fconv Tutorial Part 3

Gerd Neudert

Introduction to some basic fconv features based on version 1.08

Welcome to the third 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!

We will start with simple ligand-based extraction of binding pockets. Second, I will give a short theoretic background for automatic pocket detection, as this is necessary to understand the parameters. Finally, we will end with an example for the automatic detection.

```
fconv -pl 1CIL.pdb
```

Download 1CIL.pdb and use the command shown in the box to obtain a 6 Å pocket around the ligand (ETS).

```
fconv -pl 1CIL.pdb --r=3.0
```

Download 1CIL.pdb and use the command shown in the box to obtain a 6 Å pocket around the ligand (ETS).

To change the default cut-radius use `--r` from Options2, as shown in the box (please always watch your results in e.g. Pymol).

```
fconv -pl 1CIL.pdb --r=3.0 --f=p
```

Download 1CIL.pdb and use the command shown in the box to obtain a 6 Å pocket around the ligand (ETS).

To change the default cut-radius use `--r` from Options2, as shown in the box (please always watch your results in e.g. Pymol).

If you want to have your pocket in PDB format use `--f=p` from Options2.

```
fconv -me 1CIL.pdb
fconv -pl 1CIL.pdb --s=1CIL_metals.mol2 --r=3.0
```

Of course, you can supply the ligand separately. If you are e.g. interested in the residues around the zinc ion, you can extract this metal by the first command shown in the box. Subsequently, you can use the second command to obtain the residues within 3 Å around the zinc ion.

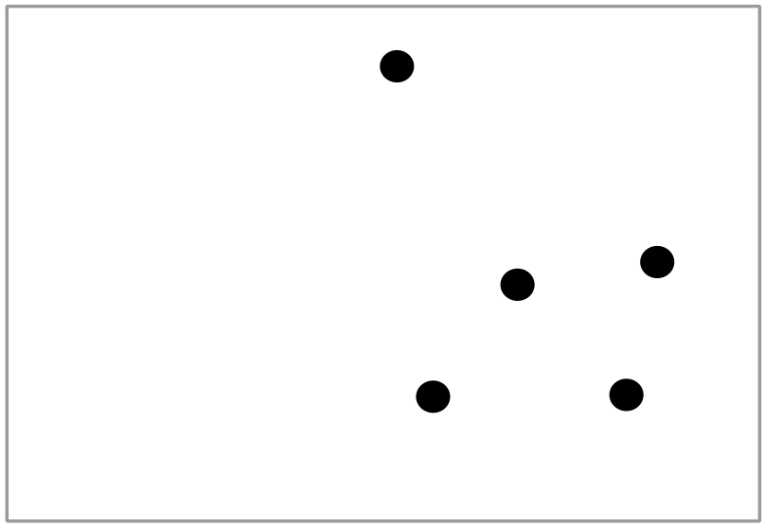
```
fconv -me 1CIL.pdb
fconv -pl 1CIL.pdb --s=1CIL_metals.mol2 --r=3.0
fconv -pp 1CIL.pdb --s=1CIL_metals.mol2 --r=3.0
```

Of course, you can supply the ligand separately. If you are e.g. interested in the residues around the zinc ion, you can extract this metal by the first command shown in the box. Subsequently, you can use the second command to obtain the residues within 3 Å around the zinc ion.

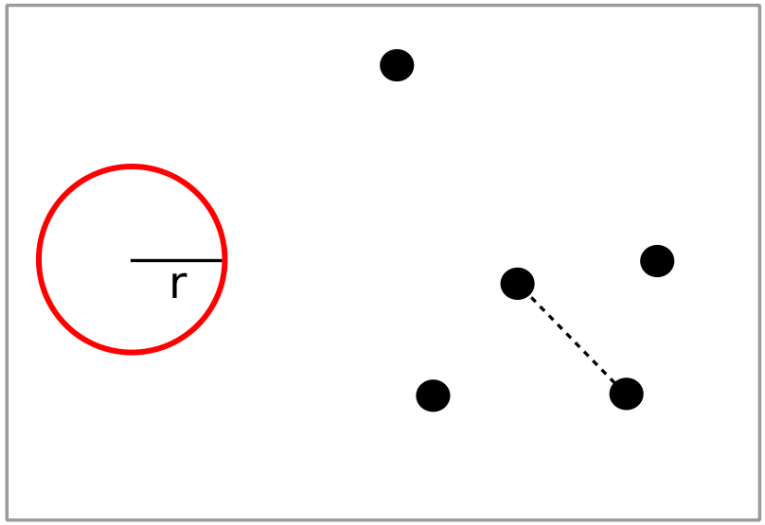
If you are not interested in complete residues, but only those atoms in the specified range you can use `-pp` instead of `-pl`, as shown in the third command.

fconv has two completely different methods for automatic cavity detection. The first one is grid-based and available by -pa from Options1. Although it's very fast, I do not suggest this method, as it depends on the orientation of the protein. Please use the help function, if you are interested. In this tutorial I will only introduce the second method, which is triangulation-based.

In fact, you don't have to know something about triangulations, but what you need is a basic understanding about so called α-shapes. A regular triangulation is just the method fconv uses to determine such α-shapes.
If you read the fconv-help concerning -pa2 from Options1 for PDB files, fconv tells you it uses two different α-levels, which can be set by the user.
On the next slide I will give a formal introduction to α-shapes, followed by a visual example which clarifies the use for cavity detection...

