Rotation is angular addition

Rotation and translation
Superposition, structure comparison
Structure classification

Convention: angles are measured counter-clockwise.
Sum of angles formula

\[
\cos (\alpha+\beta) = \cos \alpha \cos \beta - \sin \alpha \sin \beta
\]

\[
\sin (\alpha+\beta) = \sin \alpha \cos \beta + \sin \beta \cos \alpha
\]

A rotation matrix

\[
\begin{align*}
x &= |r| \cos \alpha \\
y &= |r| \sin \alpha \\
x' &= |r| \cos (\alpha+\beta) \\
    &= |r|(\cos \alpha \cos \beta - \sin \alpha \sin \beta) \\
    &= (|r| \cos \alpha) \cos \beta - (|r| \sin \alpha) \sin \beta \\
    &= x \cos \beta - y \sin \beta \\
y' &= |r| \sin (\alpha+\beta) \\
    &= |r|(\sin \alpha \cos \beta + \sin \beta \cos \alpha) \\
    &= (|r| \sin \alpha) \cos \beta + (|r| \cos \alpha) \sin \beta \\
    &= y \cos \beta + x \sin \beta
\end{align*}
\]

\[
\begin{pmatrix}
x' \\
y'
\end{pmatrix} =
\begin{pmatrix}
\cos \beta & -\sin \beta \\
\sin \beta & \cos \beta
\end{pmatrix}
\begin{pmatrix}
r \cos \alpha \\
r \sin \alpha
\end{pmatrix} =
\begin{pmatrix}
\cos \beta & -\sin \beta \\
\sin \beta & \cos \beta
\end{pmatrix}
\begin{pmatrix}
x \\
y
\end{pmatrix}
\]

rotation matrix is the same for any \(r\), any \(\alpha\).
A rotation around a principle axis

The Z coordinate stays the same. X and Y change.

\[
R_z = \begin{pmatrix}
\cos \beta & -\sin \beta & 0 \\
\sin \beta & \cos \beta & 0 \\
0 & 0 & 1
\end{pmatrix}
\]

The Y coordinate stays the same. X and Z change.

\[
R_y = \begin{pmatrix}
\cos \gamma & 0 & \sin \gamma \\
0 & 1 & 0 \\
-\sin \gamma & 0 & \cos \gamma
\end{pmatrix}
\]

The X coordinate stays the same. Y and Z change.

\[
R_x = \begin{pmatrix}
1 & 0 & 0 \\
0 & \cos \alpha & -\sin \alpha \\
0 & \sin \alpha & \cos \alpha
\end{pmatrix}
\]

A 3D rotation matrix

Is the product of 2D rotation matrices.

\[
\begin{pmatrix}
\cos \beta & -\sin \beta & 0 \\
\sin \beta & \cos \beta & 0 \\
0 & 0 & 1
\end{pmatrix} \times \begin{pmatrix}
\cos \gamma & 0 & -\sin \gamma \\
0 & 1 & 0 \\
\sin \gamma & 0 & \cos \gamma
\end{pmatrix} = \begin{pmatrix}
\cos \beta \cos \gamma & \sin \beta \cos \gamma & -\sin \gamma \\
\sin \beta \cos \gamma & \cos \beta \cos \gamma & -\sin \beta \sin \gamma \\
\sin \gamma & 0 & \cos \gamma
\end{pmatrix}
\]

n around z-axis

Rotation around y-axis

3D rotation
What happens when you move the mouse:

1. Mouse sends mouse coordinates $(\Delta x, \Delta y)$ to the running program.

2. Rotation angles are calculated:
   \[ \theta_x = \Delta x \times \text{scale}, \quad \theta_y = \Delta y \times \text{scale} \]

3. Rotation matrices are calculated:
   \[
   R_x = \begin{pmatrix}
   1 & 0 & 0 \\
   0 & \cos \theta_x & -\sin \theta_x \\
   0 & \sin \theta_x & \cos \theta_x
   \end{pmatrix},
   R_y = \begin{pmatrix}
   \cos \theta_y & 0 & -\sin \theta_y \\
   0 & 1 & 0 \\
   \sin \theta_y & 0 & \cos \theta_y
   \end{pmatrix}
   \]

What happens when you move the mouse (cont'd):

