fconv Tutorial Part 2

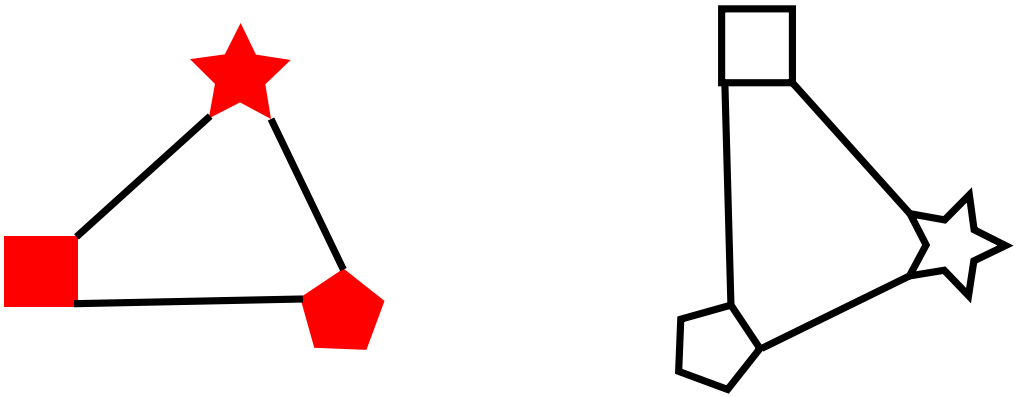
Gerd Neudert

Introduction to some basic fconv features based on version 1.08

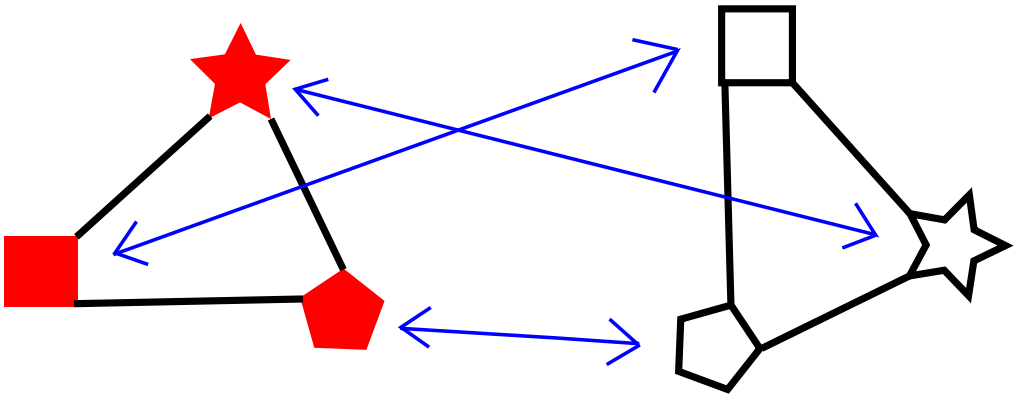
Welcome to the second part of fconv tutorials!

If you have not already done part one of the tutorials, I strongly suggest to do so before you go on!

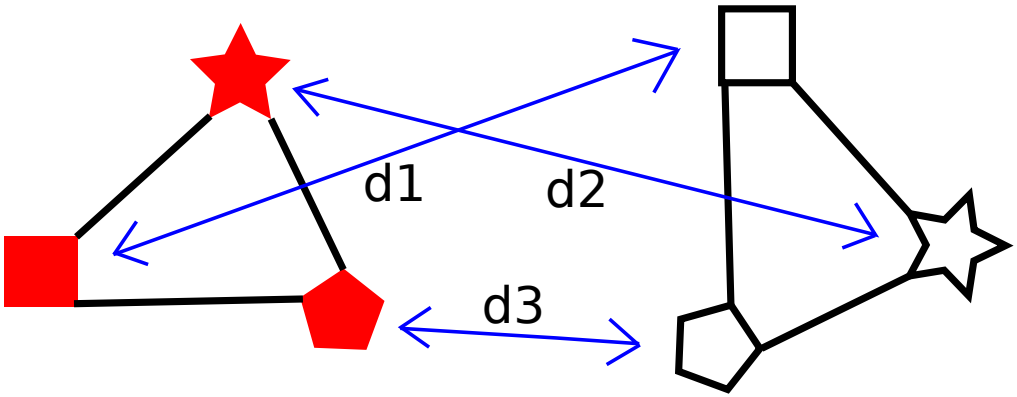
As promised, this time we will do some basic work concerning RMSD calculations, spatial superpositions and related topics. Again, only a subset of features will be introduced and you have to use the help function to discover more.
Before we start with some examples, I have to clarify some definitions.



