07/12 lecture

How to select the templates

We have to establish some rules, the template should be selected on the basis of its *complete* annotation. e.g. in the last month high-resolution laccases have been available, but they are not yet in swiss-prot, only TrEMBL. Proteins receive annotations that are 100% automatic.

Plan for today:

1- RasTop/RasMol tutorial

2- More about Structural alignment

3- Explanation of Modeller

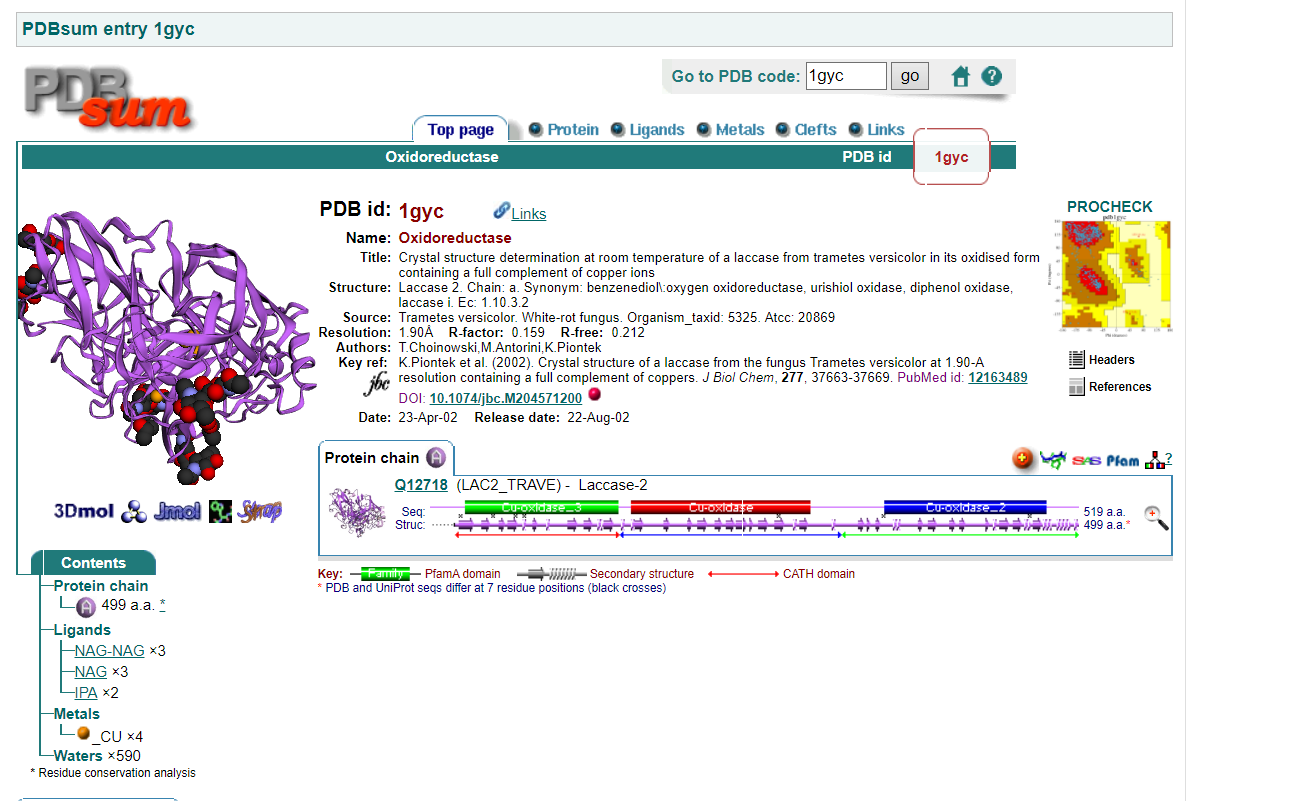
4- Carrying out first experiment

Suppose I want to visualize a structure in the PDB. Let’s take the crystal 1GYC.

Which is the functional unit of the protein? (sometimes, oligo-polymers are functional and monomers are not). If the monomer is solved at atomic resolution, it’s not enough to have it, we also have to retrieve the information of whether the functional unit is monomer, dimer, etc.

