.3	2.0	0.3	0.1	0.0
C	17.3	**54.8**	1.4	0.7	0.1	0.0
O	19.8	6.3	**48.3**	0.8	0.5	0.1
N	16.6	16.4	3.5	**49.4**	0.1	0.0
PSe	11.3	3.7	3.9	0.1	**61.8**	0.2
Halo	23.5	11.5	10.3	0.6	0.6	**40.2**

mIoU = 56.9%

NP^3 BLOB LABEL

Executing the application:

- Files organization
- Input parameters setup and format
- Example of use
- Output organization and visualization

github repo

NP³ BLOB LABEL

Files organization

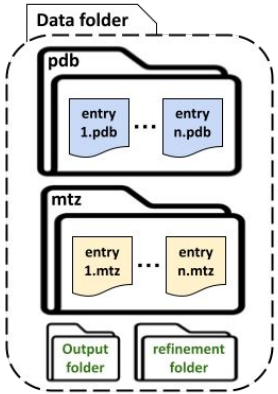
Two possibilities depending on the entries refinement status

A

Refine at least one of the entries*

Inputs:

1) **Data folder path**
   - **mtz** and **pdb** subfolders
     - One subfolder by entry
   - ***Output path is here***
   - ***Refinement folder***
2) **Entries list table path**
3) **Model checkpoint path**
4) Blob search parms



* This file organization may also be used with only refined entries. In this case, the refinement should be disabled in the entries list table and the pdb and mtz files will be copied to the refinement folder

Entries list table

+

Model .ckpt

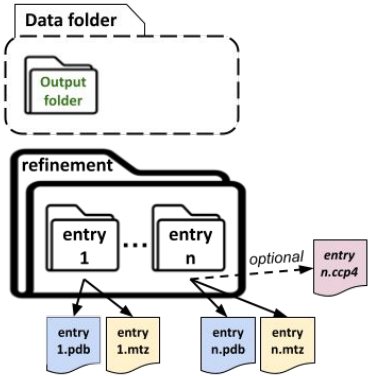
B

Use previous refinement result

Do not refine the entries (skip step 1)

Inputs:

1) **Data folder path**
   - ***Output path is here***
2) **Entries list table path**
3) **Model checkpoint path**
4) Blob search parms
5) **Refinement folder path**
   - One subfolder by entry



Input Parameters

***Mandatory parameters** in bold*

- ***data_folder*:** this is the output path
  - A folder named *np3_ligand_<DATE>* will be created to store the NP³ Blob Label results
  - If *refinement_path* is not informed, it should contain the subfolders *pdb* and *mtz*; and a folder named *refinement_<DATE>* will be created here.
- ***entries_list_path*:** the path to the CSV metadata table defining the datasets that will be processed (it must be comma separated)
- ***model_ckpt_path*:** the path to a model checkpoint file (provided by the application) - to segment the blob image
  - Model AtomC347CA56
  - Model AtomSymbolGroups
- *refinement_path*: path to a previous refinement result
- *search_blobs*: 'all' will run the find blobs (step 2) to search for all blobs that fulfill the parameters criteria or 'list' to search only for the blobs listed in specific position in the *entries_list_path* table (default: all)
- *output_name:* Used to name the output directory in following format: "np3_ligand_<output_name>_<DATE>".

The format of the entries list table and the Blob search (step 2) parameters will be presented next

Entries List Table - Metadata Format

- Depends on the Find blobs (Step 2) mode defined with the parameter *search_blobs*:
  - 'all' -> find all blobs
  - 'list' -> Search for blobs in specific positions
- In both cases it will return the blobs that fulfill the search parameters (detailed next)

Find all blobs

- *entryID* should start with a letter, have unique values and no space or special characters
  - These values must match the names of the entries input .pdb and .mtz files

entryID	**refinement**	**noHetatm**
entry**1**	*1 or 0*	*1 or 0*
…	0 *(disabled)*	0
entry**n**	1 *(enabled)*	1