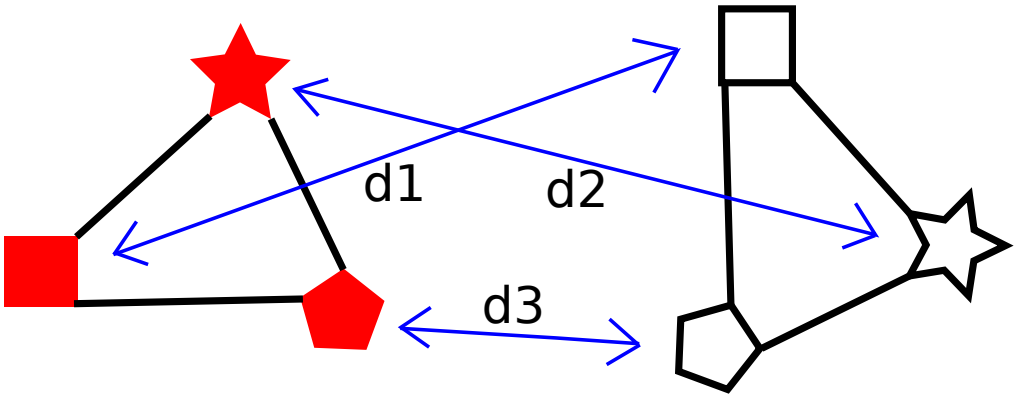
- *functional alignment* of 2 objects?



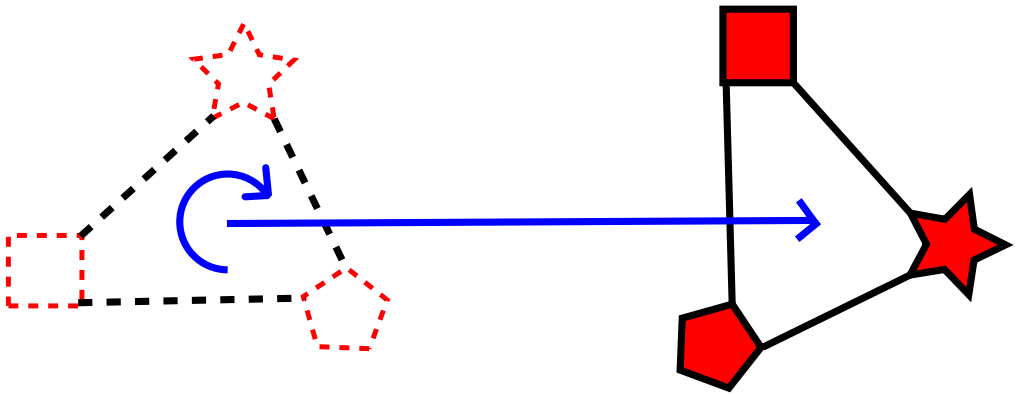
- *functional alignment* of 2 objects?
  - relation which element from A corresponds to which from B
  - no transformation of coordinates