PDBsum -> part of EMBL EBI institute

If you want to understand exactly how the enzyme works and in PDBsum it’s not listed, you have to go back to literature and see what the original paper says.



laccase> functional as a monomer

pdp>uniprot>pdb sum

PDBsum contains a quick view of the atomic resolution of a protein , information about the ligands, the protein domain… it’s like same information as the PDB entry but more concise.

Important!!! Ramachandran plot of the protein, which gives us information about how “legitimate” that structure is and how stable that protein is. Ramachandran plot also gives information about secondary structure.

(((some revising about the Ramachandran plot can be found here:

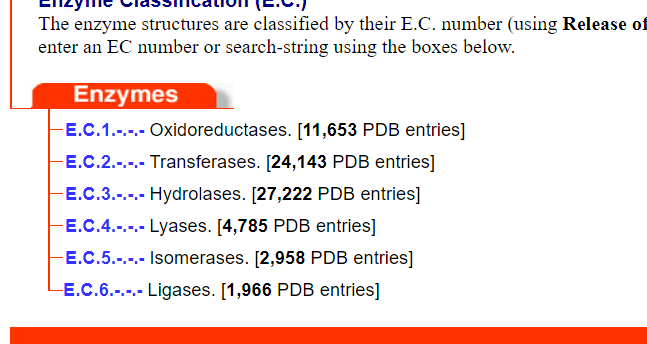
<https://proteinstructures.com/Structure/Structure/Ramachandran-plot.html>)))

After modeling with Modeller, we will retrieve several models and we will choose the best one, adopting Procheck as a validation program, but the evaluation will also consider the validity of the Ramachandran plot.

There is a derived database that collects/”cross-links” information about ec number and structure is called ECtoPDB -> European database for *enzyme structure*

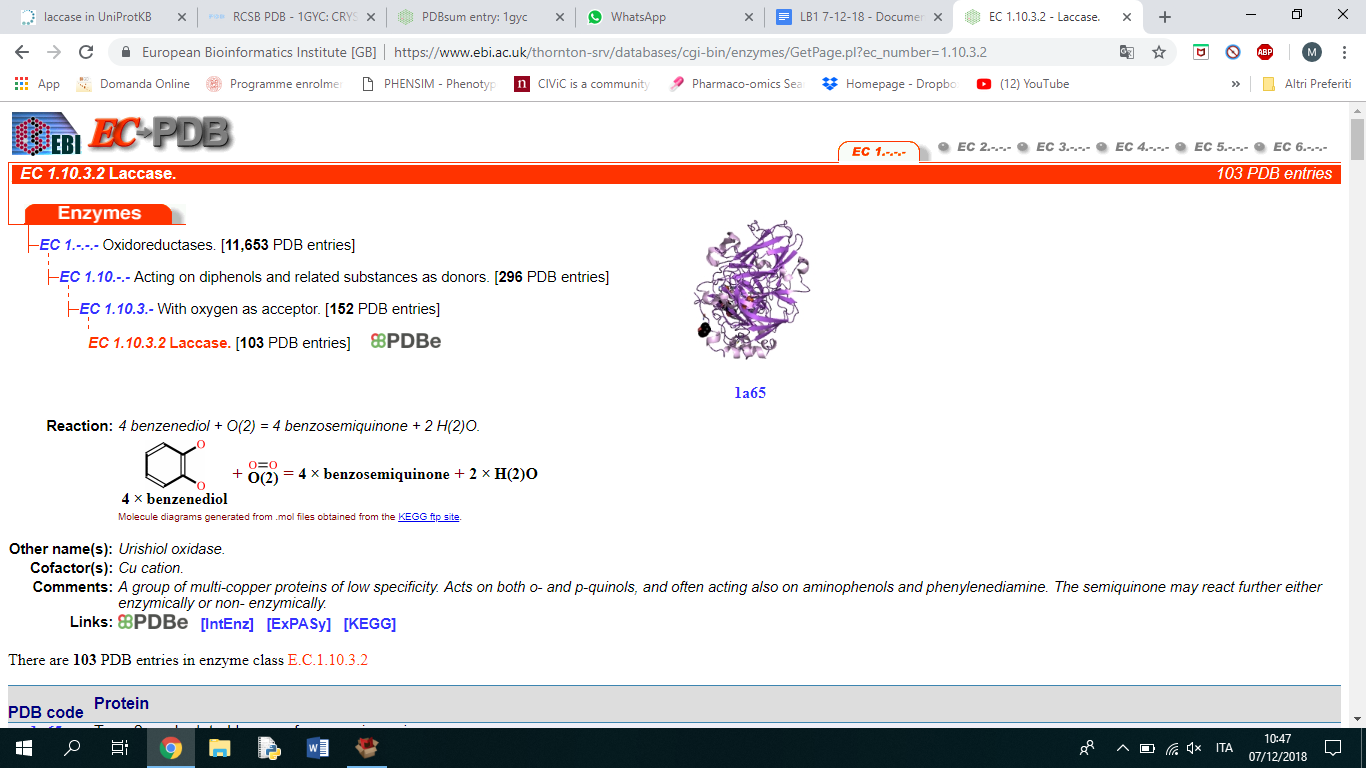
*here you have* major classes: 6 classes (EC 1->6)