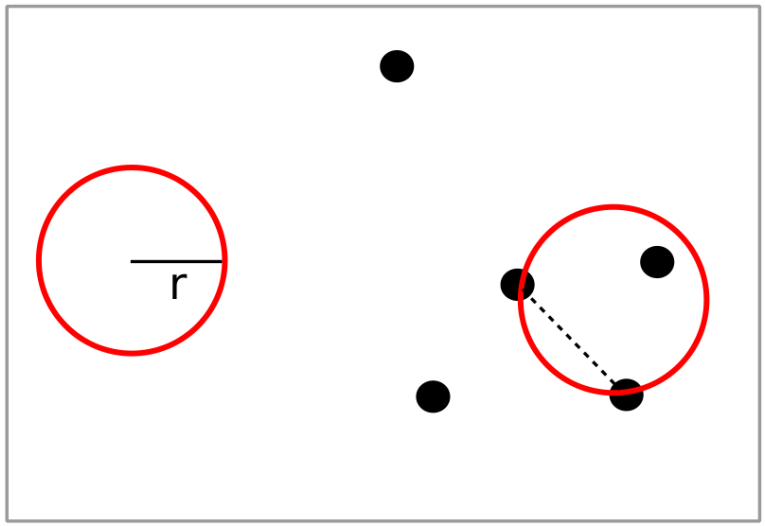


I use a 2D example for better understanding. So what you see is a set of 5 points in space. We can determine an α-shape for α-values from 0 to ∞ (the α-levels). A level of zero leads to the point set itself as the 0-shape, while a α-level $= \infty$ results in the convex hull as ∞-shape for the point set.
Now lets see how the shape is defined/constructed...



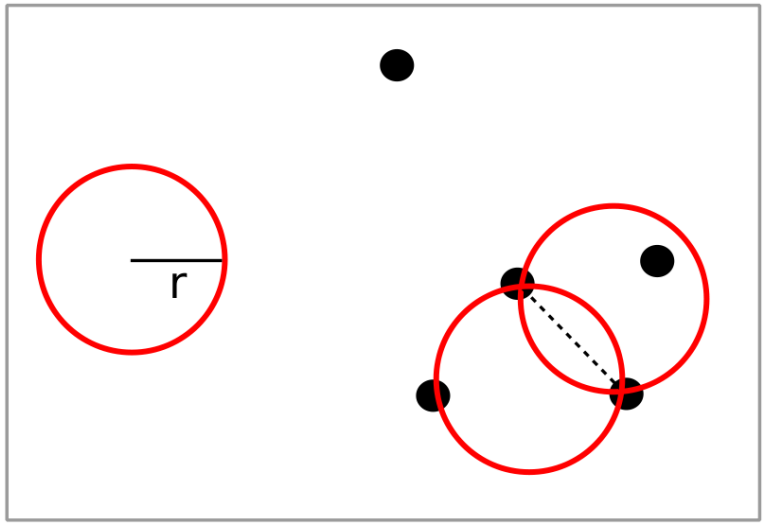
- r = α-level $\Rightarrow$ α-circle
- do not connect if other points in or on both α-circles
- do not connect if distance $>$2r

The chosen α-level corresponds to the radius of an α-circle (α-sphere in 3D)



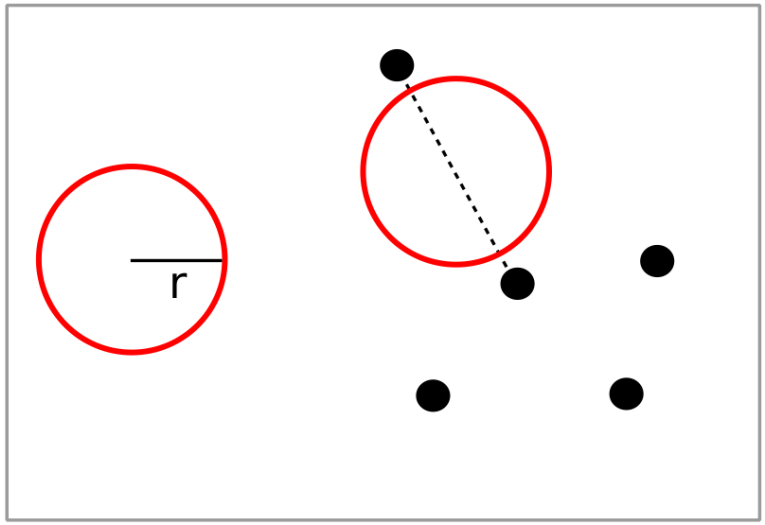
- r = α-level $\Rightarrow$ α-circle
- do not connect if other points in or on both α-circles
- do not connect if distance $>$2r

For each subset of two points (3 points in 3D), there are two possibilities to place the α-circle on both points.



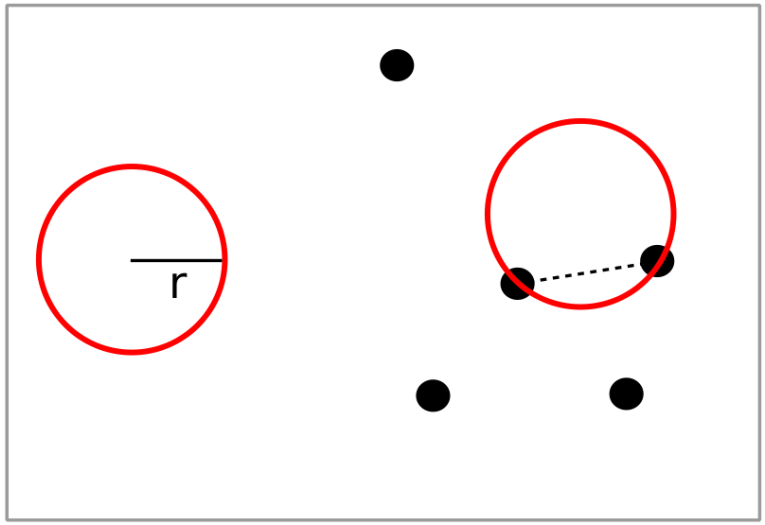
- r = α-level $\Rightarrow$ α-circle
- do not connect if other points in or on both α-circles
- do not connect if distance >2r

For each subset of two points (3 points in 3D), there are two possibilities to place the α-circle on both points.
We connect the two points by an edge, if at least one circle does not contain or touch any other point (not the case here).