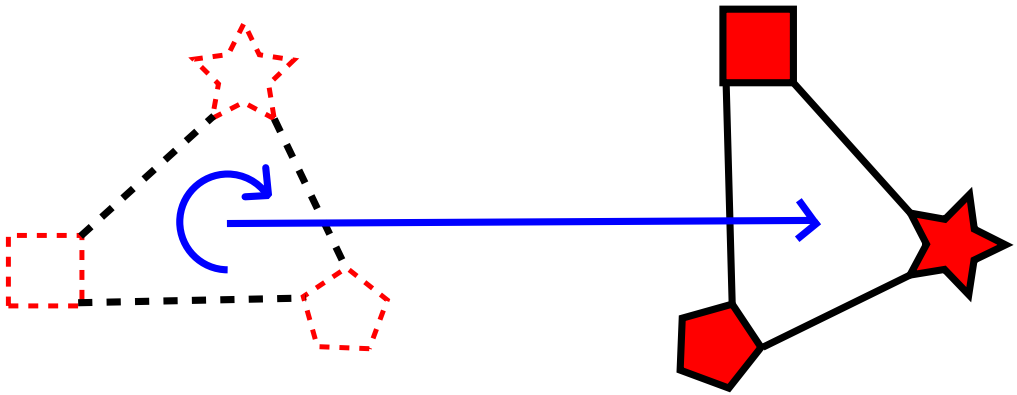
- *functional alignment* of 2 objects?
  - relation which element from A corresponds to which from B
  - no transformation of coordinates
- *RMSD* for a given functional alignment $= \sqrt{\frac{d1^2+d2^2+d3^2}{3}}$



- *functional alignment* of 2 objects?
  - relation which element from A corresponds to which from B
  - no transformation of coordinates
- *RMSD* for a given functional alignment $= \sqrt{\frac{d1^2+d2^2+d3^2}{3}}$
- *spatial alignment* of 2 objects?



- *functional alignment* of 2 objects?
  - relation which element from A corresponds to which from B
  - no transformation of coordinates
- *RMSD* for a given functional alignment $= \sqrt{\frac{d1^2+d2^2+d3^2}{3}}$
- *spatial alignment* of 2 objects?
  - coordinate transformation that minimizes RMSD
  - spatial alignment = superposition

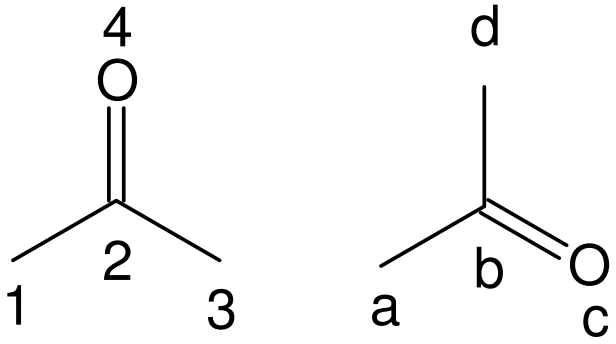


- *functional alignment* of 2 objects?
  - relation which element from A corresponds to which from B
  - no transformation of coordinates
- *RMSD* for a given functional alignment $= \sqrt{\frac{d1^2+d2^2+d3^2}{3}}$
- *spatial alignment* of 2 objects?
  - coordinate transformation that minimizes RMSD
  - spatial alignment = superposition
- *RMSD_opt* for a given functional alignment = RMSD after spatial alignment

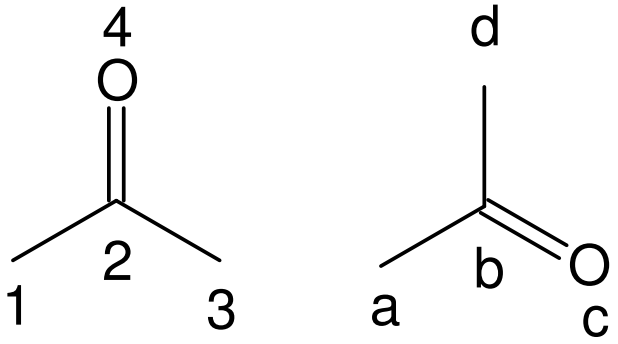
First, it is important to know that there is a big difference between calculating RMSDs (or superpositions) for small molecules on the one hand and for proteins on the other hand. We will start with the more important part, which is for small molecules (You can also apply this to proteins if you convert them to MOL2, but I do not suggest this!).

To calculate an RMSD value between two molecules, we need a functional alignment at first.

fconv uses a clique-based graphmatching (using Bron-Kerbosch) to determine functional alignments for small molecules.



