Refining tools for extracting and analyzing macromolecular tunnel geometry

May 17, 2019

Khanh Dao Duc

Note : This document 1) describes current methods and tools used to extract tunnel geometry from atomic structures and 2) proposes how to improve them, in particular in the context of our current research on ribosome. It is written for a bioinformatics project involving comp bio graduate students. It should also allow to get hands on structures and analyze them without former experience on the subject.

Introduction: Input data, visualization tools and tunnel algorithms for biological atomic structures

I shall provide here a quick summary and review of different useful tools and softwares.

Input data and visualization tools for biological atomic structures

Atomic structures Most biological macromolecular structures are compiled in the protein data bank website[1]. There is a codename for each referenced structure. For example, reference 4UG0 gives the structure of the human 80S ribosome[2]. The page of the structure contains details about the authors, associated references and annotations (sequences, codenames for the different chains etc.), more details on the quality and resolution of the structure (I am actually still learning about how to analyze it). The page also contains a download button to get the structure in format called PDB

[1] https://www.rcsb.org/

[2] https://www.rcsb.org/structure/4ug0

and/or mmCIF, which basically contains all the positional and structural information of the structure.

Visualization There are multiple softwares to visualize and analyze PDB files. The one I mainly use is *Pymol*[3]. It is free of charge for students and easy to install. One advantage of *Pymol* is that it can run python scripts, which are useful to export information for downstream analysis. For our research, we have also used Chimera[4], developed at UCSF.

Tunnel search algorithms and softwares

Various algorithms and tools have been developed to extract channel geometries from a PDB file, relying on different approaches[5]. When I first studied the ribosome tunnel and how the radius and electrostatic potential vary along it (3), I did a quick survey of the different softwares available. Some softwares are quite outdated, not very user-friendly or were not much appropriate for the analysis I wanted to carry out. For example, the software called *Chunnel* (4) uses old fortran libraries that are impossible to run on recent machines. One called *3V* (5), which uses rolling spheres to approximate the inner surface of the ribosome only works for a single species[6]. A software called *HOLLOW* (**?**) interestingly uses the idea of filling the tunnel or any cavity with spheres. Besides being numerically heavier than the one I'll describe below, I found it hard to use because you have to specify an axis and a cylinder that the algorithm will fill with small spheres, which is trickier than just specifying a starting point. The output is also not really appropriate for comparisons and will mostly be good for visualizations and compute global features like the volume only.

The method I finally came using is called *MOLE* (6). It is in my opinion the most user-friendly and it is well maintained (a new version just came out), with an online server[7](7) and user interface that does not require any

[3] https://pymol.org/2/

[4] https://www.cgl.ucsf.edu/chimera/

[5] Introduction of Sehnal *et al.* (1) gives a summary of the different classes of algorithms. For a more general recent review on the visual analysis of cavities, see for example Krone *et al.*(2).

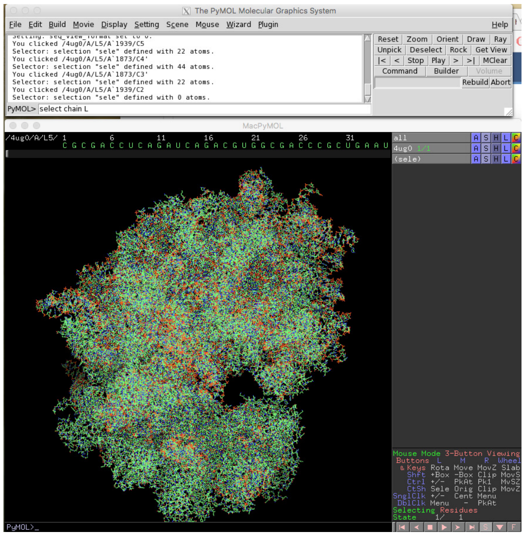
[6] See http://3vee.molmovdb.org/tunnelExtract.php. The reason invoked is that each ribosome structure uses its own coordinate system

[7] https://mole.upol.cz/

installation on one's computer. It is also quite fast[8]. Its developers have also recently introduced an online database to register channels and tunnels structures(8). Briefly, their method uses Voronoi diagrams to approximate the surface of the macromolecule, and finds optimal paths joining points at the surface[9]. The tunnel is built by putting along the path centerline spheres of maximal radius that can be contained by the structure (determined by its atoms positions and the corresponding van der Waals radius). Therefore, the tunnel is encoded by a set of 3D coordinates describing the tunnel centerline, with for each point of the centerline a radius value.

Current pipeline

I will give here a tutorial that details how to extract the exit tunnel coordinates of a ribosome. The only thing needed as a preliminary is having *Pymol* installed.