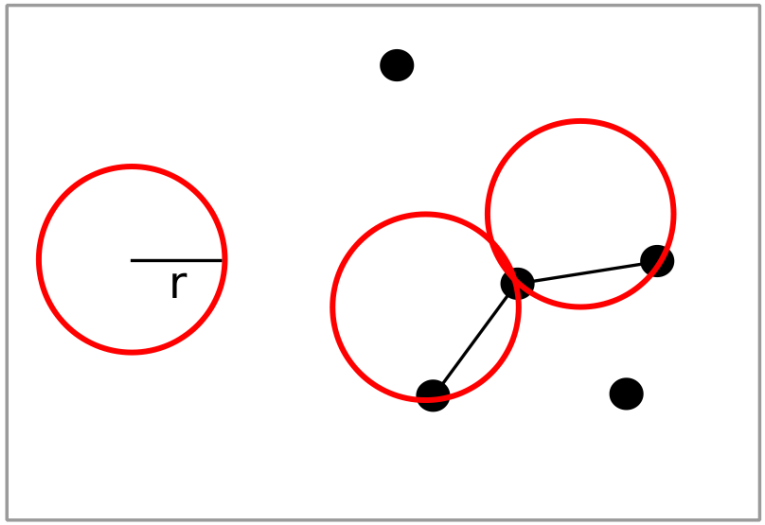
- r = α-level $\Rightarrow$ α-circle
- do not connect if other points in or on both α-circles
- do not connect if distance $>$2r

We do not connect two points, if we can't place the α-circle on them.



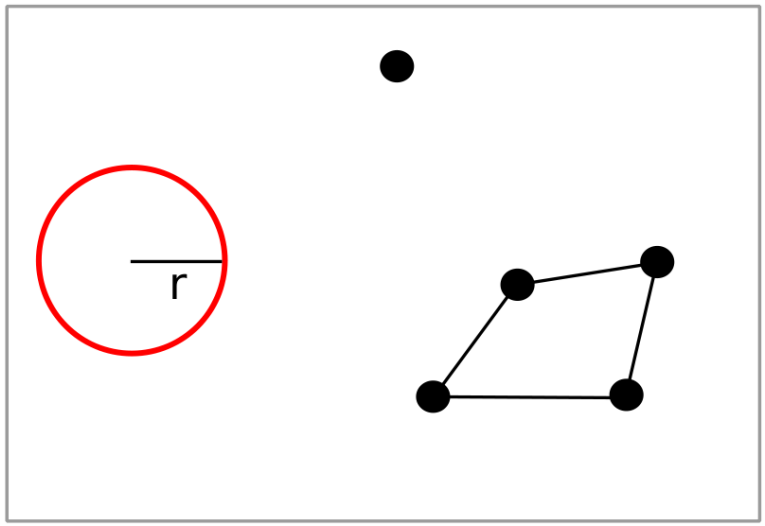
- r = α-level $\Rightarrow$ α-circle
- do not connect if other points in or on both α-circles
- do not connect if distance $>$2r

Here is the first pair we have to connect, because no other point lies within the attached α-circle.



- r = α-level $\Rightarrow$ α-circle
- do not connect if other points in or on both α-circles
- do not connect if distance $>$2r

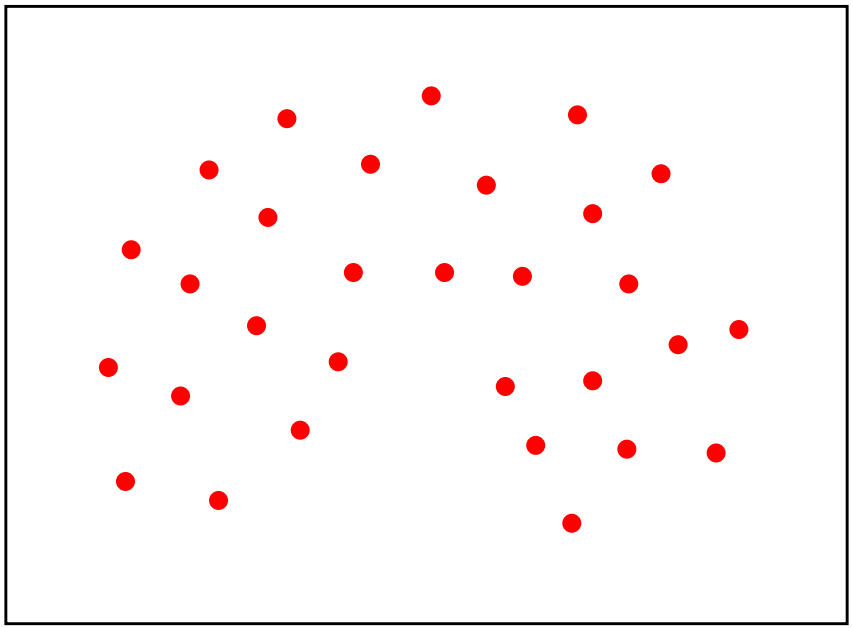
Here comes the next connection.