4. New atom coordinates are calculated:
   \[
   \vec{r}' = R_y R_x \vec{r}
   \]

5. The scene is rendered using the new coordinates.
When multiplying matrices, the order matters.

\[
R_y R_x = \begin{pmatrix}
\cos \theta_y & 0 & -\sin \theta_y \\
0 & 1 & 0 \\
\sin \theta_y & 0 & \cos \theta_y
\end{pmatrix} \begin{pmatrix}
1 & 0 & 0 \\
0 & \cos \theta_x & -\sin \theta_x \\
0 & \sin \theta_x & \cos \theta_x
\end{pmatrix}
\]

\[
= \begin{pmatrix}
\cos \theta_y & -\sin \theta_x & \sin \theta_y \sin \theta_x & -\sin \theta_y \cos \theta_x \\
0 & \cos \theta_x & -\sin \theta_x \\
\sin \theta_y & \sin \theta_x & \cos \theta_x & \sin \theta_x \cos \theta_y
\end{pmatrix}
\]

This is the matrix if the X-rotation is first, then the Y-rotation.

Rotating in opposite order gives a different matrix

\[
R_x R_y = \begin{pmatrix}
1 & 0 & 0 \\
0 & \cos \theta_y & \sin \theta_y \\
0 & \sin \theta_y & \cos \theta_y
\end{pmatrix} \begin{pmatrix}
\cos \theta_x & 0 & -\sin \theta_x \\
0 & 1 & 0 \\
\sin \theta_x & \cos \theta_x & \sin \theta_x \cos \theta_y
\end{pmatrix}
\]

\[
= \begin{pmatrix}
\cos \theta_x & 0 & -\sin \theta_y \\
-\sin \theta_x & \sin \theta_y \cos \theta_x & -\sin \theta_y \cos \theta_y \\
\sin \theta_y \cos \theta_x & \sin \theta_x & \cos \theta_x \cos \theta_y
\end{pmatrix}
\]
Reversing the rotation

For the opposite rotation, flip the matrix.

This is the “transpose”

\[
\begin{pmatrix}
A & B \\
C & D
\end{pmatrix}^T = \begin{pmatrix}
A & C \\
B & D
\end{pmatrix}
\]

\[
\begin{pmatrix}
x' \\
y'
\end{pmatrix} = \begin{pmatrix}
\cos \beta & \sin \beta \\
-\sin \beta & \cos \beta
\end{pmatrix} \begin{pmatrix}
x \\
y
\end{pmatrix}
\]

The inverse matrix = The transposed matrix.

\[
\begin{pmatrix}
\cos \beta & \sin \beta \\
-\sin \beta & \cos \beta
\end{pmatrix} \begin{pmatrix}
\cos \beta & -\sin \beta \\
\sin \beta & \cos \beta
\end{pmatrix} = \begin{pmatrix} 1 & 0 \\
0 & 1 \end{pmatrix}
\]

NOTE: \( \cos \beta \cos \beta + \sin \beta \sin \beta = 1 \)

Example: rotation in 2 steps

Rotate the vector \( \mathbf{v} = (1.,2.,3.) \) around Z by 60°, then around Y by -60°

\[
\begin{pmatrix}
\cos 60^\circ & -\sin 60^\circ \\
\sin 60^\circ & \cos 60^\circ \\
0 & 0 \\
0 & 1
\end{pmatrix}
\begin{pmatrix}
1 \\
2 \\
3
\end{pmatrix} = \begin{pmatrix}
1(0.5) - 2(0.866) + 3(0) \\
1(0.866) + 2(0.5) + 3(0) \\
0 + 0 + 3(1)
\end{pmatrix} = \begin{pmatrix}
-1.232 \\
1.866 \\
3
\end{pmatrix}
\]

\[
\begin{pmatrix}
\cos 60^\circ & -\sin 60^\circ \\
0 & 1 \\
\sin 60^\circ & \cos 60^\circ \\
0 & 3
\end{pmatrix}
\begin{pmatrix}
-1.232 \\
1.866 \\
3
\end{pmatrix} = \begin{pmatrix}
-1.232(0.5) + 1.866(0) - 3(0.866) \\
-1.232(0) + 1.866(1) + 3(0) \\
-1.232(0.866) + 1.866(0) + 3(0.5)
\end{pmatrix} = \begin{pmatrix}
-3.214 \\
1.866 \\
0.433
\end{pmatrix}
\]
Right-handed 90° rotations:

90° rotation around 

\[
\begin{pmatrix}
1 & 0 & 0 \\
0 & 0 & -1 \\
0 & 1 & 0
\end{pmatrix}
\]  

helpful hint:
For a R-handed rotation, the -$sine$ is up and to the right of the +$sine$.