The PyMOL Molecular Graphics System
File Edit Build Movie Display Setting Scene Mouse Wizard Plugin Help
Reset Zoom Orient Draw Ray
Unpick Deselect Rock Get View
|< < Stop Play > >| MClear
Command Builder Volume
Rebuild Abort
MacPyMOL
all
4ug0 1/1
(sele)
A S H L C

```
Setting: seq_view_format set to 0.
You clicked /4ug0/A/L5/A`1939/C5
Selector: selection "sele" defined with 22 atoms.
You clicked /4ug0/A/L5/A`1873/C4'
Selector: selection "sele" defined with 44 atoms.
You clicked /4ug0/A/L5/A`1873/C3'
Selector: selection "sele" defined with 22 atoms.
You clicked /4ug0/A/L5/A`1939/C2
Selector: selection "sele" defined with 0 atoms.
PyMOL> select chain L
```

```
/4ug0/A/L5/ 1        6         11        16        21        26        31
            C G C G A C C U C A G A U C A G A C G U G G C G A C C C G C U G A A U
```

```
Mouse Mode 3-Button Viewing
 Buttons  L    M    R  Wheel
  & Keys Rota Move MovZ Slab
    Shft +Box -Box Clip MovS
    Ctrl +/-  PkAt Pk1  MvSZ
    CtSh Sele Orig Clip MovZ
SnglClk +/-  Cent Menu
 DblClk Menu   -  PkAt
Selecting Residues
State    1/   1
PyMOL>_
```

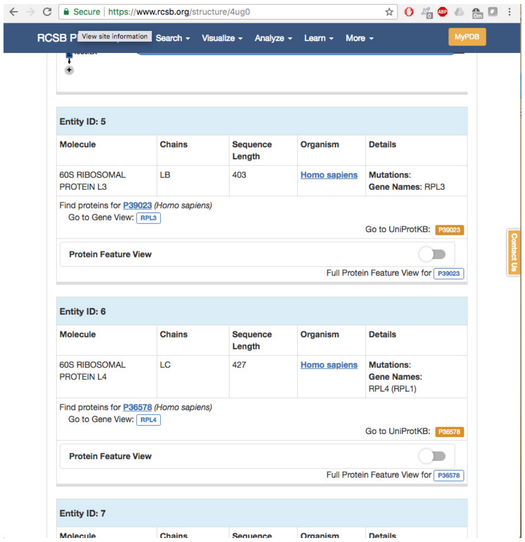
Figure 1: The default 4UG0 structure visualized with *Pymol*.

[8] It takes a few minutes to get the structure

[9] There is a software called *CAVER* (9) which seems to use similar method and quite recent. I haven't used it but *MOLE* seems to give similar or better results (6).

1. Pre-processing the data

We first download a ribosome structure, e.g. 4UG0, which can be found here. The structure is quite large ($\sim 10^5$ atoms), with a large part (like the small subunit for example) that is not useful to get the tunnel structure. To ease the computational cost, we first need to edit the structure and remove atoms that are far from the tunnel. We also need to find a starting point for the tunnel search algorithm. To do so, we make use of the fact that some parts of the tunnel, namely the “constriction site" and the “polypeptide transferase center" (PTC) are very well known and conserved[10]. First, open the structure with *Pymol* (see Fig. 1). You can choose to display the sequence (go to Display, sequence; going to sequence mode allows to choose between displaying chains, amino acids, atoms...).



Entity ID: 5

Molecule	Chains	Sequence Length	Organism	Details
60S RIBOSOMAL PROTEIN L3	LB	403	Homo sapiens	**Mutations:** **Gene Names:** RPL3

Find proteins for P39023 *(Homo sapiens)*
Go to Gene View: RPL3
Go to UniProtKB: P39023

Protein Feature View

Full Protein Feature View for P39023

Entity ID: 6

Molecule	Chains	Sequence Length	Organism	Details
60S RIBOSOMAL PROTEIN L4	LC	427	Homo sapiens	**Mutations:** **Gene Names:** RPL4 (RPL1)

Find proteins for P36578 *(Homo sapiens)*
Go to Gene View: RPL4
Go to UniProtKB: P36578

Protein Feature View

Full Protein Feature View for P36578

Entity ID: 7

Molecule	Chains	Sequence	Organism	Details

Figure 2: Reference to ribosomal protein L4 in the pdb page.