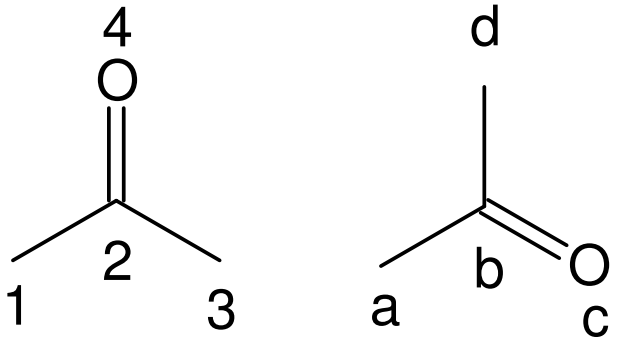
fconv uses a clique-based graphmatching (using Bron-Kerbosch) to determine functional alignments for small molecules.



- here: 2 equivalent alignments:

  1. alignment: 1a, 2b, 3d, 4c
  2. alignment: 1d, 2b, 3a, 4c

fconv uses a clique-based graphmatching (using Bron-Kerbosch) to determine functional alignments for small molecules.

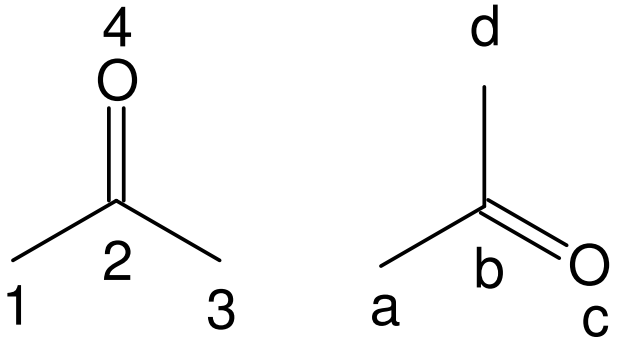


- here: 2 equivalent alignments:

  1. alignment: 1a, 2b, 3d, 4c
  2. alignment: 1d, 2b, 3a, 4c

- Prerequisite for atom-atom-match: same element + same connectivity
- Demanding identical atom types would reduce complexity, but also reduce stability (atom types may differ for same connectivity due to different geometry)

fconv uses a clique-based graphmatching (using Bron-Kerbosch) to determine functional alignments for small molecules.

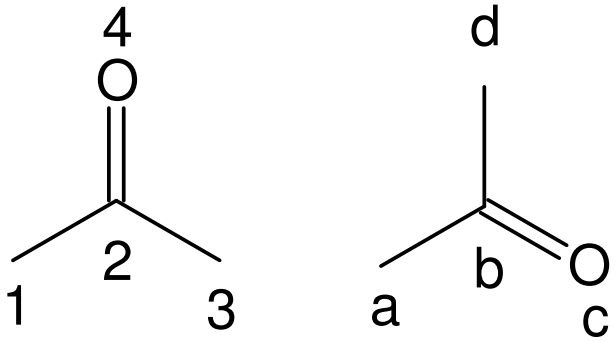


- here: 2 equivalent alignments:

  1. alignment: 1a, 2b, 3d, 4c
  2. alignment: 1d, 2b, 3a, 4c

- Prerequisite for atom-atom-match: same element + same connectivity
- Demanding identical atom types would reduce complexity, but also reduce stability (atom types may differ for same connectivity due to different geometry)

fconv uses a clique-based graphmatching (using Bron-Kerbosch) to determine functional alignments for small molecules.



- here: 2 equivalent alignments:

  1. alignment: 1a, 2b, 3d, 4c
  2. alignment: 1d, 2b, 3a, 4c

- Prerequisite for atom-atom-match: same element + same connectivity
- Demanding identical atom types would reduce complexity, but also reduce stability (atom types may differ for same connectivity due to different geometry)

```
fconv -rmsd A.mol2 --s=B.mol2
output: RMSD = float  RMSD_with_H = float  RMSD_opt = float
```

- alignments may result in different RMSD-values ⇒ fconv reports the best

```
fconv -l 3GUO.pdb
```

Please download 3GUO.pdb and type the command shown in the box. This should result in two files 3GUO_IPH_501_A.mol2 and 3GUO_IPH_501_B.mol2 corresponding to a phenol for each monomer of this dimer.

```
fconv -rmsd 3GUO_IPH_501_A.mol2 --s=3GUO_IPH_501_B.mol2
```

Please download 3GUO.pdb and type the command shown in the box. This should result in two files 3GUO_IPH_501_A.mol2 and 3GUO_IPH_501_B.mol2 corresponding to a phenol for each monomer of this dimer.