Note: there is going to be a seventh class but it’s empty now (it will be introduced as the class of Translocases in the future)



Each class is correlated to the relative number of entries in the PDB belonging to that class.

One EC number -> many many many PDB files. They might not be all proper laccases!



In order to find the right template for our target, the sequence needs to be blasted against the pdb,

In principle, we’re not using EC-PDB to search for our template at the beginning.

having a sequence, we know which type of enzyme it is and the organism it comes from.

first work with laccase: we want to functionally annotate the protein:

* using the sequence, use blast to retrieve from uniprot a putative template: select the reference database that contains only protein with structure (BLAST against PDB).
* then i find the template
* we went to the paper that describes the structure of this protein and retrieved information about what is its role and which elements/residues make it work (e.g. how the active site is structured).
* we want to perform functional annotation via structural modelling of target against template (of which we know the structure, so we use it to infer the structure of the target. The higher %identity, the better the inference)

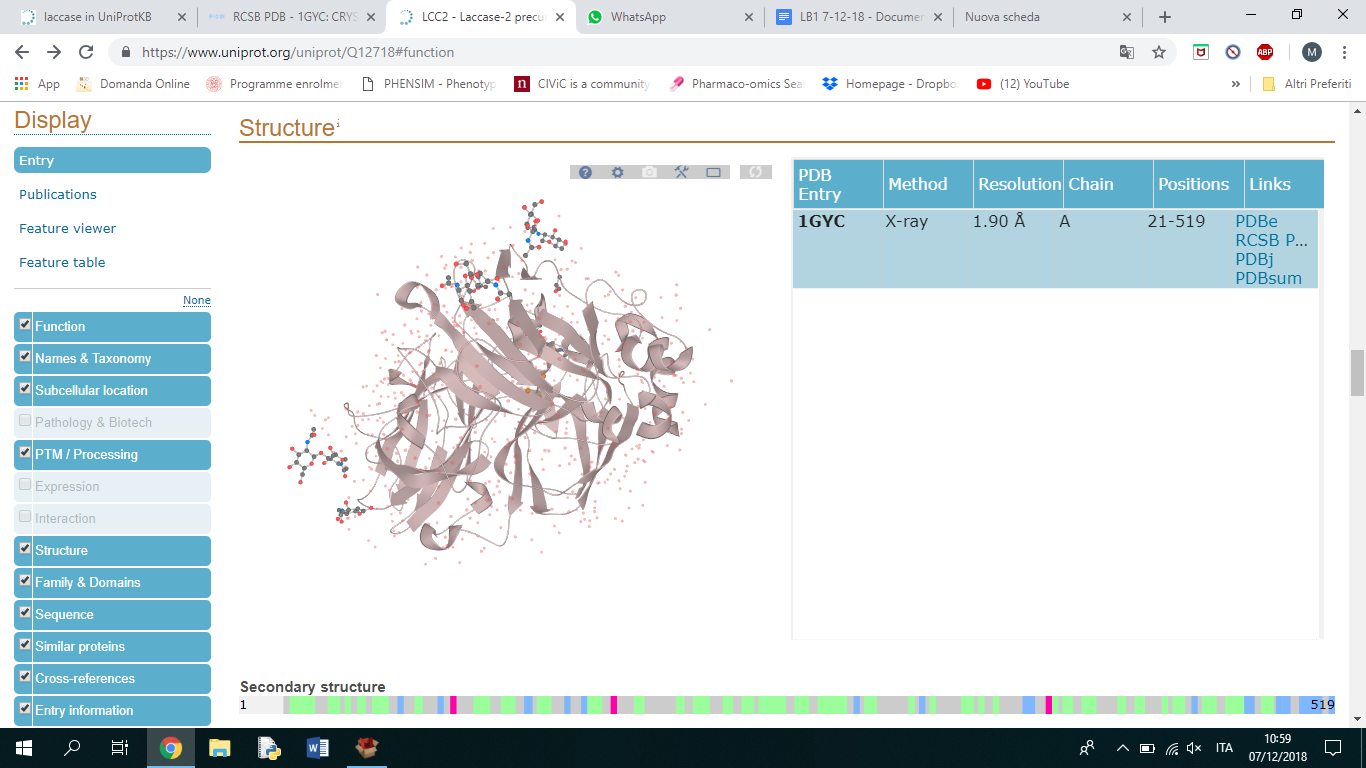
functional annotation: set of go terms that characterize the template of the family. How many terms can we transfer? How to decide? This is the aim of our work.