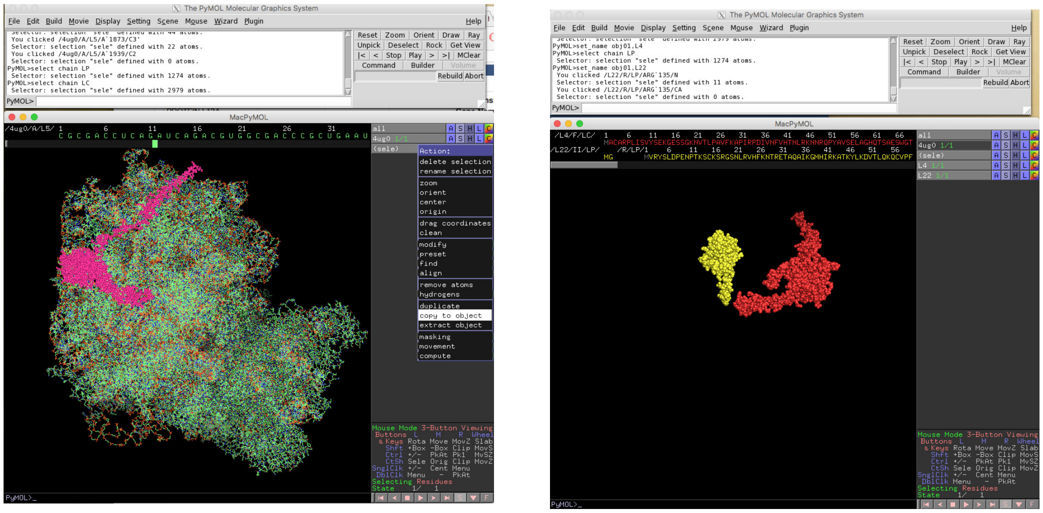
Our first goal is to locate the constriction site. The constriction site, approximated located in the middle of the tunnel, is defined by the region where two specific ribosomal proteins called uL4 and uL22 get close. In the current nomenclature system (11), u stands for universal, while other prefixes a,b and e are used to designate proteins which are respectively specific

[10] For details and info on the ribosome exit tunnel see for example the recent book by Springer on nascent polypeptide chain (10).

to archae, bacteriae and eukaryotes (for example, some part of the eukaryotic exit tunnels is associated with protein eL39, which is not present in prokaryotes)[11]. This system was introduced to alleviate the confusion between homologous proteins which were separately given different codenames. In particular, L22 was previously known as L17 in eukaryotes.

Locating the constriction site If one chooses to display the chains in *Pymol*, it appears that other codenames are used to designate uL4 and uL22. To find the right codenames, go to the PDB webpage, look at the "Macromolecules" section and look for ribosomal proteins L4 and L17 (Fig. 2): The corresponding chains are "LC" and "LP". Go to *Pymol* and in the command line type `select chain LC`.

You can copy this selection to an object and rename it L4 (Fig. 3). To visualize it better, you can for example show spheres and color it. After doing the same for L22, the constriction site should be easy to recognize.



```
 You clicked /4ug0/A/L5/A`1873/C3'
 Selector: selection "sele" defined with 22 atoms.
 You clicked /4ug0/A/L5/A`1939/C2
 Selector: selection "sele" defined with 0 atoms.
PyMOL>select chain LP
 Selector: selection "sele" defined with 1274 atoms.
PyMOL>select chain LC
 Selector: selection "sele" defined with 2979 atoms.
```

```
/4ug0/A/L5/ 1         6         11        16        21        26        31
            C G C G A C C U C A G A U C A G A C G U G G C G A C C C G C U G A A U
```

```
PyMOL>set_name obj01,L4
PyMOL>select chain LP
 Selector: selection "sele" defined with 1274 atoms.
PyMOL>set_name obj01,L22
 You clicked /L22/R/LP/ARG`135/N
 Selector: selection "sele" defined with 11 atoms.
 You clicked /L22/R/LP/ARG`135/CA
 Selector: selection "sele" defined with 0 atoms.
```

```
 /L4/F/LC/  1    6    11   16   21   26   31   36   41   46   51   56   61   66
            MACARPLISVYSEKGESSGKNVTLPAVFKAPIRPDIVNFVHTNLRKNNRQPYAVSELAGHQTSAESWGT
/L22/II/LP/    /R/LP/1    6    11   16   21   26   31   36   41   46   51   56
            MG        MVRYSLDPENPTKSCKSRGSNLRVHFKNTRETAQAIKGMHIRKATKYLKDVTLQKQCVPF
```

Figure 3: Left: Selecting the L4 protein in *Pymol*. Right: Proteins L4 and L22 displayed in *Pymol*.