Type the new command shown in the box to get the RMSD between the two phenols. As we don't have any hydrogens, there is no difference between RMSD and RMSD_with_H. For phenol one might expect an RMSD_opt of value zero, but it's 0.01 Å. This may be due to crystallographic coordinate errors or even due to the float precision of the spatial alignment function.

```
fconv -oa 3GUO_IPH_501_A.mol2 --s=3GUO_IPH_501_B.mol2
```

Please download 3GUO.pdb and type the command shown in the box. This should result in two files 3GUO_IPH_501_A.mol2 and 3GUO_IPH_501_B.mol2 corresponding to a phenol for each monomer of this dimer.

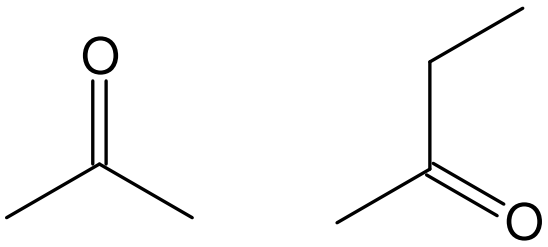
Type the new command shown in the box to get the RMSD between the two phenols. As we don't have any hydrogens, there is no difference between RMSD and RMSD_with_H. For phenol one might expect an RMSD_opt of value zero, but it's 0.01 Å. This may be due to crystallographic coordinate errors or even due to the float precision of the spatial alignment function.

If you want to superimpose 3GUO_IPH_501_A.mol2 on 3GUO_IPH_501_B.mol2 you can use the command shown above (This will overwrite 3GUO_IPH_501_A.mol2 with the new coordinates!).

```
fconv -rmsd 3GUO_IPH_501_A.mol2 --s=3GUO_IPH_501_B.mol2 ---d
```

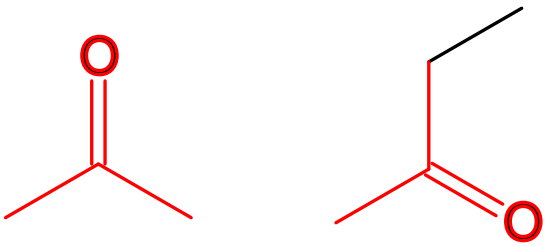
As shown in the box, you can also use the debug-flag `---d`. This will result in a list with the functional alignment used. You might notice, that there are two lists, one for RMSD and one for RMSD_opt. This is, because the best functional alignment for RMSD may differ from the best functional alignment for RMSD_opt (not in the case of phenol, but for other symmetric molecules this may happen).

If you use '-rmsd' on non-identical molecules, fconv will report an RMSD value of -1., indicating that no complete match was possible (-oa gives a warning). However you might be interested in the RMSD (or superposition) of the maximum common substructure.



- Search for maximum common substructure
- Bron-Kerbosch again (maximum common subgraph)

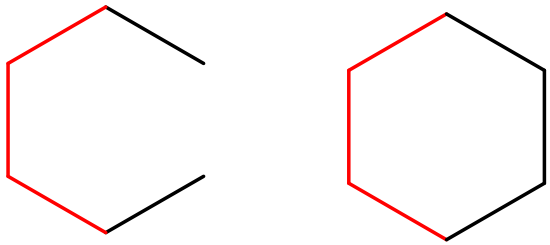
If you use '-rmsd' on non-identical molecules, fconv will report an RMSD value of -1., indicating that no complete match was possible (-oa gives a warning). However you might be interested in the RMSD (or superposition) of the maximum common substructure.