if we understand the characteristics of the template (and from that, the characteristics of the family, since same family = same function achieved with same active site conformation), then after having realized that our model has all the constraints that have been derived, we can derive the GO terms that characterize this template. (i.e. we can transfer functional annotation from the template to the target, because we have checked that they are most likely similar in structure).

In the UniProt entry corresponding to 1GYC, there is only one PDB entry. Since it’s obtained by X-ray, we also have a resolution measure which is 1.90Å. Uniprot is linked to PDB showing also what residues have been successfully crystallized. For this entry, the crystal is covering positions 21-519 (length = 499 in the PDB entry, length = 519 in UniProt, because the first 20aa belong to the signal peptide that was cleaved during the amplification and purification process of the protein, so it’s not present in the crystal)

In the PDB entry 5Z1X the resolution is higher.

starting from here:



### Structure section

This section provides information on the tertiary and secondary structure of a protein.

The information is filed in different subsections. The current subsections and their content are listed below:

|  |  |
| --- | --- |
| **Subsection** | **Content** |
| [Turn](https://www.uniprot.org/manual/turn) | Turns within the experimentally determined protein structure |
| [Beta strand](https://www.uniprot.org/manual/strand) | Beta strand regions within the experimentally determined protein structure |
| [Helix](https://www.uniprot.org/manual/helix) | Helical regions within the experimentally determined protein structure |
| [3D structure databases](https://www.uniprot.org/manual/cross_references_section) | Cross-references that point to data collections other than UniProtKB (i.e. PDB) |

Visualization of the structure in UniProt entries

At some point in the UniProt entry, I have a structure function. If a crystal of the protein has been obtained, I can visualize it. I have at the basis the **pdb file** (donwload the pdb file with the original data, that contains the *coordinates* of the electron density map of the atom. it is already a reduced representation of the electron density data aka **real protein structure**. It’s already an approximation!) .

pdb file: complicated to be visualized simply by eye.

so a quick overview about the pdb file is possible through the global visualization program.

several database have already a reduce representation of the electron density focused on the function:

Visualizers provide a user-friendly visualization, to have a quick look at the ”””protein structure”””, represented by highlighting the backbone.

any PDB file coming from electron density data has to go through refinement by means of adding the hydrogen atom coordinates (in x ray diffraction hydrogen atoms are not represented).

i can use a program that is called DSSP -> algorithm to predict secondary structure regions (dictionary of secondary structure protein)

this program dssp, on the basis of the chemical physical and geometric parameteres, this programs compute in which position you may have an hydrogen and the position of hydrogen bonds.

the crystallographers when depositing the protein also declare where the secondary structure regions are located.

computes in which position are the H located (in order to know where the H-bonds defining the secondary structure are located, I need of course to know where the hydrogens are). (also side note/reminder, secondary structure is almost always defined by H-bonds involving the atoms of the **backbone**, think of how alpha-helices are structured).