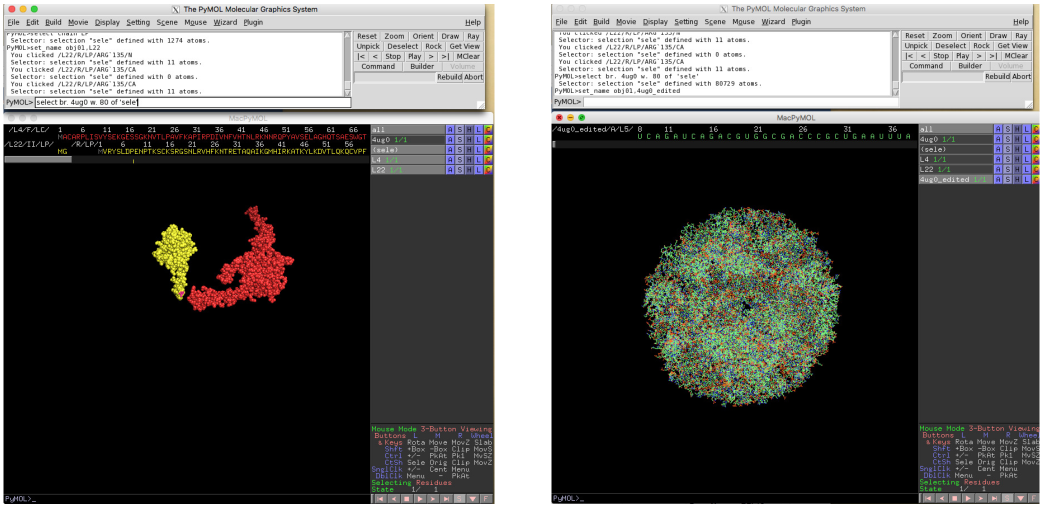
Editing the structure The tunnel is $\sim 80 - 100$Å long. We thus shall only select atoms located within 80 Å of the constriction site to reduce the

[11] L also designates proteins of the large subunit, in contrast with that of the small subunit (S).

cost of the tunnel search algorithm. Click on one of the amino acids located at the constriction site to select it and type the following command to get all residues of the ribosome located within 80 Angstroms[12] (Fig. 4):

```
select br. 4ug0 w. 80 of 'sele'.
```

As for L4 and L22, copy this to an object (called 4ug0_edited). Then, go to File, Save_Molecule, select 4ug0_edited and save it as a pdb file. This is the structure that *MOLE* will use to compute the tunnel.



```
 Selector: selection "sele" defined with 1274 atoms.
PyMOL>set_name obj01,L22
 You clicked /L22/R/LP/ARG`135/N
 Selector: selection "sele" defined with 11 atoms.
 You clicked /L22/R/LP/ARG`135/CA
 Selector: selection "sele" defined with 0 atoms.
 You clicked /L22/R/LP/ARG`135/CA
 Selector: selection "sele" defined with 11 atoms.
PyMOL> select br. 4ug0 w. 80 of 'sele'
```

```
 /L4/F/LC/  1    6    11   16   21   26   31   36   41   46   51   56   61   66
            MACARPLISVYSEKGESSGKNVTLPAVFKAPIRPDIVNFVHTNLRKNNRQPYAVSELAGHQTSAESWGT
/L22/II/LP/    /R/LP/1    6    11   16   21   26   31   36   41   46   51   56
            MG      MVRYSLDPENPTKSCKSRGSNLRVHFKNTRETAQAIKGMHIRKATKYLKDVTLQKQCVPF
```

```
 Selector: selection "sele" defined with 11 atoms.
 You clicked /L22/R/LP/ARG`135/CA
 Selector: selection "sele" defined with 0 atoms.
 You clicked /L22/R/LP/ARG`135/CA
 Selector: selection "sele" defined with 11 atoms.
PyMOL>select br. 4ug0 w. 80 of 'sele'
 Selector: selection "sele" defined with 80729 atoms.
PyMOL>set_name obj01,4ug0_edited
PyMOL>
```

```
/4ug0_edited/A/L5/ 8    11   16   21   26   31   36
                   U C A G A U C A G A C G U G G C G A C C C G C U G A A U U U A
```

The PyMOL Molecular Graphics System
File Edit Build Movie Display Setting Scene Mouse Wizard Plugin Help
Reset Zoom Orient Draw Ray
Unpick Deselect Rock Get View
|< < Stop Play > >| MClear
Command Builder Volume
Rebuild Abort
MacPyMOL
all
4ug0 1/1
(sele)
L4 1/1
L22 1/1
4ug0_edited 1/1
Mouse Mode 3-Button Viewing
Selecting Residues
State 1/ 1
PyMOL>_

Figure 4: Left: Selecting some amino acid at the constriction site. Right: Finding all residues located within 80 Å from this selected amino acid (the tunnel gets visible).