Search for blobs in specific positions

- *entryID* and *blobID* should have unique values when combined
- *blobID* should start with a letter and have no space or special characters
- x, y and z with the position to search for blobs

entryID	**refinement**	**noHetatm**	**blobID**	**x**	**y**	**z**
entry**1**	*1 or 0*	*1 or 0*	blob_1_0	1.1	2.2	3.3
…	0 *(disabled)*	0	…	…	…	…
entry**n**	1 *(enabled)*	1	blob_n_0	0	1	3

Blobs Search Parameters

- *sigma_cutoff*: A numeric defining the sigma cutoff to be used to search for blobs in the difference electron density map. Values greater or equal than 2σ are **recommended**. (default: 3.0)
  - Values closer to 2σ may retrieve low quality blobs (which could have a fragmented density)
  - Values closer to 3σ may retrieve only high quality blobs.
  - Values smaller than 1.5σ are **not recommended**, because they could create a blob image with too much noise, very big and slow to process.
  - This value will directly affect the blobs image creation, and thus ,the prediction result
- *blob_min_volume*: A numeric defining the minimum volume that a blob must have to be considered. Only used when *search_blobs* is 'all'. Default to 24 A^3, what is equivalent to the volume of a water. Smaller values will allow retrieving smaller blobs, the opposite is also true. (default: 24.0)
- *blob_min_score*: A numeric defining the minimum score that a blob must have to be considered. Only used when *search_blobs* is 'all'. The score of a blob is equal to the sum of the difference electron density values of its points. (default: 0)
- *blob_min_peak*: A numeric defining the minimum intensity that the peak (most intense point) of a blob must have for the blob to be considered. Only used when *search_blobs* is 'all'. (default: 0)

Examples from PDB - Input

entries_list_top_down.csv

entryID	refinement	noHetatm
4bam	1	1
4xhe	1	1
5f07	1	1
5i8b	1	1
5jtt	1	1
5m8g	1	1
5ybo	1	1
6dir	1	1
4rvn	1	1

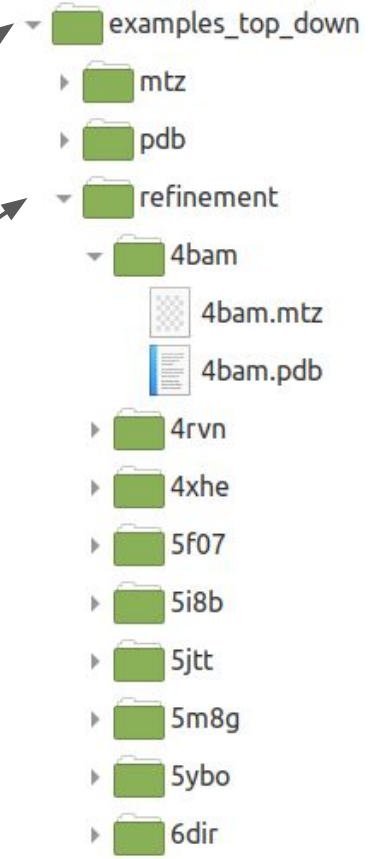
Input:

data_folder

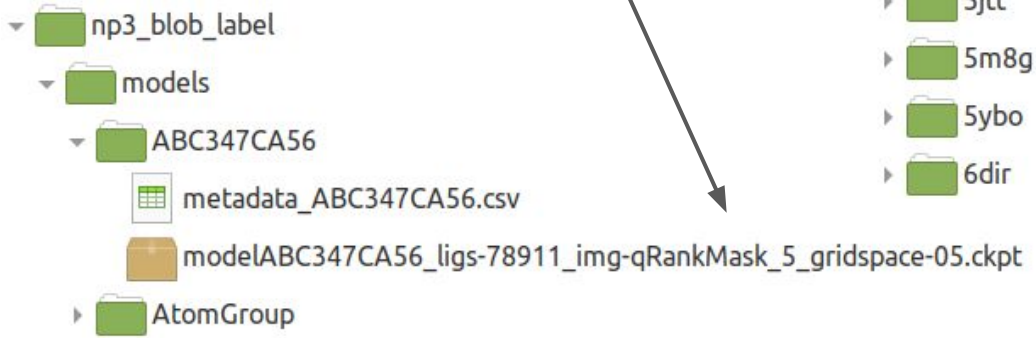
entries_list_path

refinement_path

model_ckpt_path



NP³ Blob Label repository and provided example:

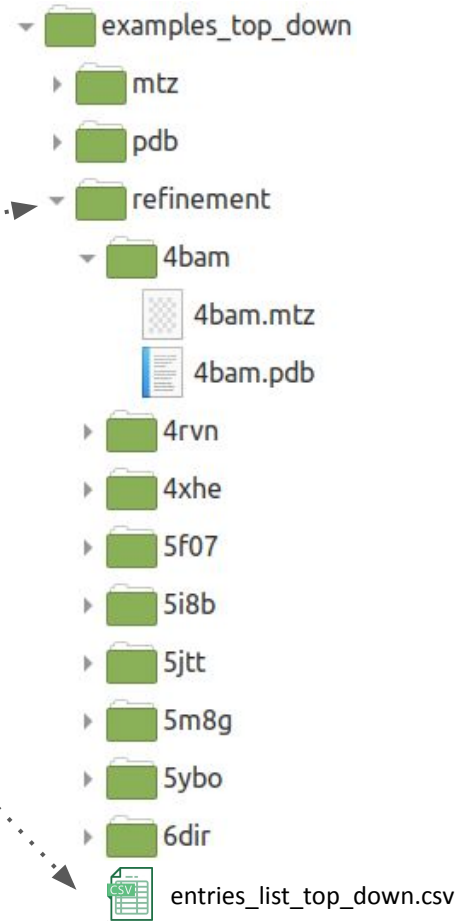


Examples from PDB - Run

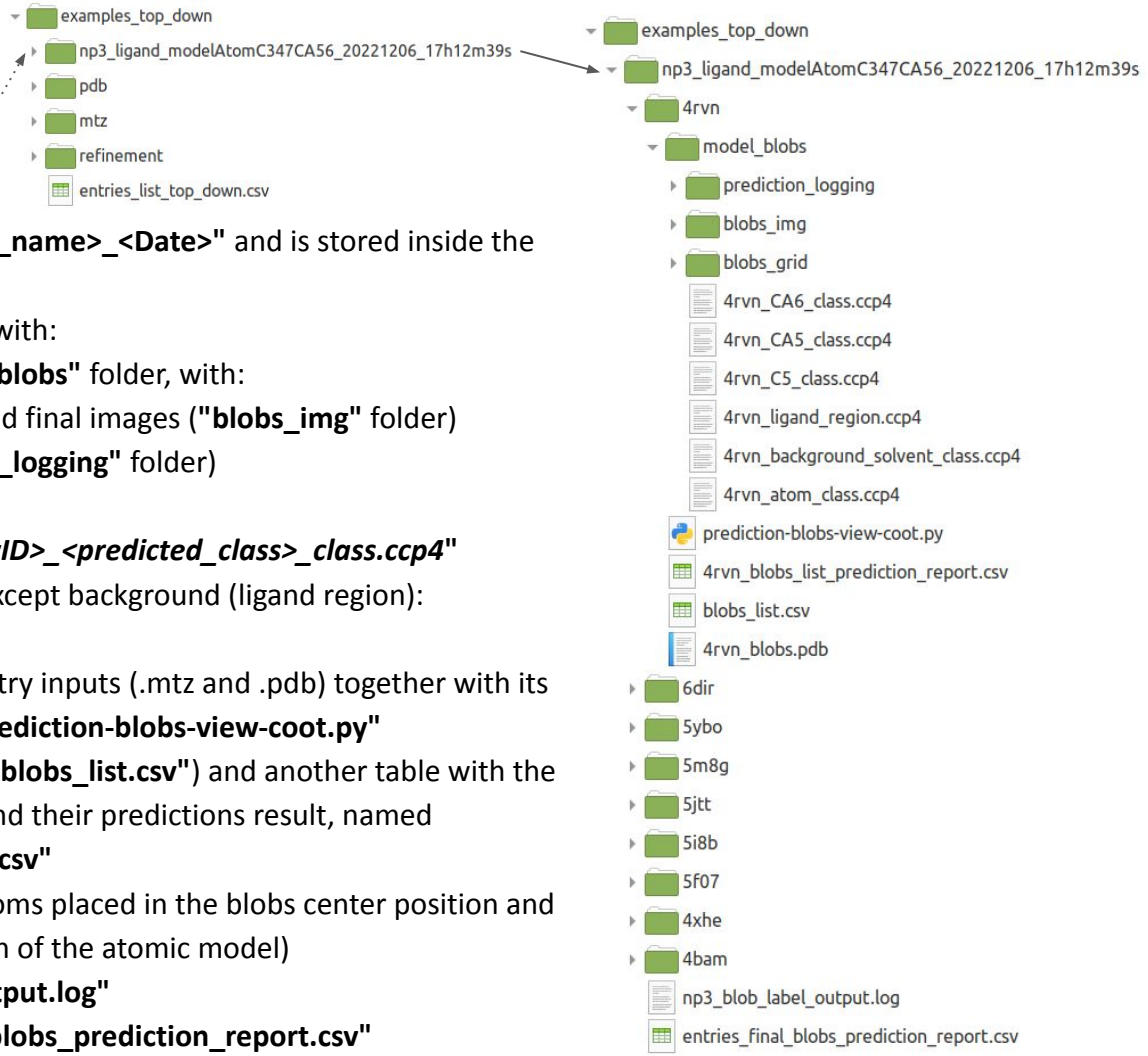
- With the requirements installed, open a terminal window inside the NP³ Blob Label repository and run:

```
(np3_blob_label) $ python np3_blob_label.py --data_folder
examples_top_down/ --refinement_path examples_top_down/refinement/
--entries_list_path examples_top_down/entries_list_top_down.csv
--model_ckpt_path
models/AtomC347CA56/modelAtomC347CA56_ligs-78911_img-qRankMask_5_gridsp
ace-05_k1.ckpt --output_name modelAtomC347CA56
```

NP³ Blob Label repository:



Examples from PDB - Output



- The **output** folder is named **"np3_ligand_<output_name>_<Date>"** and is stored inside the *data_folder* path. It contains:
  - One subfolder by entry, named **"<entryID>"**, with:
    - The blobs' modeling result in the **"model_blobs"** folder, with:
      - The blobs' grid (**"blobs_grid"** folder) and final images (**"blobs_img"** folder)
      - The prediction logging file (**"prediciton_logging"** folder)
      - The predicted classes in CCP4 maps:
        - One map by predicted class: **"*<entryID>_<predicted_class>_class.ccp4*"**
        - One map with all predicted points except background (ligand region): **"*<entryID>_ligand_region.ccp4"***
    - A Coot script to automatically open the entry inputs (.mtz and .pdb) together with its outputs (CCP4 maps with predictions): **"prediction-blobs-view-coot.py"**
    - A table listing the blobs that were found (**"blobs_list.csv"**) and another table with the respective entry report, listing the blobs and their predictions result, named **"<entryID>_blobs_list_prediction_report.csv"**
    - A **"<entryID>_blobs.pdb"** file with fake atoms placed in the blobs center position and in a fake chain (will always be the last chain of the atomic model)
  - One Logging file named **"np3_blob_label_output.log"**
  - One global report file named **"entries_final_blobs_prediction_report.csv"**

Examples from PDB - Output Report Table

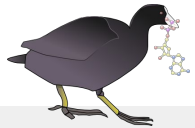
At the end of the process, the user may inspect all the found blobs of all entries in the final report table:

- **np3_ligand_<output_name>_<Date>/entries_final_blobs_prediction_report.csv**

The final report table contains one found blob by row with their information by column, such as the blobs' intensity, volume, score and position, and their predicted classes by size (number of labeled points in each class). This table may help the user summarize the findings and prioritize further analysis.