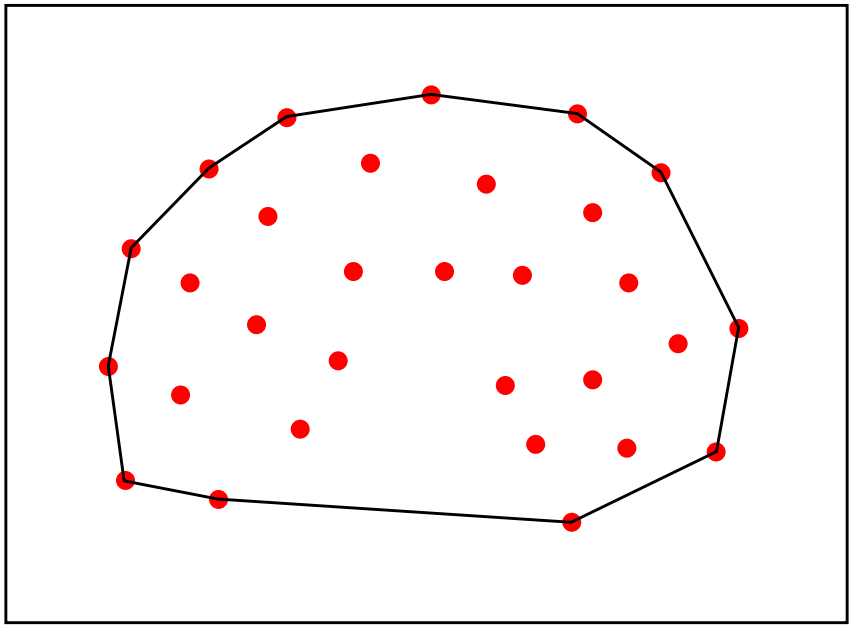
- r = α-level $\Rightarrow$ α-circle
- do not connect if other points in or on both α-circles
- do not connect if distance $>$2r

Finally, after testing all pairs, we got the resulting α-shape as shown above. In 2D an α-shape can consist of points, edges and triangles. In 3D there are also tetrahedra possible.

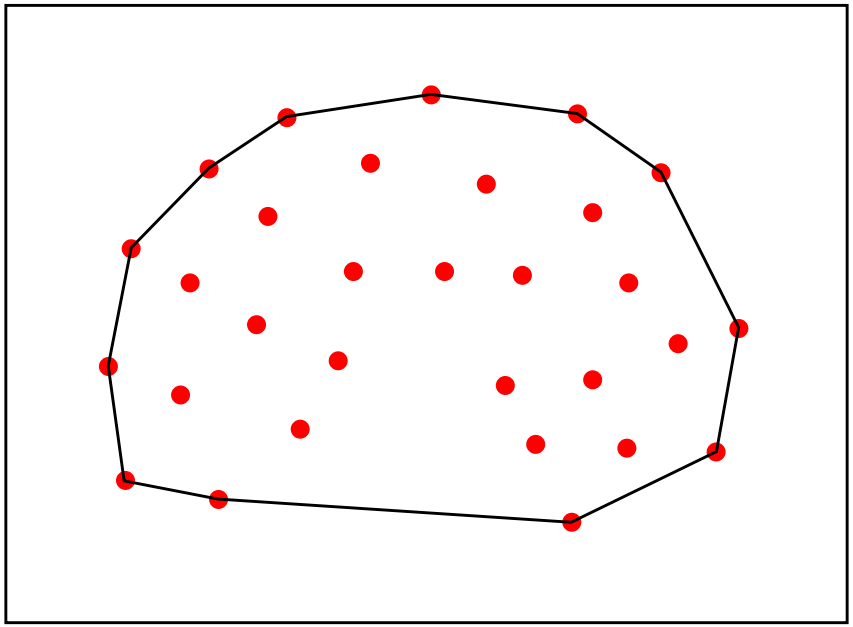
Now we will have a look on another 2D example with a higher number of points, which illustrates the use for cavity detection. . .



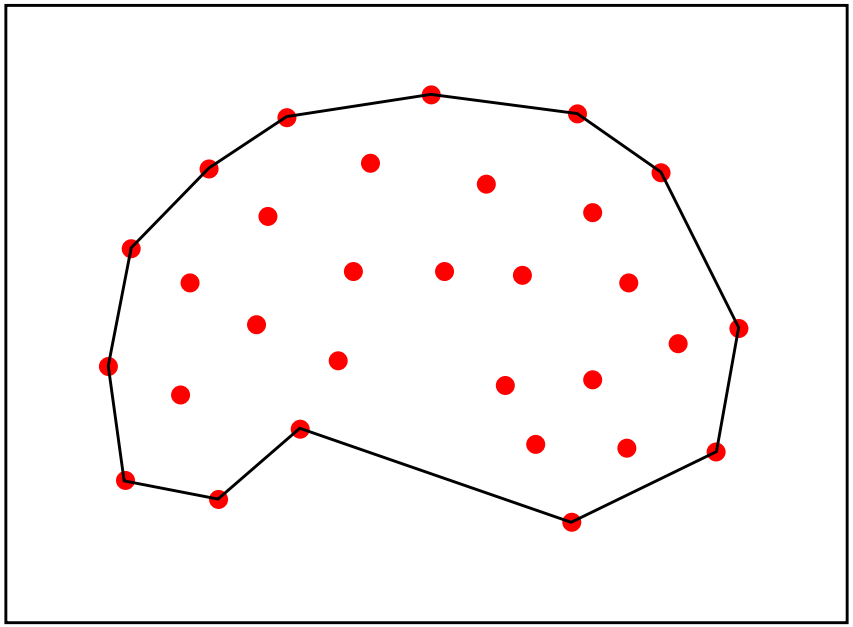
We start with a point cloud (corresponding to an α-level of zero) . . .