Visualisers have their own rules to define when to display a beta strand rather than an alpha helix or a loop. In order to do this a program like DSSP, that can predict the place of H, is needed.

es:for the crystallographers the alpha helix are divided in two pieces

----> GRAPHICAL VISUALIZATION NEEDS THE PDB FILE TO RUN FIRST THROUGH THE DSSP

every program that allows you to derive the graphical visualization needs another programs that allocates the hydrogen.-->DSSP

not all protein have deposited electron density maps

## Electron Density Maps

Electron density maps combine the structural model (coordinates) and the experimentally-collected data from an X-ray structure determination and serve to represent the fit of the model to the data. There are two types of electron density maps commonly used by researchers: the 2fo-fc map and the fo-fc map. The fo-fc (also called a difference or omit map) map shows what has been overrepresented or not accounted for by the model, while the 2fo-fc map includes the fo-fc map and electron density around the model.

These two maps are then used to correct the model when possible. Even in the best quality structures, there are areas of poor electron density, which may represent sections of the model that exist in multiple conformations. This can be seen in long side chains or surface loops of the model.

For additional background, please see [Structure Factors and Electron Density](http://pdb101.rcsb.org/learn/guide-to-understanding-pdb-data/structure-factors-and-electron-density) at PDB-101 and for additional information on the generation and downloading of maps at the RCSB, please see the [X-ray Electron Density Maps FAQ](http://www.rcsb.org/pages/help/edmaps).

The Electron Density Maps tab offers various display options:

* Map: Turn on or off the 2fo-fc map display (colored blue) and/or the fo-fc (negative density is colored red and positive peaks are colored green).
* Scroll: Select the map whose isolevel value will be changed with the mouse via "Ctrl scroll-wheel". See also: the Level option.
* Level: The value at which the isosurface of the map is shown (in sigma values). This value can be adjusted using the sliders for each map type or using the mouse via "Ctrl scroll-wheel".
* Map Style: Select the map display style (mesh, smooth, or flat).
* Map Size: For efficiency, only a portion of the map is displayed. The map size option adjusts the size (in Å along each of the three crystallographic axes) of this portion.
* Ligand: Recenter the view around the selected ligand.
* Default maps view: Adjusts the view settings to be optimized for electron density map display.

Suggested visualiser → Rasmol

* easier to use

we are now going to install it on linux

sudo apt install <program\_name>

Linux repositories ---> ??? (what are those? how are they related to sudo apt command? Why manual installing of packages should be avoided?)

Side note on apt -> apt is a package manager

header in pdb file → key to retrieve info in automatic way and to display molecules in visualiser

-contain many info

remarks→ experimental conditions

secondary structure, as declared by the crystallographer =/= displayed by the visualiser that uses its own rules

glycosylation

heteroatoms → water (only oxygen coordinates coming from the actual electron density data, hydrogens are predicted), ions

ligands

Some of this information (based on the header) is discarded by the visualizing software, some is used to display and render the molecule.

RASMOL MANUAL! (refer to this for commands)

