

Image Tutorial: Similar Binding Sites

Background

β -1,4-glycanase (gold, Protein Data Bank entry [1exp](#)) and cellobiohydrolase I (cyan, Protein Data Bank entry [1cel](#)) have different folds but a similar pattern of amino acids near a glucose residue. This possible example of convergent evolution was identified in Table 5 of:

[Detection of protein three-dimensional side-chain patterns: new examples of convergent evolution.](#) Russell RB. *J Mol Biol.* 1998 Jun 26;279(5):1211-27.

Image How-To

The recipe here is just an example; there are usually many routes to the same or similar results. Not all of the steps can be done with commands – for example, there is no substitute for interactively finding the best views – but commands for much of the setup below are collected into the Chimera command file [convergent.com](#).

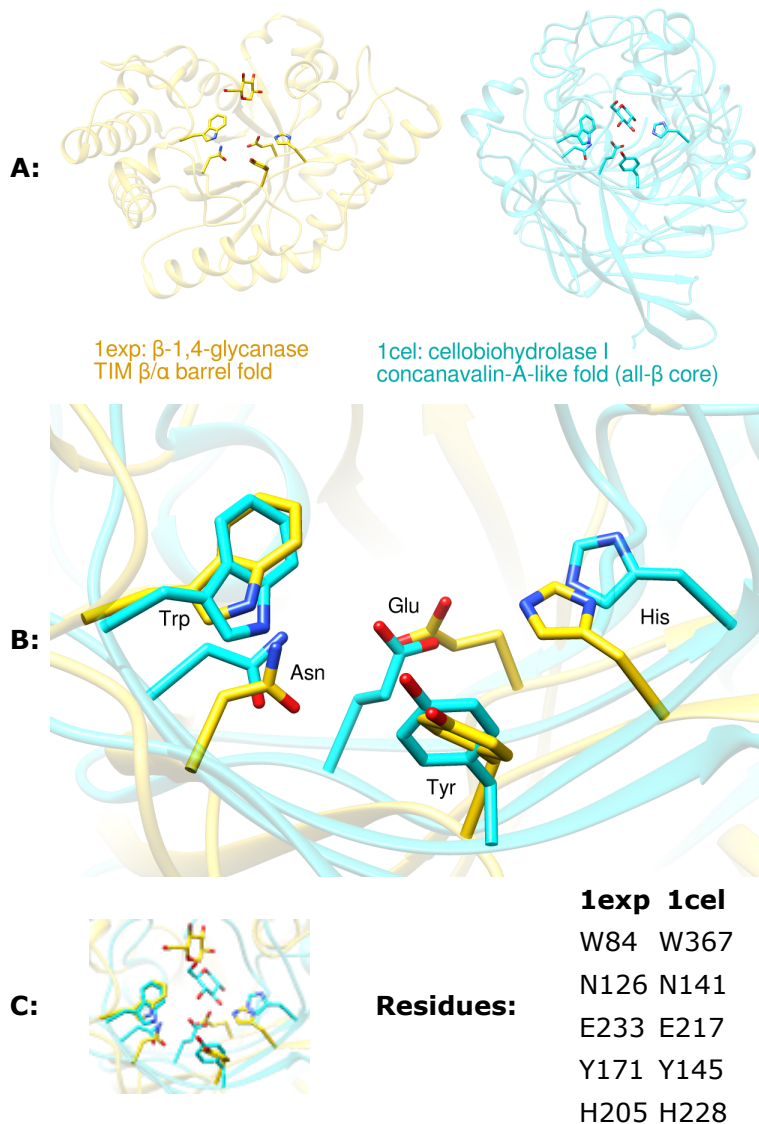
See also: [presets](#), [tips on preparing images](#)

The images show (A) the two proteins side by side, (B) a closeup of the superimposed sites without glucose, and (C) a thumbnail of the superimposed sites. Often it is best to create a set of views that, while different, are consistent or related to each other by only simple transformations. We will generate the view shown in B first and then base the others upon it.