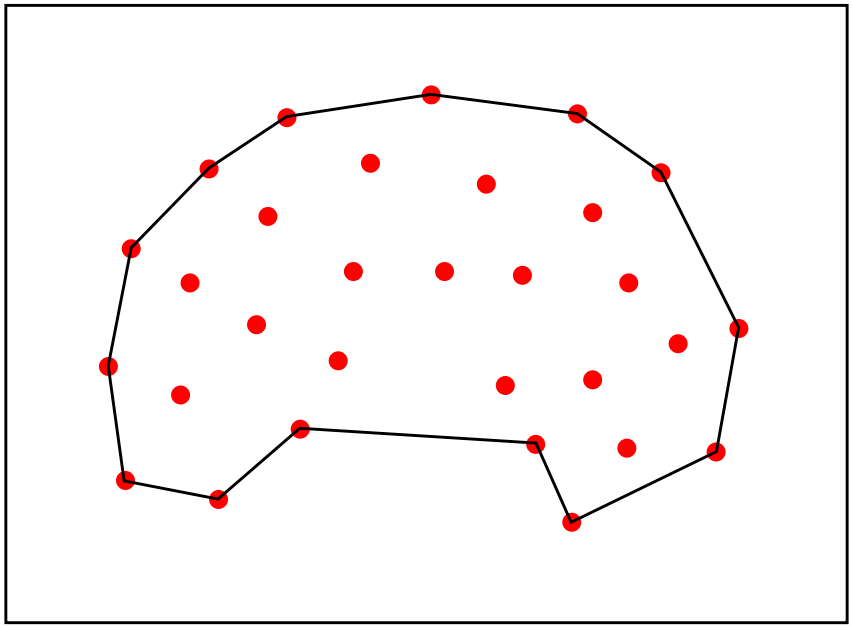
For a very high α-level we got the convex hull ...



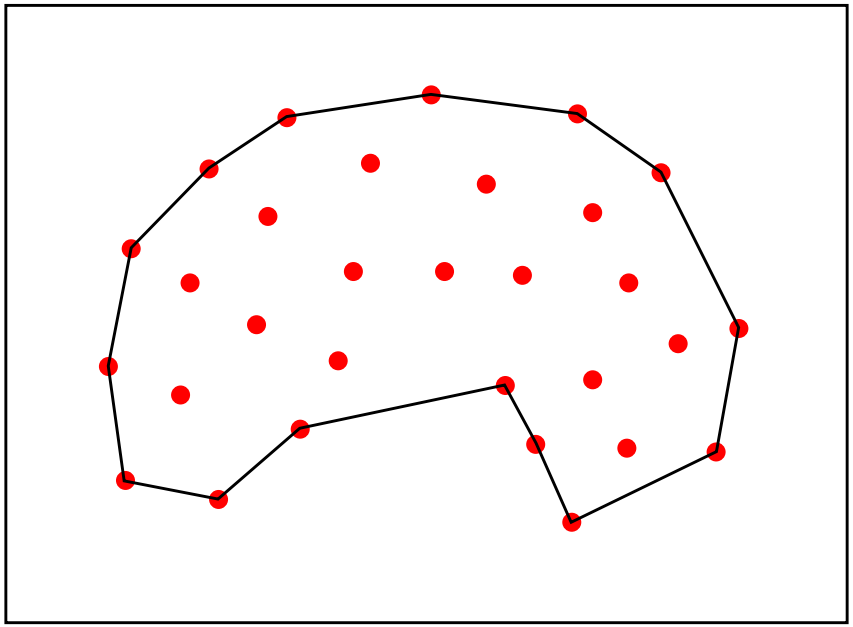
Now watch the α-shape as it changes with a step-wise decrease of the α-level on the next 10 slides . . .



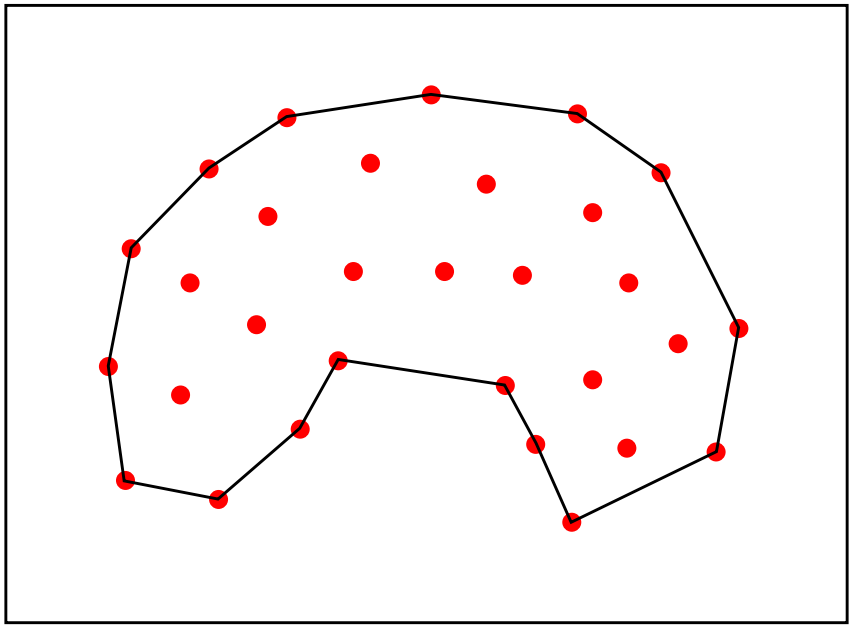
Now watch the α-shape as it changes with a step-wise decrease of the α-level on the next 9 slides . . .