<http://www.rasmol.org/doc/rasmol.html>

Download the pdb file -> <https://files.rcsb.org/view/1GYC.pdb>

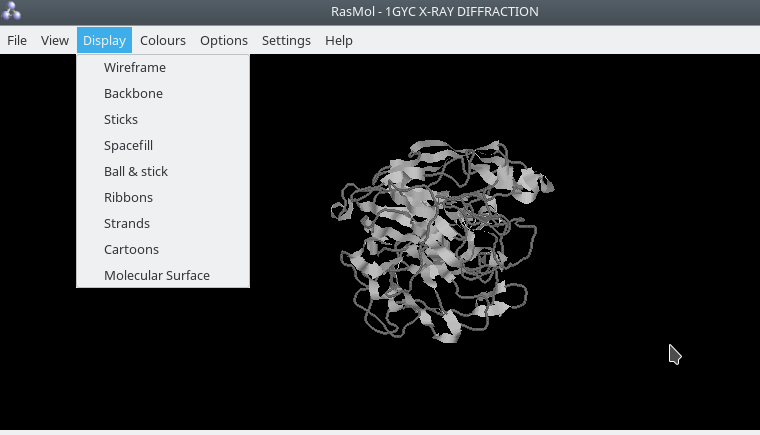
open file from terminal

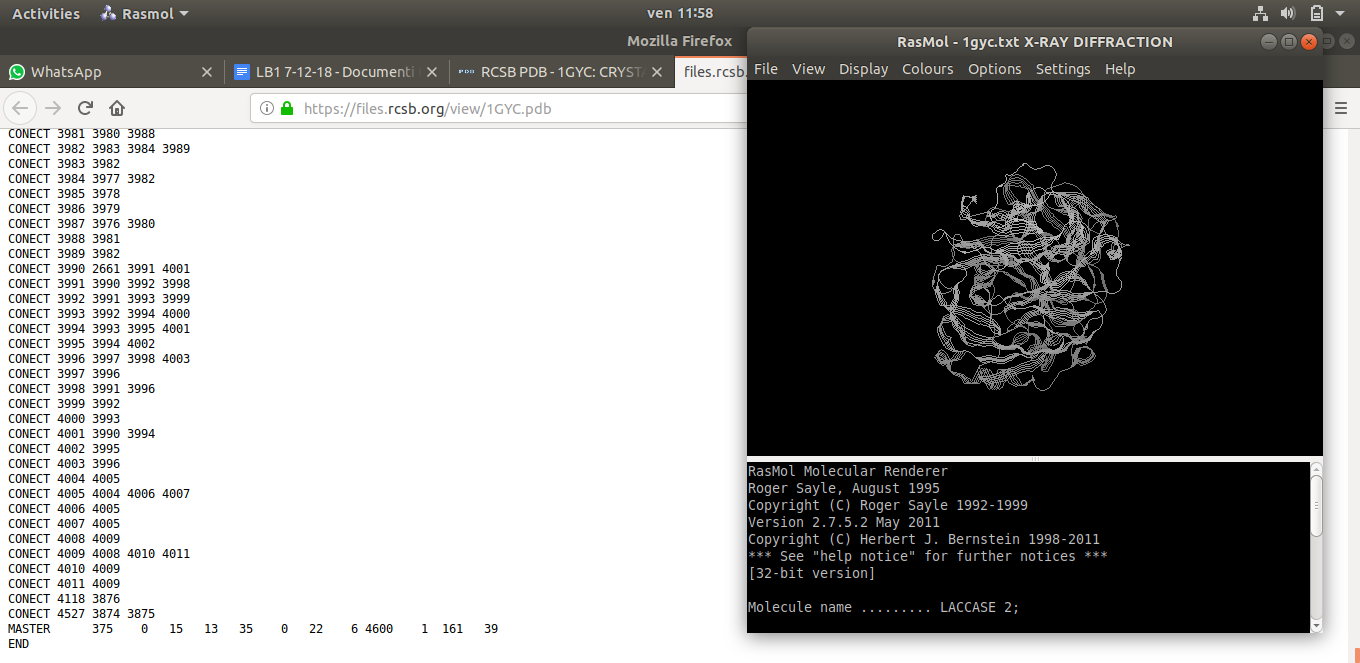
rasmol <file\_name.pdb>

then View>Command prompt

by default it displays basic information on the structure

To change the modality of representation Display> select one option





Secondary structure information -> sometimes the sec structure displayed on the visualizer has some aa of difference

is this content matching the real content of the x ray diffraction? there are a couple of residues of difference

display:

wireframe: representing your molecule in a sort of complete way.

sticks: atoms are colored according to the cpk way, chemical coding mode to give a color to different atoms. oxigen: red, sulfore: yellow, nitrogen: blu, carbon: white.

here you have also the skeleton of all the residues and the backbone.

spacefill: approximation of the volume of the electron density (with approximation) around each point .

there is a sort of a electron density for each point represented with the wan der waals sphere.

thi s is a way to representing:steric hindrance of the molecule = occupancy of volume in the solvent

here you have the red sphere represent the oxygen (so the water molecule)

Van der Waals surfaces of each atom are displayed -> many bubbly spheres

CPK -> what is the rationale behind the color code? it’s just a convention

if you type ‘restrict not hetero’ in the command line →it displays everything but the hetero atoms (((refer to manual link above for some more nice commands)))

background <color> changes the background color

focus on the active site of the molecule:

the relative location of the coppers and the nearby residue .

coppers are stabilized by coordination bonds with histidin .

which is the atom in histidine molecule that stabilize coppers? nitrogen.

there is a command called select within(distance, object to measure the distance from) that allows you to know the distance between the atoms.

select cu=copper: to see where the coppers are

pick distance

then we can click on the atom that we want to measure.

ant the position of the copper comes out.

and then i can retrieve the distance between copper to copper.