[Start Chimera](#) and enlarge the window as desired. Show the [Command Line](#) (for example, with **Favorites... Command Line**). Fetch **1exp** and then **1cel** from the [Protein Data Bank](#):

Command: [open](#) **1exp**
Command: [open](#) **1cel**
Command: [focus](#)

Similar binding sites in different protein folds



Move and scale the structures [using the mouse](#) and the [Side View](#) as you wish throughout the tutorial.

1cel includes two copies of the protein, chains A and B; one copy can be deleted:

Command: [delete](#) **:.b**

Command: [focus](#)

Use the ribbons preset, and for now, hide all atoms and bonds:

Command: [preset](#) **apply int 1**

Command: [~disp](#)

Set up the basic coloring scheme and white background:

Command: [color](#) **gold #0**

Command: [color](#) **cyan #1**

Command: [background](#) **solid white**

Display the residues identified by Russell and use them to superimpose the structures:

Command: [alias](#) **site1 #0:84,126,233,171,205**

Command: [alias](#) **site2 #1:367,141,217,145,228**

Command: [alias](#) **both site1 | site2**

Command: [disp](#) **both**

Command: [match](#) **iterate 2.0 site2 site1**

Command: [focus](#) **both**

Command: [color](#) **byhet**

The aliases in the above [match](#) command specify entire residues. The match command requires equal numbers of atoms from the two structures, specified in the same order. In this case, the residues are of the same types (thus containing the same numbers of atoms) and the atoms of each residue will be in the same order because they are automatically sorted by name.

Make the sticks and ribbons smoother by increasing subdivision:

Command: [set subdivision](#) **10**

De-Emphasizing the Ribbons

The display is still quite "busy," in part because the two proteins have different folds. We will de-emphasize the ribbons by making them transparent and slimmer. Make the ribbons 75% transparent (25% opaque):

Command: [transparency](#) **75,r**

The **,r** part indicates that only ribbons should be affected, not atoms, bonds, surfaces, *etc.*

By default, transparent surfaces (including objects as well as surfaces proper) appear more opaque when viewed edge-on as compared to face-on. The transparent ribbons will be less obtrusive with *non-angle-dependent* transparency:

Command: [set flatTransparency](#)

The sidechain sticks terminate as rounded "nubbins" inside the ribbon. To make these ends flat

instead:

Command: [repr wire @ca](#)

The ribbon dimensions can be adjusted in [Ribbon Style Editor](#). Start that tool (under **Tools... Depiction** in the menu) and adjust the **Ribbon Scaling** values:

2° Structure	Width	Height
Coil	0.20	0.20
Helix	0.60	0.20
Sheet	0.60	0.20
Arrow (base)	1.0	0.20
Arrow (tip)	0.20	0.20

The **Nucleic** values only pertain to DNA and RNA. Click **Apply** to adjust the ribbon scaling (secondary-structure-specific dimensions). Click **Save As...** to name and save the scaling for later use. For example, if the scaling is named **slim**, it could also be applied with:

Command: [ribscale slim](#)

Generating Positions (Views)

The displayed sidechains are somewhat obscured by other parts of the proteins. Rotate, translate, and scale the structures, and in the [Side View](#), adjust the front and back clipping planes to simplify the view. You may also want to adjust the window dimensions. After generating a good view of the set of displayed residues (such as in [figure B](#)), save the position:

Command: [savepos closeup](#)

It does not have to be perfect; the same command can be used again later to re-save the position after further adjustments. You may also wish to [save the session](#) (see the **File** menu) at this stage and whenever you are pleased with your previous work and want to ensure it will not be lost.

Next, generate and save positions for the other two figures. The intent is to preserve the orientations in the **closeup** position while possibly translating up/down and left/right and changing the scale.

1. position for [figure C](#), closeup with glucose molecules:

Display the glucose residues, both named BGC (**Note**: older versions of these PDB entries used the name GLC instead):

Command: [disp :bgc](#)

If you have changed the view from the **closeup** position, restore it:

Command: [reset closeup](#)

Scale and translate everything (avoiding rotation) so that the glucose molecules are visible, optionally adjust the clipping planes in the [Side View](#), and then save another position:

Command: **savepos closeup2**

Scaling can be done [with the mouse](#), or with the command **scale**, or by moving the small square in the **Side View**. Translations (up/down, left/right) can be performed [with the mouse](#) or with the command **move**. If you make a mistake, you can simply revert to the **closeup** position and start over.