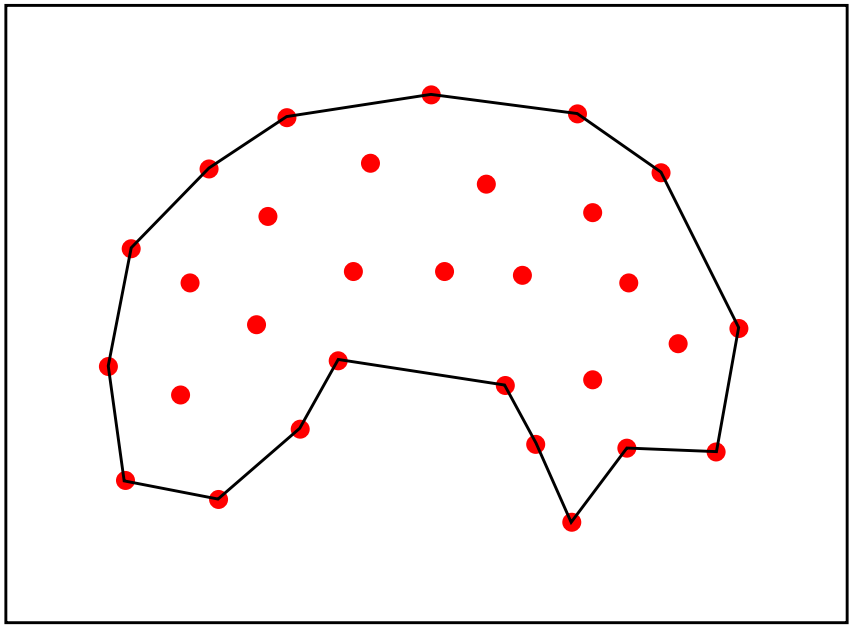
Now watch the α-shape as it changes with a step-wise decrease of the α-level on the next 8 slides . . .



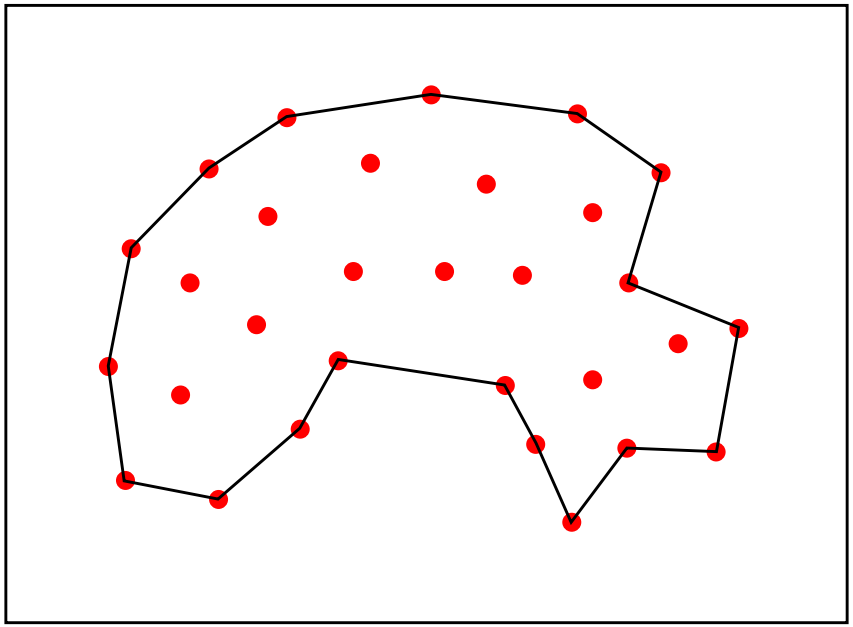
Now watch the α-shape as it changes with a step-wise decrease of the α-level on the next 7 slides . . .



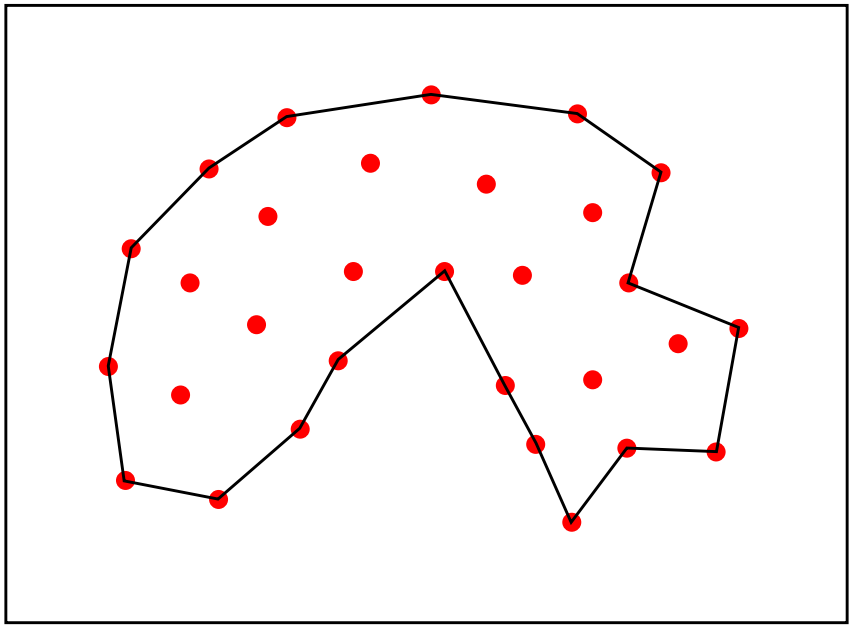
Now watch the α-shape as it changes with a step-wise decrease of the α-level on the next 6 slides . . .