In class exercise: rotate a point

\((x, y, z) = (1., 4., 7.)\)

Rotate this point by 90° around the Z-axis

Then...

Rotate the new point by 90° around the Y-axis.

What are the new coordinates?
3D rotation conventions:

### Euler angles, $\alpha \beta \gamma$

<table>
<thead>
<tr>
<th>$z''$</th>
<th>$x$</th>
<th>$z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>$-\sin \gamma$</td>
<td>$0$</td>
</tr>
<tr>
<td>$\sin \gamma$</td>
<td>$\cos \gamma$</td>
<td>$0$</td>
</tr>
</tbody>
</table>

| $0$ | $0$ | $1$ |

| $\sin \gamma$ | $0$ | $\sin \beta$ |
| $\cos \gamma$ | $0$ | $\cos \beta$ |

| $\sin \beta$ | $\cos \beta$ | $1$ |

Each rotation is around a principle axis.

### Polar angles, $\phi \psi \kappa$

<table>
<thead>
<tr>
<th>$z''''$</th>
<th>$y'''$</th>
<th>$z''$</th>
<th>$-y'$</th>
<th>$-z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\cos \phi$</td>
<td>$-\sin \phi$</td>
<td>$0$</td>
<td>$\cos \phi$</td>
<td>$-\sin \phi$</td>
</tr>
<tr>
<td>$\sin \phi$</td>
<td>$\cos \phi$</td>
<td>$0$</td>
<td>$0$</td>
<td>$1$</td>
</tr>
</tbody>
</table>

| $\cos \kappa$ | $-\sin \kappa$ | $0$ | $\cos \kappa$ | $-\sin \kappa$ |
| $\sin \kappa$ | $\cos \kappa$ | $0$ | $0$ | $1$ |

| $0$ | $0$ | $1$ |

| $\sin \phi$ | $0$ | $\cos \phi$ |
| $0$ | $0$ | $1$ |

| $-\sin \phi$ | $0$ | $\cos \phi$ |
| $0$ | $0$ | $1$ |

Net rotation = $\kappa$, around an axis axis defined by $\phi$ and $\psi$

---

**Polar angles**

- $z$ = north pole
- $x$ = prime meridian
- $\phi$ = longitude
- $\psi$ = latitude

Rotation of $\kappa$ degrees around an axis axis located at $\phi$ degrees longitude and $\psi$ degrees latitude
Special properties of rotation matrices

- They are square, 2x2 or 3x3 (higher dimensions in principle)
- The product of any two rotation matrices is a rotation matrix.
- The inverse equals the transpose, \( R^{-1} = R^T \)
- Every row/column is a unit vector.
- Any two rows/columns are orthogonal vectors.
- The cross-product of any two rows equals the third.
- \(|x| = |Rx|\), where \( R \) is a rotation matrix.

Read more about rotation matrices at:
http://mathworld.wolfram.com/RotationMatrix.html

RMSD

*Root Mean Square Deviation* in superimposed coordinates is the standard measure of structural difference. Similar to “standard deviation” which is the square-root of the variance.  

\[
\sqrt{\frac{\sum_{i=1,N}(\bar{x}_i - \bar{y}_i)^2}{N}}
\]

Where \( x_i \) are the coordinates from molecule 1 and \( y_i \) are the *equivalent* coordinates from molecule 2.

*Which atoms are equivalent is based on an alignment.*
pseudo-pseudocode program for computing rmsd

```
sum = 0
N = 0
while <there is data to read> {
    /** Read coordinates for two aligned positions.
    read (x1,y1,z1), (x2,y2,z2)
    /** Compute the distance^2
    d = (x1-x2)^2 + (y1-y2)^2 + (z1-z2)^2
    /** sum it.
    sum = sum + d
    /** keep track of how many pairs there are.
    N = N + 1
}
/** Average and take square root.
rmstd = sqrt(sum/N)
```

Least squares superposition

Problem: find the rotation matrix, $M$, and a vector, $v$, that minimize the following quantity:

$$\sum_i \left| \vec{Mx_i} + \vec{v} - \vec{y_i} \right|^2$$

Where $x_i$ are the coordinates from one molecule and $y_i$ are the *equivalent* coordinates from another molecule.

*equivalent based on alignment*
Mapping structural equivalence =
aligning the sequence

Any position that is aligned is included in the
sum of squares.