The description of the columns of this table are presented in the file 'np3_blob_label/docs/report_table_columns_description.csv'

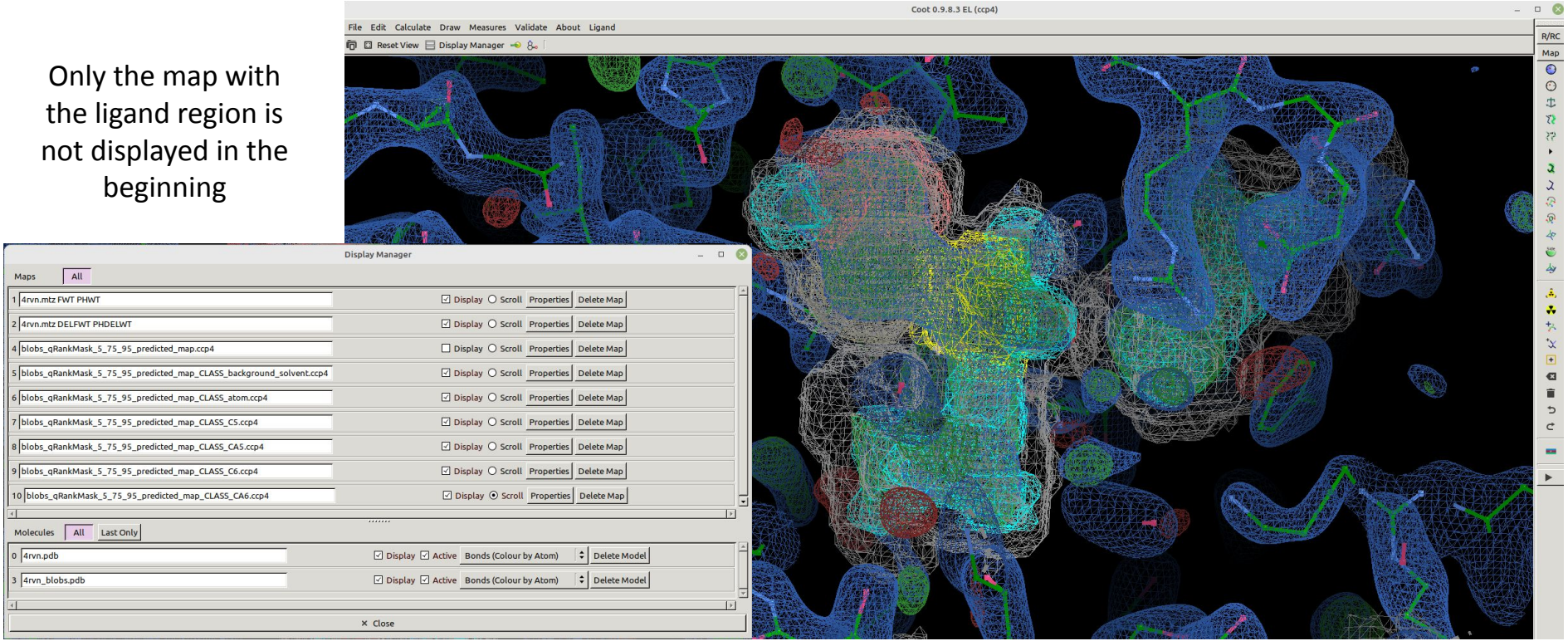
Examples from PDB - Output Visualization in Coot



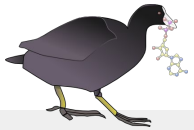
- To execute the NP³ Blob label coot script for entry **4rvn**, open a terminal window and run:

```
$ coot --script examples_top_down/np3_ligand_20221117_13h05m31s/4rvn/prediction-blobs-view-coot.py
```