>command: pick distance

>click on two atoms.

protein characteristic: active site (3 coppers), glycosylation event stabilized the protein in order to increase the molecular way, disulphide bridge is also important in this molecule.

Disulphide bonds do not exceed 2A?? (average bonding distance for disulphide bonds: 2 Ångstrom)

the general idea is to visualise the key residues for the protein in order to show whether the target protein and the template are similar or not. Is the active site conserved?

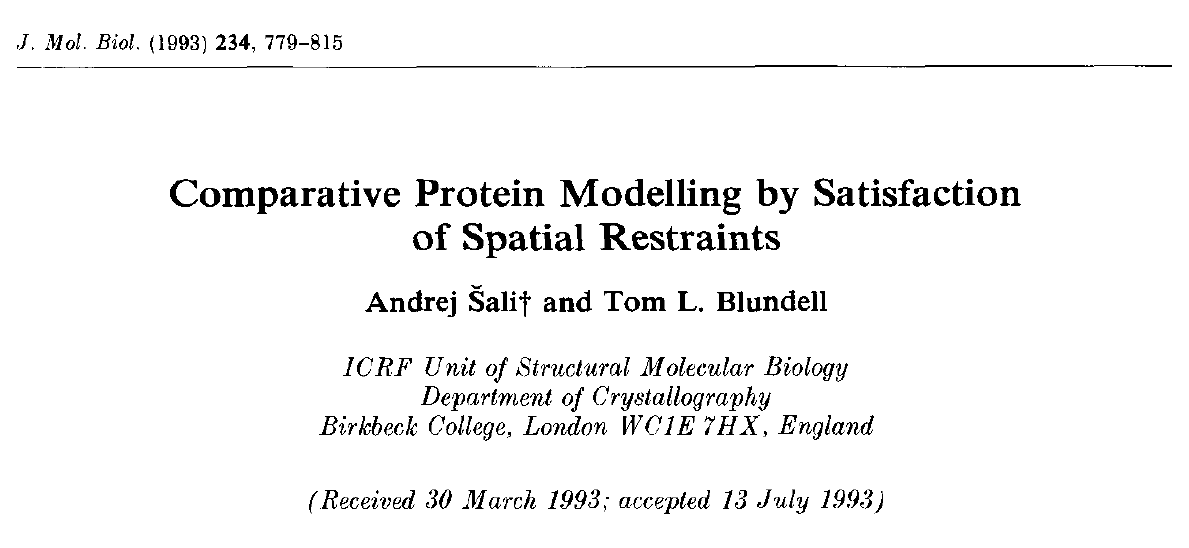
superimposition of the model with the original template

“who doesn’t know how to use a visualisation program? it’s your problem, try to exercise” cit.

modeller:

Read the Modeller article from dropbox over the weekend

AA 2018-2019/LB1-A/IV-ALIGNMENT/III-3Dprediction/references/SAli\_JMolBiol\_1993.pdf



Modeller is based on *satisfaction of spatial restraints!*

you should specify the release of the modeller you’re using

1993 -> today: many releases of Modeller. So release used must be specified in methods.

Manual contains the instructions to use it (ground breaking!)

modeller: computing the structure of a sequence on the basis of spatial renstains

what is modeler ? modeller is a computer programs that models three-dimensional structures of proteins and their assemblies by satisfaction of spatial restraints.

modeller is most frequently used for homology or comparative protein structure modeling: the user provides an alignment of a sequence to be modeled with known related structures and modeller will automatically calculate a model with all non -hydrogen atoms.

What are these spatial restraints we are talking about? How are they satisfied upon protein modelling?

Quotation of modeller in the references -> quote original paper that is already in folder + mention release

*When* is modeller used? -> building/modelling by homology/comparison.