4DFR:A  ISLIAALAVDRVILKDMAPWNLADLAVRIFKRNTLDKPVINGRHTWESIG-RPLPGKNI
1DFR:_  TAFLWAQNRNLILDKDHLPLRLHDLLHYFRAQTGVKIMVVGRTYESFPKRPILPERTNV

4DFR:A  ILSSQ-PGTDVRVTVKSVDEAIAC--GDEVPEIMVIGGRVYEQLPKQAQLYLVTHIDA
1DFR:_  VLTHQEDYQAQGAVVHDVAAVFAYAKQHLQQELVIAGGAQIFTAIFKDDVDTLLVTRLAG

4DFR:A  EVEGDTHFDPYEPDDWESVSEFHDGACWS--HSCSFKILERR
1DFR:_  SFEGDTKMIPLNNDDFTKVSSRTVEDF--NPAIHTYEVWQKK

Unaligned positions are not.

Least squares

At the position of best superposition, we have an approximate equality:

\[ M\overrightarrow{x}_i + \overrightarrow{v} \cong \overrightarrow{y}_i \]

First we eliminate \( \overrightarrow{v} \) by translating the center of mass of both molecules to the origin. Now we have:

\[ M\overrightarrow{x}_i' \cong \overrightarrow{y}_i' \]

We have one equation \((i)\) for each atom, \( M \) has 9 unknowns.

If there are more equations than unknowns, there is a unique solution. What is it? See the next two slides.
Least squares

Least squares solves a set of linear equations in the form:

\[
\begin{align*}
  x_{11}a_1 + x_{12}a_2 + \ldots + x_{1N}a_N &= y_1 \\
  x_{21}a_1 + x_{22}a_2 + \ldots + x_{2N}a_N &= y_2 \\
  \vdots & \vdots \\
  x_{M1}a_1 + x_{M2}a_2 + \ldots + x_{MN}a_N &= y_M
\end{align*}
\]

This is 'shorthand' notation for the equations.

Multiply both sides by transpose of x. "Squaring"

\[
  x^T x = (x^T x)^{-1} x^T y
\]

Summary: \( a = (x^T x)^{-1} x^T y \)
least-squares superimposed molecules

In class exercise: Superimpose 2 molecules by hand

• Open insightII
• Open two PDB files: **3shd.pdb, 1jzm.pdb**
• Display first chain ("A" usually) as "trace only"
• Choose a better color if necessary.
• Adjust the depth queuing if necessary.

• Superimpose the two molecules

Use the reference pages that follow.

<table>
<thead>
<tr>
<th>Do these pairs:</th>
<th>3sdh.pdb</th>
<th>1jzm.pdb</th>
<th>(hard)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3sdh.pdb</td>
<td>1h97.pdb</td>
<td>(harder)</td>
<td></td>
</tr>
<tr>
<td>3sdh.pdb</td>
<td>1phn.pdb</td>
<td>(hardest)</td>
<td></td>
</tr>
</tbody>
</table>
**InsightII reference:**

**getting a molecule:**
Molecule-->Get-->  
Getfile type=PDB  
Filename=

**Trace of chain A only:**
Molecule-->Display-->  
Display operation=Only  
Molec. attrib.=Atoms  
Atom set=Trace  
Molec. Spec=A*

left-click on a molecule to see the atom spec. Look for chain ID, if any.

**InsightII reference:**

**change depth queuing:**
hit **F12**  

**Color molecule:**
Molecule-->Color-->  
Attribute=Atoms  
Molec. Spec=  
Atom set=specified  
Color=<select>

**Delete objects:**

* type: Delete Object  
or Delete Object *
**InsightII reference:**

*How to superimpose two molecules:*

- Select first molecule, hit F10 and click on it.
  - Only first molecule will move. Move it to superimpose on 2nd molecule.
- Select "world" (hit F11)
  - Rotate both together to get a different view.
- Select either molecule (F10 + click)
  - Refine the superposition. Repeat until converged.

Note: you can connect to a molecule by typing: `connect XXX` where XXX is the "object name" for your molecule.
You can repeat and edit this or any command using the arrow keys.

**InsightII reference:**

*How to Save your work:*

File-->save_folder
- Don't check "replace" unless the file already exists.

*How to Save coordinates in PDB format:* This will create a file with just the atoms that are displayed. (i.e. first display all heavy atoms)

Molecule-->Put-->

```
put file type=PDB
Assembly/molec=
Filename=
transformed=checked
displayed=checked
reference_obj=checked
reference name=
```

Don't check "replace" unless the file already exists.

Command: `put molecule PDB XXX file.pdb transformed displayed reference YYY`
Superimpose 2 molecules automatically

Structure superposition programs have to do two things:

1. **Align the sequence**
2. **Minimize the RMSD**

Can't do first without the second.
Can't do the second without the first.