Only the map with the ligand region is not displayed in the beginning



Examples from PDB - Output Visualization in Coot

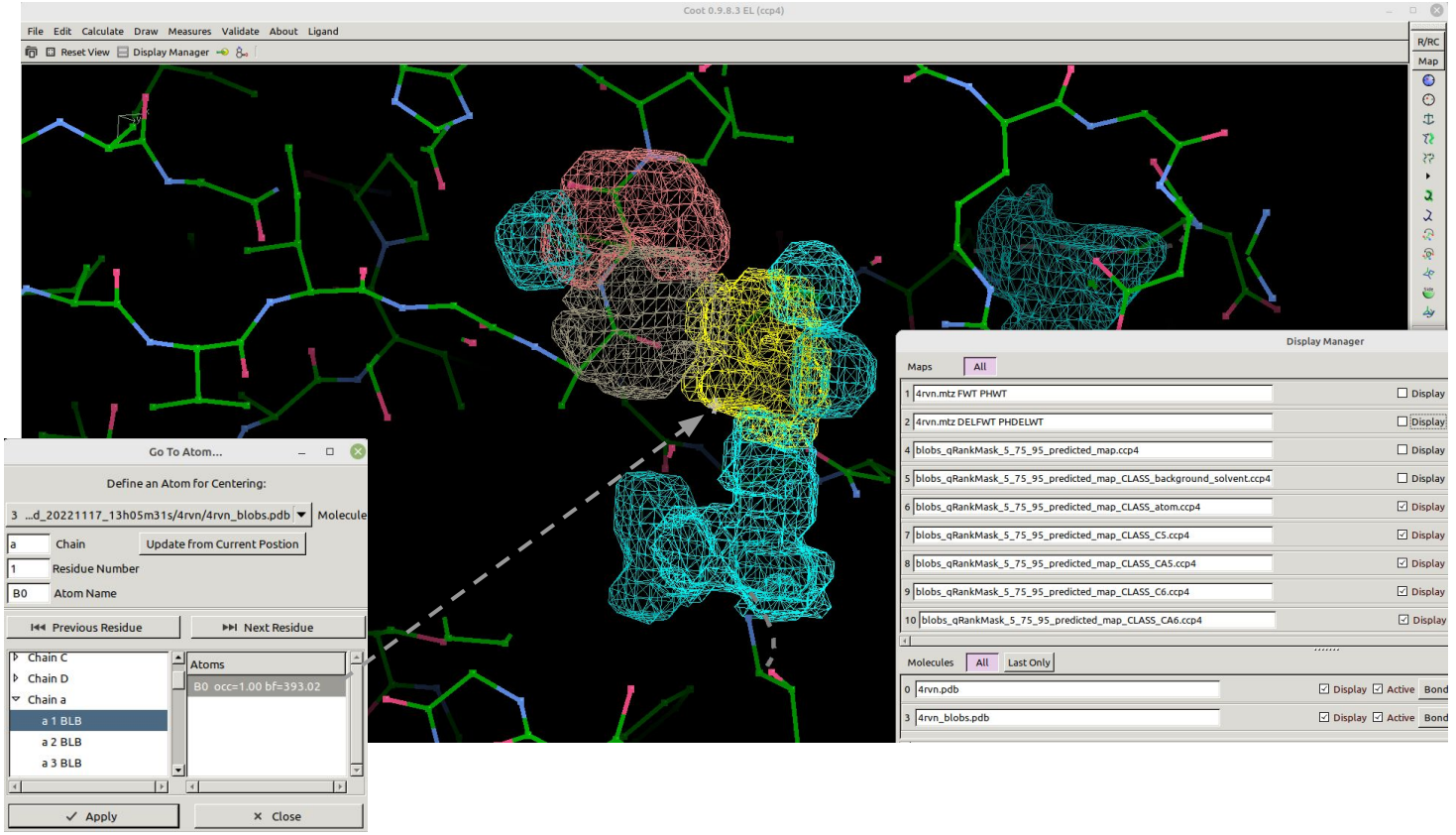


```
$ coot --script examples_top_down/np3_ligand_20221117_13h05m31s/4rvn/prediction-blobs-view-coot.py
```