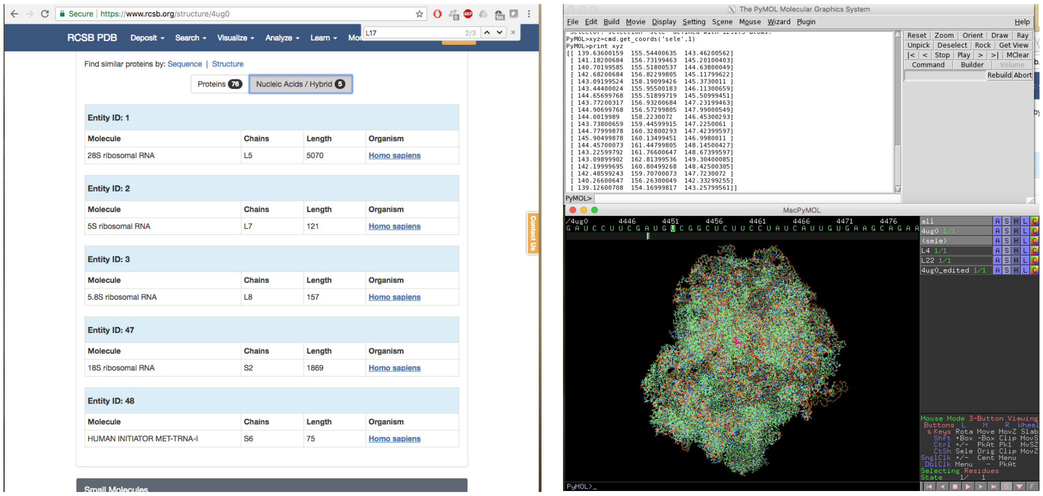
Locating the PTC To get better results with the tunnel search, one should also give a point to initiate the search. For the ribosome exit tunnel, it is natural to take the polypeptide transferase center, which is where the amino acid of the tRNA gets incorporated to the nascent polypeptide chain. In human, this PTC can be located at a specific nucleotide of the ribosomal RNA, that is U4452 [13]. First find the codename for the rRNA 25S as previously (L5). Go to *Pymol*, display the residues and select U4452 (select chain

[12] More details on the selection algebra in *Pymol* can be found here.

[13] Fortunately, the PTC is well conserved so in practice, for other species, one can easily find the PTC by doing sequence alignment of the ribosomal RNAs and find the nucleotide aligned with U4452 in eukaryotes. For E. coli, U2585 should be the nucleotide to align with other bacterial ribosome...

L5 to help finding it). Now print its coordinates by running on the command line:

```
xyz=cmd.get_coords('sele',1)
print xyz
```



Secure | https://www.rcsb.org/structure/4ug0

Find similar proteins by: Sequence | Structure

Entity ID: 1			
Molecule	Chains	Length	Organism
28S ribosomal RNA	L5	5070	Homo sapiens

Entity ID: 2			
Molecule	Chains	Length	Organism
5S ribosomal RNA	L7	121	Homo sapiens

Entity ID: 3			
Molecule	Chains	Length	Organism
5.8S ribosomal RNA	L8	157	Homo sapiens

Entity ID: 47			
Molecule	Chains	Length	Organism
18S ribosomal RNA	S2	1869	Homo sapiens

Entity ID: 48			
Molecule	Chains	Length	Organism
HUMAN INITIATOR MET-TRNA-I	S6	75	Homo sapiens

```
PyMOL>xyz=cmd.get_coords('sele',1)
PyMOL>print xyz
[[ 139.63600159  155.54400635  143.46200562]
 [ 141.18200684  156.73199463  145.20100403]
 [ 140.70199585  155.51800537  144.63800049]
 [ 142.68200684  156.82299805  145.11799622]
 [ 143.09199524  158.19099426  145.3730011 ]
 [ 143.44400024  155.95500183  146.11300659]
 [ 144.65699768  155.51899719  145.50999451]
 [ 143.77200317  156.93200684  147.23199463]
 [ 144.90699768  156.57299805  147.99000549]
 [ 144.0019989   158.2230072   146.45300293]
 [ 143.73800659  159.44599915  147.2250061 ]
 [ 144.77999878  160.32800293  147.42399597]
 [ 145.90499878  160.13499451  146.9980011 ]
 [ 144.45700073  161.44799805  148.14500427]
 [ 143.22599792  161.76600647  148.67399597]
 [ 143.09899902  162.81399536  149.30400085]
 [ 142.19999695  160.80499268  148.42500305]
 [ 142.48599243  159.70700073  147.7230072 ]
 [ 140.26600647  156.26300049  142.33299255]
 [ 139.12600708  154.16999817  143.25799561]]
```

Figure 5: Left: The reference for 25S RNA in the pdb page. Right: Coordinates of the PTC displayed in *Pymol*.