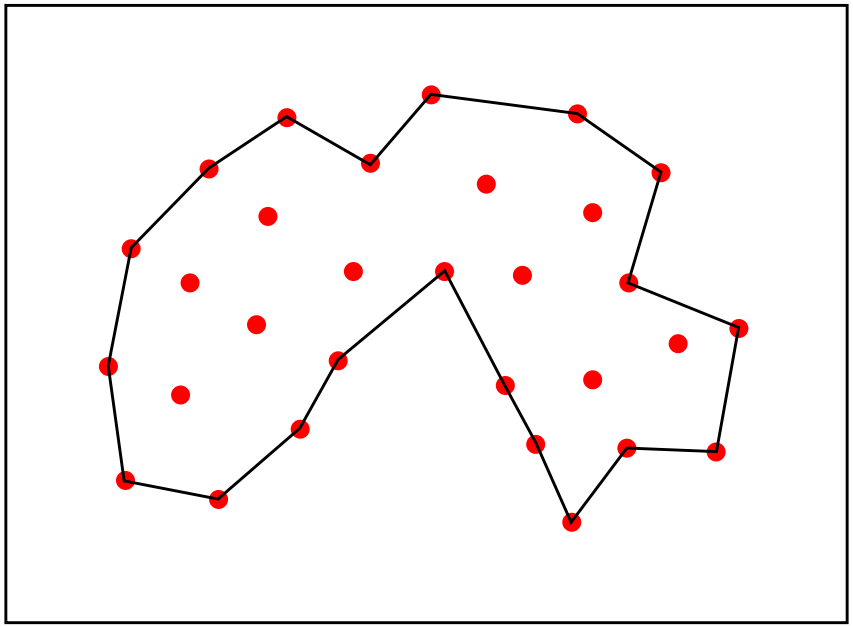
Now watch the α-shape as it changes with a step-wise decrease of the α-level on the next 5 slides . . .



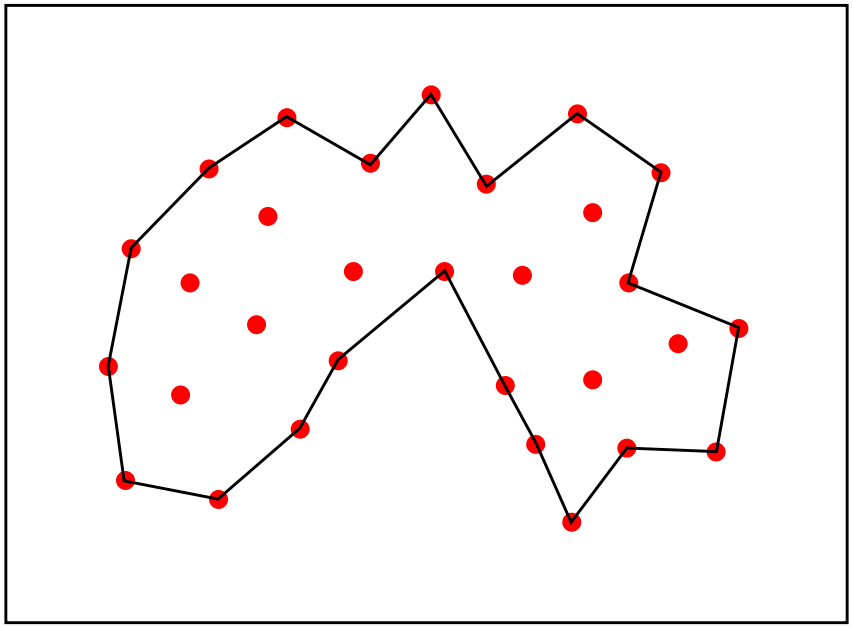
Now watch the α-shape as it changes with a step-wise decrease of the α-level on the next 4 slides . . .



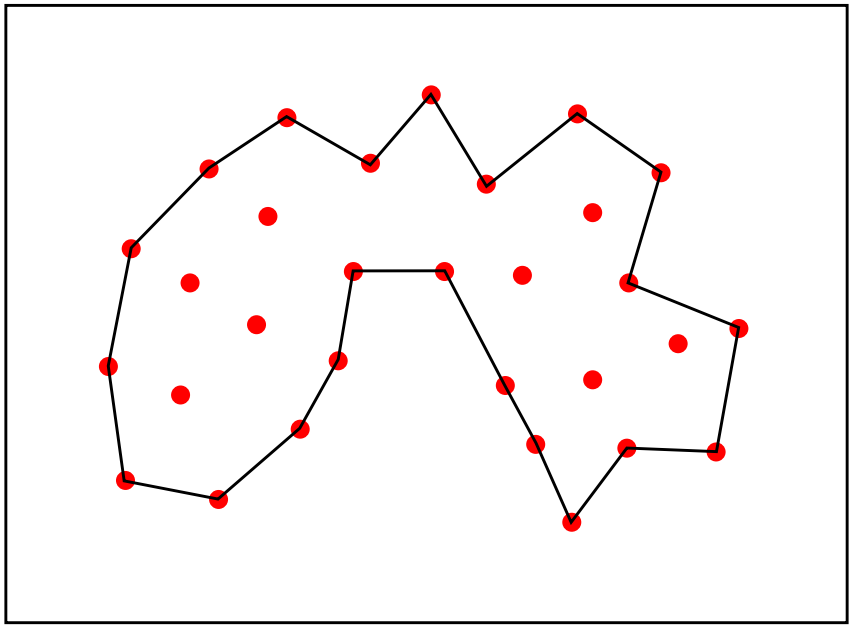
Now watch the α-shape as it changes with a step-wise decrease of the α-level on the next 3 slides . . .