For a better visualization of the result the user may set off the display of the experimental electron density maps and the background class map

The background class may help visualize the format of the blobs' image

The Coot "Go To Atom…" tool may help navigate through the result, by centering in each found blob



NP³ Blob Label Commands - Examples

- Run NP³ Blob Label using a previous refinement result to search for all blobs in the given entries:

```
$ python np3_blob_label.py  --data_folder examples_top_down/  --entries_list_path examples_top_down/entries_list_top_down.csv
--refinement_path examples_top_down/refinement/  --model_ckpt_path
models/AtomC347CA56/modelAtomC347CA56_ligs-78911_img-qRankMask_5_gridspace-05.ckpt
```

- Run NP³ Blob Label to refine the entries specified in the entries list table and to search for blobs in specific positions:

```
$ python np3_blob_label.py  --data_folder examples_top_down/  --entries_list_path
examples_top_down/entries_list_top_down_search_positions.csv   --model_ckpt_path
models/AtomC347CA56/modelAtomC347CA56_ligs-78911_img-qRankMask_5_gridspace-05.ckpt    --search_blobs list
```

- Run NP³ Blob Label using a previous refinement result to search for all blobs in the given entries and to apply the model to segment the blobs' image in atoms' groups classes:

```
$ python np3_blob_label.py --data_folder examples_top_down/ --entries_list_path examples_top_down/entries_list_top_down.csv
--refinement_path examples_top_down/refinement/   --model_ckpt_path
models/AtomSymbolGroups/modelAtomSymbolGroups_ligs-78911_img-qRankMask_5_gridspace-05.ckpt
```

- Check the list of parameters and mandatory parameters for NP³ Blob Label:

```
$ python np3_blob_label.py
```

- Check the list of parameters with their full description, default values and expected values for NP³ Blob Label:

```
$ python np3_blob_label.py  --help
```

NP³ Blob Label Commands - Examples

- Run NP³ Blob Label using a previous refinement result to search for all blobs in the given entries with sigma cutoff equal to 2:

```
$ python np3_blob_label.py --data_folder examples_top_down/ --entries_list_path examples_top_down/entries_list_top_down.csv
--refinement_path examples_top_down/refinement/ --model_ckpt_path
models/AtomC347CA56/modelAtomC347CA56_ligs-78911_img-qRankMask_5_gridspace-05.ckpt    --sigma_cutoff 2
```

- Run NP³ Blob Label using a previous refinement result to search for all blobs in the given entries with sigma cutoff equal to 2 σ and minimum blob volume equal to 50 Å³:

```
$ python np3_blob_label.py --data_folder examples_top_down/ --entries_list_path examples_top_down/entries_list_top_down.csv
--refinement_path examples_top_down/refinement/ --model_ckpt_path
models/AtomC347CA56/modelAtomC347CA56_ligs-78911_img-qRankMask_5_gridspace-05.ckpt    --sigma_cutoff 2 --blob_min_volume 50
```

- Run NP³ Blob Label using a previous refinement result to search for all blobs in the given entries with sigma cutoff equal to 2.5 σ and number of CPU cores for parallelization equal to 4:

```
$ python np3_blob_label.py --data_folder examples_top_down/ --entries_list_path examples_top_down/entries_list_top_down.csv
--refinement_path examples_top_down/refinement/ --model_ckpt_path
models/AtomC347CA56/modelAtomC347CA56_ligs-78911_img-qRankMask_5_gridspace-05.ckpt    --sigma_cutoff 2.5 --parallel_cores 4
```

Get in contact with the NP³ team
for more information about the NP³ Blob Label application

- Open an issue/discussion in the github repository:
  - github/issues
- Send an e-mail with your question or comment:
  - *daniela.trivella@lnbio.cnpem.br*

Thanks!