Often it is impossible to get a good sequence alignment, even though there is *structural homology*.

Remote homologs are more likely than close homologs

The existence of large numbers of remote homologs shows us that true structural similarity is hard to see in the amino acid sequence. Structural conservation is stronger than sequence conservation.

![Graph showing likelihood vs. percent identity for structural homologs](image)

**The "twilight zone"**
Example of structural homologs (analogs)

4DFR: Dihydrofolate reductase
1YAC: Octameric Hydrolase Of Unknown Specificity

5.9% sequence identity (best alignment)
1YAC structure solved without knowing function.
Alignment to 4DFR and others implies it is a hydrolase of some sort, probably uses NAD cofactors.

Example of structural homologs (analogs)

DHFR in yellow and orange. YAC in green and purple
sheets only
helices only
Structural alignment algorithms

Alignment algorithms create a one-to-one mapping of subset(s) of one sequence to subset(s) of another sequence.

**Structure-based alignment types:**

- **Geometric--intermolecular**
  - Algorithms may be do this by minimizing the intermolecular distances or root-mean-square deviation (rmsd) in superimposed alpha-carbon positions.

- **Geometric--intramolecular**
  - Algorithms minimize the difference between aligned contact maps or distance matrices. Intramolecular distances are used.

- **Non-Geometric**
  - Algorithms align structural properties, such as %buried, or secondary structure type, usually using dynamics programming (DP)

---

DALI: a intramolecular geometric structural alignment algorithm

DALI: (Distance matrix-based ALIgnment?)

Liisa Holm & Chris Sander

1. Generate a distance matrix for each protein
   
The distance matrix contains all pairwise distances. (symmetrical)
Shapes in distance matrices

In this slide “1” means close in space.

Aligning two distance matrices

Cut-and-paste alignment of distance matrices

Resulting sequence alignment
DALI algorithm

S is a score that is a maximum when the alignment is optimal.

\[ S = \sum_{i=1}^{L} \sum_{j=1}^{L} \phi^R(i, j) \]

where,

\[ \phi^R(i, j) = \theta^R - |d_{ij}^A - d_{ij}^B| \]

Phi is a constant (theta) minus the absolute difference of the two distance matrices, after they are aligned to each other.

Making the DALI pairs list

Each pair of 6x6’s corresponds to a gapped alignment.
DALI alignment: generate pairs list

extend pairs list
Can non-sequential alignments be found?

Non-sequential similarities exist.

Two different folds, superimposed with the loops removed.
SSAP alignment

A View is the set of all vectors from one residue.

Each residue has its own "View".

SSAP alignment: views

View for Template residue $i$

View for Target residue $j$

$i$ and $j$ must have similar backbone angles, otherwise the score is zero.

The difference between the two views is a measure of how similar the structures are, when viewed from $i$ and $j$. residue level score matrix
SSAP alignment: residue level score matrix

For each potential \( ij \) pair, we find the best DP alignment that includes it (i.e. global DP starting at 0,0 and ending at \( ij \), plus global DP starting at \( ij \) and ending at the lower right-hand corner).

SSAP alignment: summary matrix

The summary matrix is made up of DP scores for each \( ij \) position. Then a second round of DP is run, through the summary matrix. Final score is scaled to the range 0-100, where 100 is an identical match.
Two other servers for structure-based alignment

CE: Combinatorial Extension  
http://cl.sdsc.edu/ce.html

VAST:  
CE alignment

Structure Alignment - 2DRC:A Neighbors

<table>
<thead>
<tr>
<th>Structure</th>
<th>ID</th>
<th>Sequence</th>
<th>2DRC:A Neighbors</th>
</tr>
</thead>
<tbody>
<tr>
<td>2H0A</td>
<td>1/24</td>
<td>HELA</td>
<td>YEEHNSQKMYVR</td>
</tr>
<tr>
<td>2H0B</td>
<td>4/24</td>
<td>HELA</td>
<td>YEEHNSQKMYVR</td>
</tr>
<tr>
<td>2DRC:A</td>
<td>4/64</td>
<td>HEEVYTYVYV YEEHNSQKMYVR</td>
<td>ZM0100</td>
</tr>
<tr>
<td>2H0A</td>
<td>94/93</td>
<td>YEEHNSQKMYVR YEEHNSQKMYVR</td>
<td>ZM0100</td>
</tr>
</tbody>
</table>

CE alignment of 15 analogs

helix

sheet
Exercise: use CE to get structural homologs

Set your browser to http://cl.sdsc.edu/ce.html
Find structural alignments by selecting from ALL or REPRESENTATIVES from the PDB.
Submit your protein and chain. Or use 4dfr:A
Select 2 structures. Then hit: “Get alignment” (or use 4DFR and 1YAC)
Download as PDB file. Save it.
For use in InsightII, divide the file into 2, one for each protein.