- Search for maximum common substructure
- Bron-Kerbosch again (maximum common subgraph)

```
fconv -rmsd2 A.mol2 --s=B.mol2
fconv -oa2 A.mol2 --s=B.mol2
```

If you use '-rmsd' on non-identical molecules, fconv will report an RMSD value of -1., indicating that no complete match was possible (-oa gives a warning). However you might be interested in the RMSD (or superposition) of the maximum common substructure.



- Search for maximum common substructure
- Bron-Kerbosch again (maximum common subgraph)

CAVE

- Possibly less atoms will match than you might expect, because information of shortest bond paths is used to reduce complexity!

```
fconv -l 2ZAS.pdb
```

Please download 2ZAS.pdb and type the command shown in the box. This should result in a file 2ZAS_1OH_460_A.mol2 corresponding to the ligand 4-(1-methyl-1-phenylethyl)phenol.

```
fconv -oa 3GUO_IPH_501_A.mol2 --s=2ZAS_1OH_460_A.mol2 ---d
```

Please download 2ZAS.pdb and type the command shown in the box. This should result in a file 2ZAS_1OH_460_A.mol2 corresponding to the ligand 4-(1-methyl-1-phenylethyl)phenol.

Use the command from the box and fconv will tell you 'found no alignment...'.

```
fconv -oa2 3GUO_IPH_501_A.mol2 --s=2ZAS_1OH_460_A.mol2 ---d
```

Please download 2ZAS.pdb and type the command shown in the box. This should result in a file 2ZAS_1OH_460_A.mol2 corresponding to the ligand 4-(1-methyl-1-phenylethyl)phenol.

Use the command from the box and fconv will tell you 'found no alignment...'.

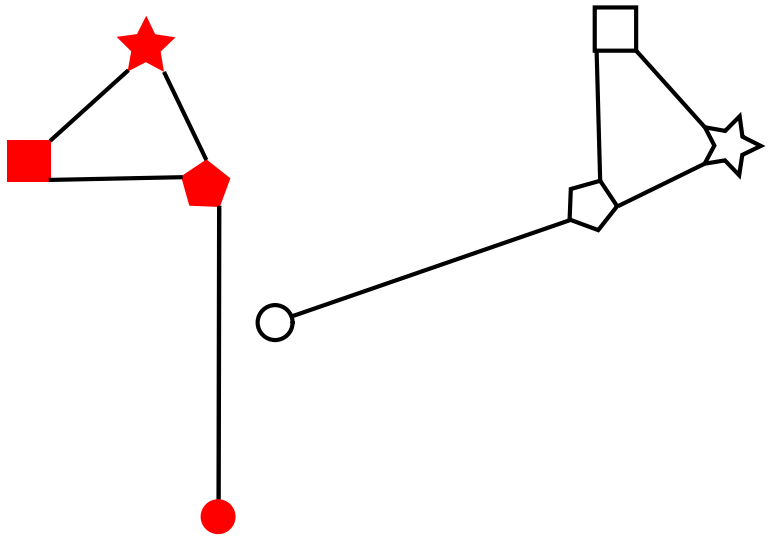
Now use -oa2, which will result in the desired substructure alignment (watch the results in e.g. Pymol). Again, `---d` lists the used functional alignment.

```
fconv -ss 2ZAS.pdb --s=3GUO_IPH_501_A.mol2
```