2. position for [figure A](#), the two proteins side by side:

Again we wish to leave the orientation of the structures the same as in the **closeup** position, but separate them and zoom out so that all parts are visible. Restore the **closeup** position:

Command: **reset closeup**

Zoom out (scale down) and turn off clipping by unchecking the **Clip** option in the **Side View**. Scale and vertically translate the structures as needed to leave room for a title at the top and other text at the bottom. The proteins are still superimposed, in the middle of the window. Translate one structure to the left and the other to the right:

Command: **move x -25 models #0**

Command: **move x 30 models #1**

The exact translation distances needed will differ according to your starting position. When a satisfactory side-by-side view has been achieved, save the position:

Command: **savepos sidebyside**

Adding Labels and Saving Scenes

Chimera [scenes](#) are similar to positions but contain more information such as colors, display styles, and which atoms and labels are displayed. If scenes are saved and restored, it will not be necessary to manually display and undisplay the glucose residues and the appropriate labels between saving the different images. However, a drawback of scenes is that they increase session file size.

Start **2D Labels** (under **Tools... Utilities**) to annotate the images.

1. [figure B](#):

Command: **reset closeup**

Command: **~disp :bgc**

and then use **2D Labels** to label the residues. Click to start a label and type to enter its contents. Click the [color well](#) to set label color, in this case **black**. The appropriate font size depends on the size of the window; use what you think looks best. Each label can be repositioned by dragging. When done creating the labels for this view, save the scene:

Command: **scene sceneB save**

Next, hide all of these labels using the **Shown** checkboxes in the **2D Labels**

dialog.

2. [figure C](#):

Command: [reset](#) **closeup2**

Command: [disp](#) **:bgc**

Command: [scene](#) **sceneC save**

We will not make the window thumbnail-sized until right before saving the image.

3. [figure A](#):

Command: [reset](#) **sidebyside**

and then use [2D Labels](#) to add a **black** title above the structures and color-coded descriptions below them. The label color for the gold structure was first set to gold (by clicking the [color well](#) and entering the name **gold** into the [Color Editor](#)) and then darkened until suitably visible on the white background. The label color for the cyan structure was similarly derived from **cyan**. Symbols such as Greek letters can be inserted using the menu in the [2D Labels](#) dialog. When done creating the labels for this view, save the scene:

Command: [scene](#) **sceneA save**

Now all of the work other than saving the images is done. You should definitely [save the session](#) (see the **File** menu) at this point; it will include the labels, saved positions, and saved scenes. The images can be saved now or later, after quitting from Chimera, restarting Chimera, and [restoring the session](#).

Saving the Images

- [figure A](#):

1. as needed, [start Chimera](#) and [restore the session](#) (for example, using **File... Restore Session**)
2. restore the corresponding scene:

Command: [scene](#) **sceneA reset**

3. choose **File... Save Image**, specify file location, name and type (**PNG** was used), adjust size as desired (width 1200 pixels was used), and click **Save**

- [figure B](#):

1. restore the corresponding scene:

Command: [scene](#) **sceneB reset**

2. choose **File... Save Image**, specify file location, name and type (**PNG** was used), adjust size as desired (width 1200 pixels was used), and click **Save**

- [figure C](#):

1. restore the corresponding scene:

Command: [scene](#) **sceneC reset**

2. it was decided that this thumbnail image for the [Chimera Image Gallery](#) should be 110 by 87 pixels, a different aspect ratio than the window in the session. Resize the window to the planned thumbnail size, then scale as needed:

Command: [window size](#) **110 87**

Command: [scale](#) **1.2**

Of course, the scaling could have been done interactively with the mouse, but instead a scale factor was determined empirically (by resetting and trying different values) and written down so that the thumbnail could also be reproduced exactly from the session that had already been saved.

3. choose **File... Save Image**, specify file location, name and type (**PNG** was used), and click **Save**; in this case, the size was kept the same as the Chimera graphics window (width 110 pixels, height 87 pixels)