*Input* -> **optimal sequence** alignment between target (what we want to structurally define) and template (what we already have fully characterized), in **PIR** format (refer to yesterday’s tutorial).

spatial restraints: we know that the structure is characterized by its interaction with the solvent.

how can i have a protein folded in this way? in the specific way of the template.

what does spatial restraints means: all the pairwise interaction among atoms(covalent bonds).

when the protein is folded in the solvent there are several non bonding interactions that at the level of the pdb file we may recognize in terms of geometrical distances.

we can divided them in: ion ion interaction, dipol dipole interaction, ion dipole interaction (non covalent interaction derivable from the pdb file).

what is a spatial restraint : we have a structure and we know its characterized 1st of all by its interaction with the solvent (stereochemical hindrance of the molecule) how can we have a protein folded in this way => why the protein is stable? series of *[lamenti sulla vita]* all the pairwise interactions among all atoms

Even the heteropolymer chain has covalent bonds

when the protein is folded what is going on? there are a list of non bonding interactions that **if you don’t know when you are sitting in front of me i will send you home**

we categorize them in 3 types => ion-ion, dipol-dipol, ion-dipol

also H bonds, the most relevant in respect to the folding

derived only from the pdb file+

all these pw interactions they are there

Bonding interactions are “coded” in the PDB file as distances between atoms that are interacting. These distances represent constraints.

All the interactions that serve as a source of stability, provided that the interactors are at a certain distance, are **spatial restraints**, if i’m able to return them in some sort of geometrical restriction.(making stabilizing interaction in order to generate a stabilized protein)

modeler: these new sequence can assume the same folding of the template.

we consider the folder of the protein..

but think of the effect of the substitution of a lateral chain.. the more identity share the two sequences the better it is. because if we change the lateral chain of the protein , the modeller gives you a compromised solution. whatever dihedral angle will come out by the substitution of a residues it’s simply guessing it can be equal or not to the original ones.

level of sequence identity of our template/target couple: 69% (pretty high)

the fold is somehow linked at the molecular level to both backbone and lateral side chains;

We are basically allowed to take our pdb file and replace and take away whatever

pdb to pir adaptation

take the pdb file of the template, eliminate whatever heteroatom except for the coppers, put the copper coordinates at the bottom of the list of atoms (right after the last amino acid atom coordinates)

parameters => seq length

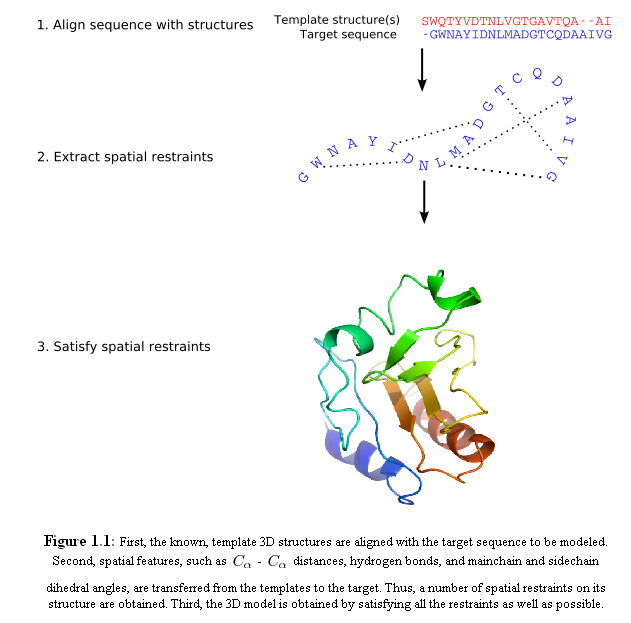
=> heteroatom we want to include

=> we don’t consider the glycosyl groups for now

modeler objective function: parametre that gives the quality of the modelel

modeler: extracting from the protein all the possible sources of stability of the protein

workflow:



(from Casadio’s slides pack “Modeller.pdf”)

Question that Modeller answers to -> whether the structure of the protein is conserved when a slightly different sequence is adapted to it!

the higher the identity percentage, the higher the quality of the modelling (because more similar residues will adapt better to the template structure)

signal peptide is not included in the template structure, so it is not included in the sequence derived from the pdb structure either. So since our target HAS a 20aa signal peptide at the N-terminus, we have to open the FASTA with a text editor and delete those pesky 20 amino acids. In this way, Modeller will not be “confused”

wasn’t the guy talking about some more cutting?

the best template is the one which covers the most of the part

Common problem -> miscalculation of alignment length.