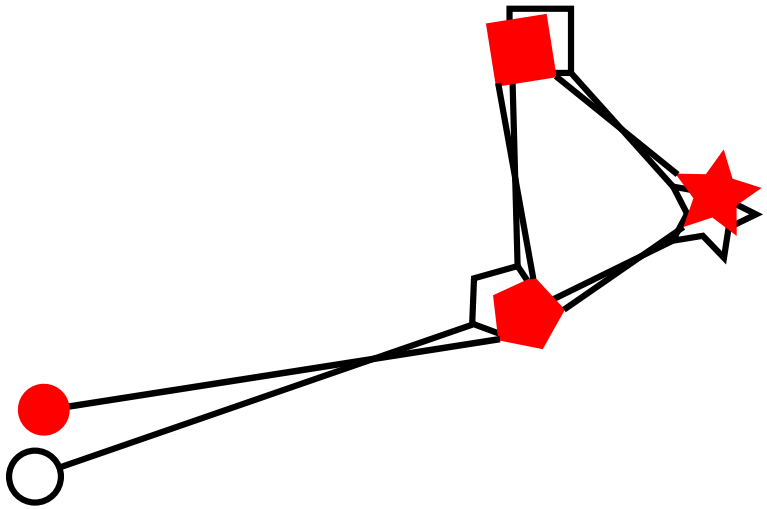
It is also possible to search for a substructure occurrence within a set of files. You can modify such a substructure search by several requirements for your query structure, but this is beyond the scope of this tutorial. Thus I will only present a very basic example (see 'fconv -h' for advanced stuff).
If you use the command shown in the box, fconv should report that the query substructure (our phenol) is found 2 times in ligand 1OH_460 from 2ZAS.pdb. It's found 2 times, because there are two possible functional alignments for the phenol substructure.

In the case of proteins (so for fconv in the case of PDB files) one is usually interested in the RMSD after superposition only. For modes `-rmsd` and `-oa`, the functional alignment is done by a sequence alignment (using Needleman-Wunsch). Only C-α carbons are used for calculation and even from those carbons only a subset is considered. This is due to the fact that one is usually interested in a best possible alignment of the conserved parts instead of a minimum overall RMSD. The next slide will explain this visually...

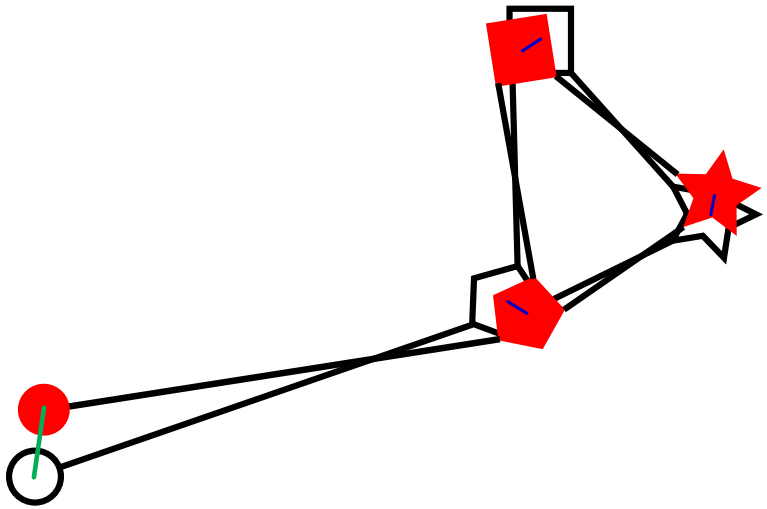
- 2 cycles:
  1. Superimpose C-α for functional alignment
  2. Remove pairs with distance $> \varepsilon$ (green) from functional alignment
  3. Superimpose with reduced set of C-α



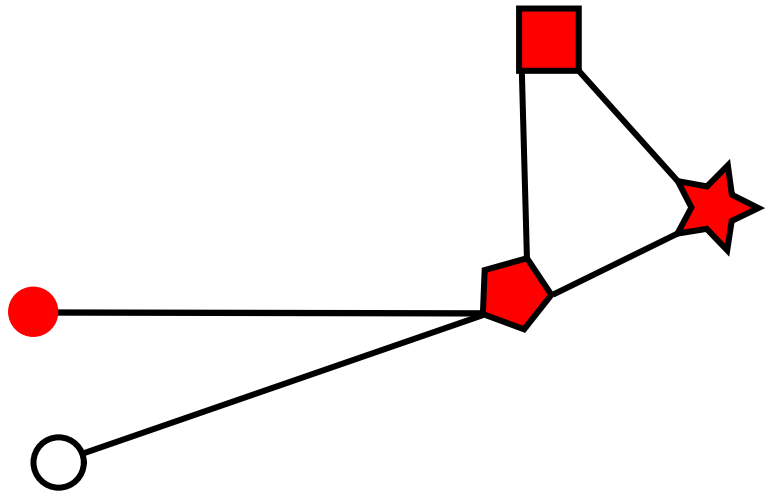
- 2 cycles:
  1. Superimpose C-α for functional alignment
  2. Remove pairs with distance $> \varepsilon$ (green) from functional alignment
  3. Superimpose with reduced set of C-α