Structure classification: The SCOP database

- Contains information about classification of protein structures and within that classification, their sequences
- http://scop.berkeley.edu
SCOP classification hierarchy

(1) class  --- global characteristics (no evolutionary relation)

(2) fold  --- Similar “topology”. Distant evolutionary cousins?

(3) superfamily

(4) family  --- Clear structural homology

(5) protein  --- Clear sequence homology

(6) species  --- functionally identical

unique sequences

Fold
domain

Most genes represent multidomain proteins

~40% of known structures (crystal, NMR) are multidomain proteins, but

**Most** of all proteins are multidomain. (~60% in unicellular organisms, ~90% in eukaryotes).

Where do my domains start and end??
Multidomain proteins

Domain boundaries can be seen as "weak" connections in the structure.

"Weak" means few contacts and few chain cross-overs.

In class exercise: Find the domains

Download 1akl from the PDB (or choose randomly, it doesn't matter)
Open in in InsightII.
Display it as a "trace"
Are there multiple domains? What residues lie at the domain boundary? Color domains differently.
protein classes

1. all $\alpha$ (126)
2. all $\beta$ (81)
3. $\alpha/\beta$ (87)
4. $\alpha+\beta$ (151)
5. multidomain (21)
6. membrane (21)
7. small (10)
8. coiled coil (4)
9. low-resolution (4)
10. peptides (61)
11. designed proteins (17)

Not true classes

possibly not complete, or erroneous

class: $\alpha/\beta$ proteins

Mainly parallel beta sheets (beta-alpha-beta units)

Folds:
TIM-barrel (22)
swivelling beta/beta/alpha domain (5)
spoIIa-like (2)
flavodoxin-like (10)
restriction endonuclease-like (2)
ribokinase-like (2)
chelatase-like (2)

Many folds have historical names. “TIM” barrel was first seen in TIM. These classifications are done by eye, mostly.
fold: flavodoxin-like
3 layers, α/β/α; parallel beta-sheet of 5 strand, order 21345

Superfamilies:
1. Catalase, C-terminal domain (1)
2. CheY-like (1)
3. Succinyl-CoA synthetase domains (1)
4. Flavoproteins (3)
5. Cobalamin (vitamin B12)-binding domain (1)
6. Ornithine decarboxylase N-terminal "wing" domain (1)
7. Cutinase-like (1)
8. Esterase/acetylhydrolase (2)
9. Formate/glycerate dehydrogenase catalytic domain-like (3)
10. Type II 3-dehydroquinate dehydratase (1)

Note the term: “layers”
These are not domains. No implication of structural independence.
Note how beta sheets are described: number of strands, order (N->C)

fold-level similarity
common topological features

catalase

flavodoxin

At the fold level, a common core of secondary structure is conserved. Outer secondary structure units may not be conserved.
Superfamily: Flavoproteins

- Flavodoxin-related (7)
- NADPH-cytochrome p450 reductase, N-terminal domain
- Quinone reductase

These molecules do not superimpose well, but side-by-side you can easily see the similar topology. Sec struct’s align 1-to-1, mostly.

Family: quinone reductases

- Proteins: binds FAD
  - quinone reductase type 2
  - NADPH quinone reductase

Different members of the same family superimpose well. At this level, a structure may be used as a molecular replacement model.
CATH

- Class
- Architecture
- Topology
- Homology

http://www.biochem.ucl.ac.uk/bsm/cath_new/index.html

TOPS topology cartoons

A simple way to draw a protein

- beta strand pointing up
- beta strand pointing down
- alpha helix
- connections

- A parallel beta sheet
- An anti-parallel beta sheet
TOPS topology cartoons

A right-handed $\beta\alpha\beta$ unit

A left-handed $\beta\alpha\beta$ unit (rarely seen)

TOPS and contact maps

A "contact map" for a $\beta\alpha\beta$ unit.
In class exercise: TOPS cartoon

Display 4dfr chain A in InsightII
Use Molecule-->render_secondary-->

helices
sheets
turns
loops

Draw a TOPS cartoon of this protein.