Pick for example $(x, y, z) = (146, 160, 147)$.

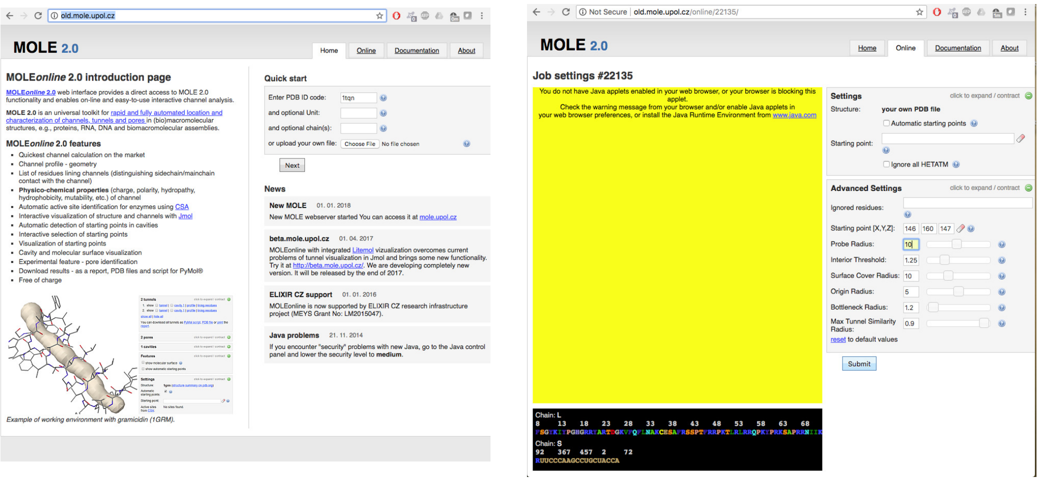
2. Running the tunnel search algorithm

We now have all the required input data to run the tunnel search. Go to the MOLEonline website[14]. After upload the pdb file (4ug0_edited.pdb). Choose Advanced Settings, fix the starting point to (146,160,147) and Probe Radius parameter to 10[15] (Fig. 6). We are now good to submit the job, which should take a couple of minutes. Once the algorithm is done, results are displayed in the main page. The algorithm should find 7 different tunnels. There is a report that can be downloaded for complete details of the pores,

[14] I am using a version of MOLEonline (2.0) that got recently updated to version 3.0 (available here) (Fig. 6). I tried the new version just after it came out but had some trouble to getting good results, because the algorithm gave too many possible structures. Maybe this issue has been fixed since then...

[15] I played with this parameter and noticed that this value basically gives better results; maybe it would be interesting to look more in details at the algorithm to see why

cavities and tunnels found. There is also a Python script, which can be run by *Pymol* to display the tunnels.



MOLE*online* 2.0 introduction page

MOLE*online* 2.0 web interface provides a direct access to MOLE 2.0 functionality and enables on-line and easy-to-use interactive channel analysis.

MOLE 2.0 is an universal toolkit for rapid and fully automated location and characterization of channels, tunnels and pores in (bio)macromolecular structures, e.g., proteins, RNA, DNA and biomacromolecular assemblies.

MOLE*online* 2.0 features

- Quickest channel calculation on the market
- Channel profile - geometry
- List of residues lining channels (distinguishing sidechain/mainchain contact with the channel)
- **Physico-chemical properties** (charge, polarity, hydropathy, hydrophobicity, mutability, etc.) of channel
- Automatic active site identification for enzymes using CSA
- Interactive visualization of structure and channels with Jmol
- Automatic detection of starting points in cavities
- Interactive selection of starting points
- Visualization of starting points
- Cavity and molecular surface visualization
- Experimental feature - pore identification
- Download results - as a report, PDB files and script for PyMol®
- Free of charge

2 tunnels
click to expand / contract
1. show □ tunnel | □ cavity 1 | profile | lining residues
2. show □ tunnel | □ cavity 1 | profile | lining residues
show all | hide all
You can download all tunnels as PyMol script, PDB file or print the report.
2 pores
1 cavities
Features
□ show molecular surface
□ show automatic starting points
Settings
Structure: 1grm (structure summary on pdb.org)
Automatic starting points: ☑
Starting point:
Active sites from CSA: No sites found.

Example of working environment with gramicidin (1GRM).