- 2 cycles:
  1. Superimpose C-α for functional alignment
  2. Remove pairs with distance $> \varepsilon$ (green) from functional alignment
  3. Superimpose with reduced set of C-α



- 2 cycles:
  1. Superimpose C-α for functional alignment
  2. Remove pairs with distance $> \varepsilon$ (green) from functional alignment
  3. Superimpose with reduced set of C-α



```
fconv -rmsd 3LJJ.pdb --s=3M35.pdb
```

Download 3LJJ.pdb and 3M35.pdb and watch the two trypsin complexes in a molecule viewer. As you can see they are not aligned (spatially). If you apply the command from the box, fconv tells you, that 223 from 223 residues were matched during the sequence alignment and that all 223 corresponding α-carbons were used for spatial alignment, resulting in an RMSD of 0.22 Å.

```
fconv -oa 3LJJ.pdb --s=3M35.pdb
```

Download 3LJJ.pdb and 3M35.pdb and watch the two trypsin complexes in a molecule viewer. As you can see they are not aligned (spatially). If you apply the command from the box, fconv tells you, that 223 from 223 residues were matched during the sequence alignment and that all 223 corresponding α-carbons were used for spatial alignment, resulting in an RMSD of 0.22 Å.

Now use '-oa' to superimpose 3LJJ.pdb with 3M35.pdb and watch the result in a molecule viewer. The output is the same, but this time the coordinates of 3LJJ.pdb are transformed.

It is important to know, that the sequence alignment of fconv currently works without any substitution matrix. Thus it is only reliable for structures with a high sequence identity.
If you have only low identities or if a sequence alignment is not possible (e.g. you want to align polyalanine chains you obtained during crystallographic refinement steps), fconv can use geometric informations to match α-carbons. This is based on a graph matching, where interatomic distances and angles between C-α-carbons are used.
But also for structures with 100% sequence identity, this can be useful, as we will see in the next example...

```
fconv -oa 1A8E.pdb --s=1BP5.pdb --t=opt.pdb
```

Download 1A8E.pdb and 1BP5.pdb, open them in a molecule viewer and switch on cartoon view. The sequence identity of the two proteins is 100%, but they are crystallized in two different spacegroups and show a significant difference in their folding pattern. If you are using Pymol try to align them with this program (`align 1A8E, 1BP5`). Pymol reports an RMSD of 6.62 Å and as you can see, there is no part which you can identify as conserved in this visualization.

```
fconv -oa 1A8E.pdb --s=1BP5.pdb --t=opt.pdb
```

Download 1A8E.pdb and 1BP5.pdb, open them in a molecule viewer and switch on cartoon view. The sequence identity of the two proteins is 100%, but they are crystallized in two different spacegroups and show a significant difference in their folding pattern. If you are using Pymol try to align them with this program (`align 1A8E, 1BP5`). Pymol reports an RMSD of 6.62 Å and as you can see, there is no part which you can identify as conserved in this visualization.

Now use '-oa' (sequence-based) mode with fconv. This time don't overwrite the original file, but specify a target as shown in the box. Don't be confused by the low RMSD fconv reports, because only 15 C-α were used in the final superposition. If you watch the result in Pymol you will find out, that the fconv-solution is exactly the same as the Pymol-solution (as both are sequence-based).

```
fconv -oa2 1A8E.pdb --s=1BP5.pdb --t=opt2.pdb
```

Now apply the structure-based superposition as shown in the box. This will take more time (about 10 seconds on my PC) and fconv should report an RMSD of 0.22 Å for 55 matched α-carbons. If you watch the result in Pymol, you will see that the domain from residue 96 until residue 244 is now superimposed very well. Now it is much easier to analyze the differences between the two foldings.

- Part 3 will cover the extraction of binding pockets

Thanks go out to...

- Prof. Dr. Gerhard Klebe
- fconv bug reporters