Quick start

Enter PDB ID code: 1tqn

and optional Unit:

and optional chain(s):

or upload your own file: Choose File No file chosen

Next

News

New MOLE 01. 01. 2018
New MOLE webserver started You can access it at mole.upol.cz

beta.mole.upol.cz 01. 04. 2017
MOLEonline with integrated Litemol vizualization overcomes current problems of tunnel visualization in Jmol and brings some new functionality. Try it at http://beta.mole.upol.cz/. We are developing completely new version. It will be released by the end of 2017.

ELIXIR CZ support 01. 01. 2016
MOLEonline is now supported by ELIXIR CZ research infrastructure project (MEYS Grant No: LM2015047).

Java problems 21. 11. 2014
If you encounter "security" problems with new Java, go to the Java control panel and lower the security level to **medium**.

Job settings #22135

You do not have Java applets enabled in your web browser, or your browser is blocking this applet.
Check the warning message from your browser and/or enable Java applets in your web browser preferences, or install the Java Runtime Environment from www.java.com

```
Chain: L
8    13   18   23   28   33   38   43   48   53   58   63   68
FSGYKIYPGHGRRYARTDGKVFQFLNAKCESAFRSSPTFRRPKTLRLRRQPKYPRKSAPRRNIIK
Chain: S
92   367  457  2    72
RUUCCCAAGCCUGCUACCA
```

Settings click to expand / contract

Structure: **your own PDB file**

□ Automatic starting points

Starting point:

□ Ignore all HETATM

Advanced Settings click to expand / contract

Ignored residues:

Starting point [X,Y,Z]: 146 160 147

Probe Radius: 10

Interior Threshold: 1.25

Surface Cover Radius: 10

Origin Radius: 5

Bottleneck Radius: 1.2

Max Tunnel Similarity Radius: 0.9

reset to default values

Submit

Figure 6: Left: The MOLEonline main page Right: After uploading the structure, we set the parameters of the tunnel search algorithm.

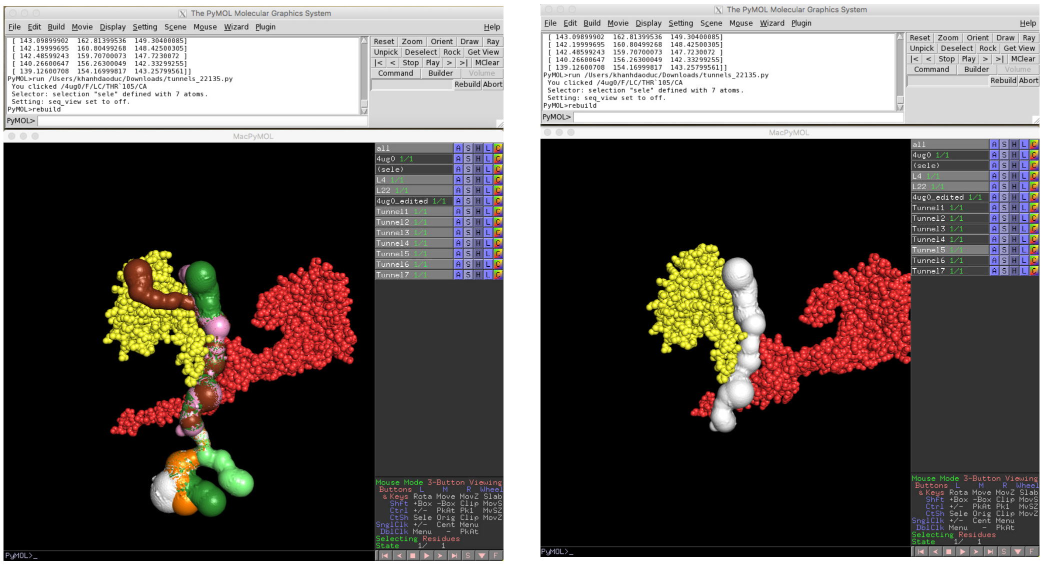
3. Visualization of the output

Download and run the python script in pymol (Go to File, Run Script), which should display the 7 tunnels that the algorithm found[16]. You can see that most of them are just due to the algorithm finding tunnels in the wrong direction (towards the small subunit and tRNA instead of the tunnel exit), or crawling weirdly along the surface as it reaches the exit. After removing these, you should now see the ribosome exit tunnel (Fig. 7).

4. Exporting the coordinates

If you are interested in getting the geometry, one last thing to do is to export the coordinates of the tunnel, which are given by 1) the centerline

[16] By default, my version of Pymol would display the tunnels in surface mode with a quality that makes not them being displayed properly (Fig. 7). To fix it, you can choose to increase the quality (takes more ressources to run Pymol) or display spheres instead of surface.



```
[ 143.09899902  162.81399536  149.30400085]
[ 142.19999695  160.80499268  148.42500305]
[ 142.48599243  159.70700073  147.7230072 ]
[ 140.26600647  156.26300049  142.33299255]
[ 139.12600708  154.16999817  143.25799561]]
PyMOL>run /Users/khanhdaoduc/Downloads/tunnels_22135.py
 You clicked /4ug0/F/LC/THR`105/CA
 Selector: selection "sele" defined with 7 atoms.
 Setting: seq_view set to off.
PyMOL>rebuild
```

```
[ 143.09899902  162.81399536  149.30400085]
[ 142.19999695  160.80499268  148.42500305]
[ 142.48599243  159.70700073  147.7230072 ]
[ 140.26600647  156.26300049  142.33299255]
[ 139.12600708  154.16999817  143.25799561]]
PyMOL>run /Users/khanhdaoduc/Downloads/tunnels_22135.py
 You clicked /4ug0/F/LC/THR`105/CA
 Selector: selection "sele" defined with 7 atoms.
 Setting: seq_view set to off.
PyMOL>rebuild
```

Figure 7: Left: The tunnels found by MOLE. Right: The ribosome exit tunnel ('Tunnel 5') found after removing artefacts.

2) the radius at each point of the centerline. To get the coordinates of the centerline, run the following on the command line[17][18]:

```
xyz=cmd.get_coords('Tunnel5',1)
r=[]
cmd.iterate_state(1,'Tunnel','r.append(vdw)',space=locals(),atomic=0)
python
from pymol import stored
np.savetxt('tunnel_coordinates.txt',xyz,fmt='\%.2f')
np.savetxt('tunnel_radius.txt',r,fmt='\%.2f')
python end
```

[17] It is also possible to directly write a python script and run it with *Pymol*, as done previously with the Tunnel script generated by MOLE...

[18] There are many things that can be done once the tunnel is found. For example, one can look for the chains that are associated with the tunnel, the presence of charged amino acids around the tunnel, get all the atoms or residues located at a certain distance, their properties etc.

Problem setting

Advantages and drawbacks of the current method

Extracting refined geometry

Tunnel metric comparison

Bibliography

1. Sehnal D, Vařeková RS, Berka K, Pravda L, Navrátilová V, Banáš P, et al. MOLE 2.0: advanced approach for analysis of biomacromolecular channels. Journal of cheminformatics. 2013;5(1):39.

2. Krone M, Kozlíková B, Lindow N, Baaden M, Baum D, Parulek J, et al. Visual analysis of biomolecular cavities: state of the art. In: Computer Graphics Forum. vol. 35. Wiley Online Library; 2016. p. 527–551.

3. Dao Duc K, Song YS. The impact of ribosomal interference, codon usage, and exit tunnel interactions on translation elongation rate variation. PLoS genetics. 2018;14(1):e1007166.

4. Coleman RG, Sharp KA. Finding and characterizing tunnels in macromolecules with application to ion channels and pores. Biophysical journal. 2009;96(2):632–645.

5. Voss NR, Gerstein M. 3V: cavity, channel and cleft volume calculator and extractor. Nucleic acids research. 2010;38(suppl_2):W555–W562.

6. Sehnal D, Vařeková RS, Berka K, Pravda L, Navrátilová V, Banáš P, et al. MOLE 2.0: advanced approach for analysis of biomacromolecular channels. Journal of cheminformatics. 2013;5(1):1.

7. Pravda L, Sehnal D, Berka K, Navrátilová V, Toušěk D, Bazgier V, et al. Channelsdb and Moleonline-Database and Tool for Analysis of Biomacromolecular Tunnels and Pores. Biophysical Journal. 2018;114(3):342a–343a.

8. Pravda L, Sehnal D, Svobodová Vařeková R, Navrátilová V, Toušek D, Berka K, et al. ChannelsDB: database of biomacromolecular tunnels and pores. Nucleic acids research. 2017;46(D1):D399–D405.

9. Chovancova E, Pavelka A, Benes P, Strnad O, Brezovsky J, Kozlikova B, et al. CAVER 3.0: a tool for the analysis of transport pathways in dynamic protein structures. PLoS computational biology. 2012;8(10):e1002708.

10. Ito K, editor. Regulatory Nascent Polypeptides. Springer; 2014.

11. Ban N, Beckmann R, Cate JH, Dinman JD, Dragon F, Ellis SR, et al. A new system for naming ribosomal proteins. Current opinion in structural biology. 2014;24:165–169.