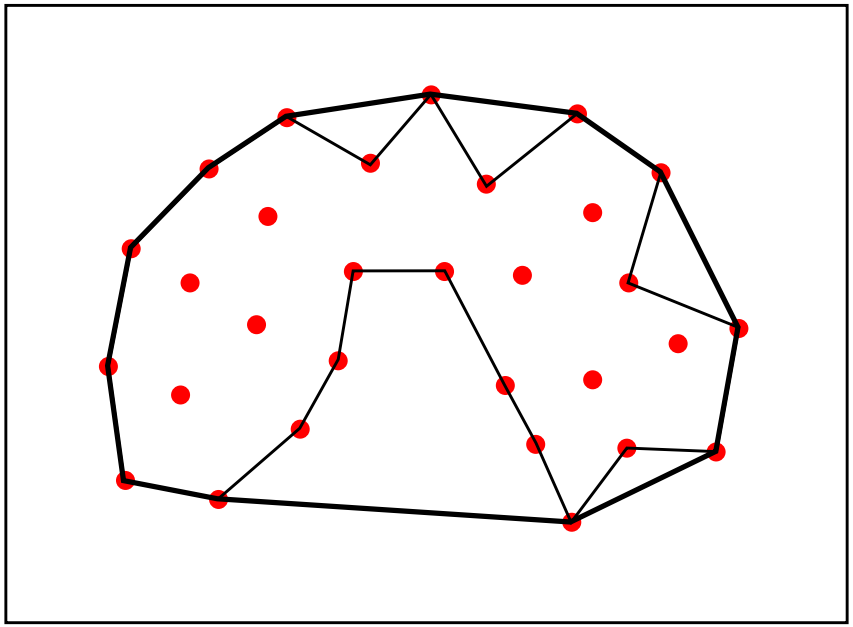
Now watch the α-shape as it changes with a step-wise decrease of the α-level on the next 2 slides . . .



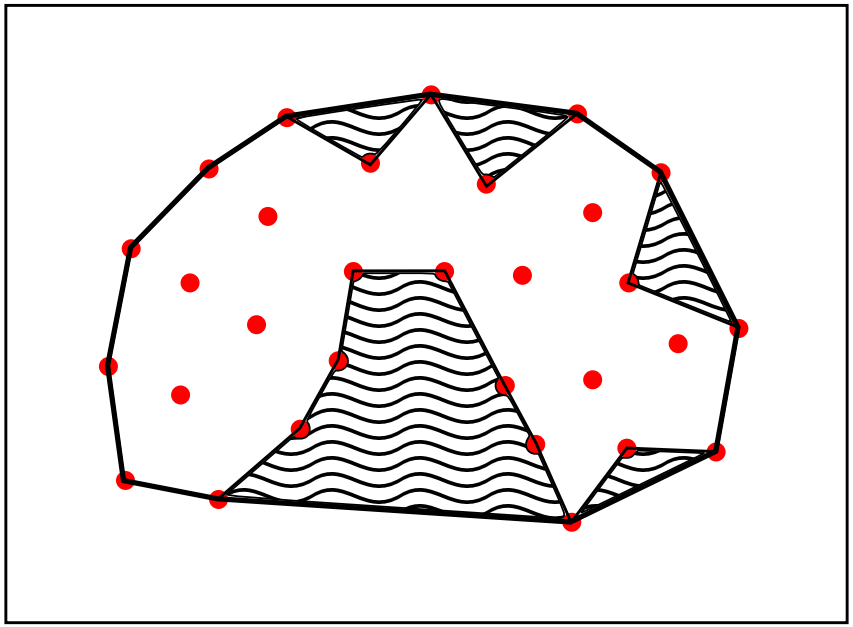
Now watch the α-shape as it changes with a step-wise decrease of the α-level on the next slide . . .



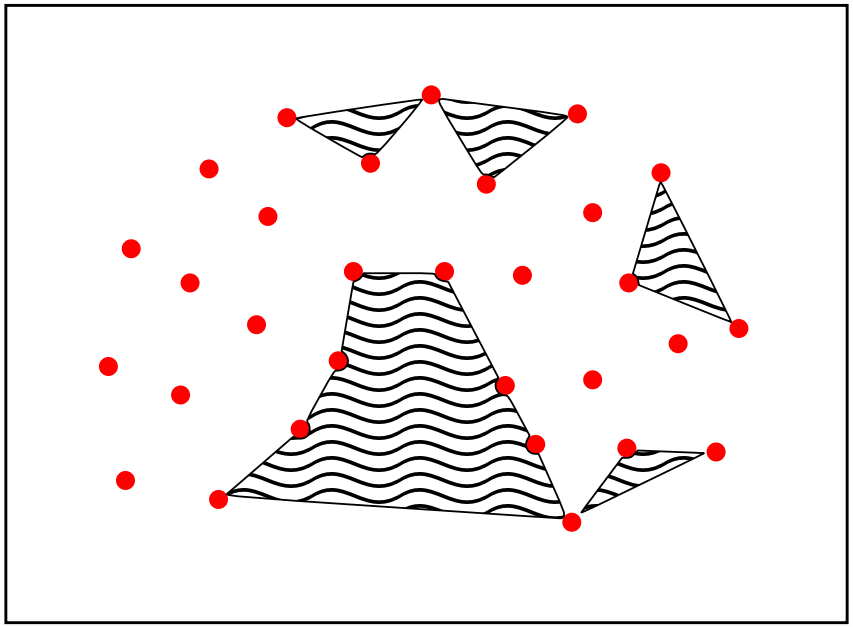
Now let us define this level as our lower α-level. We can choose a high α-level as our upper level...



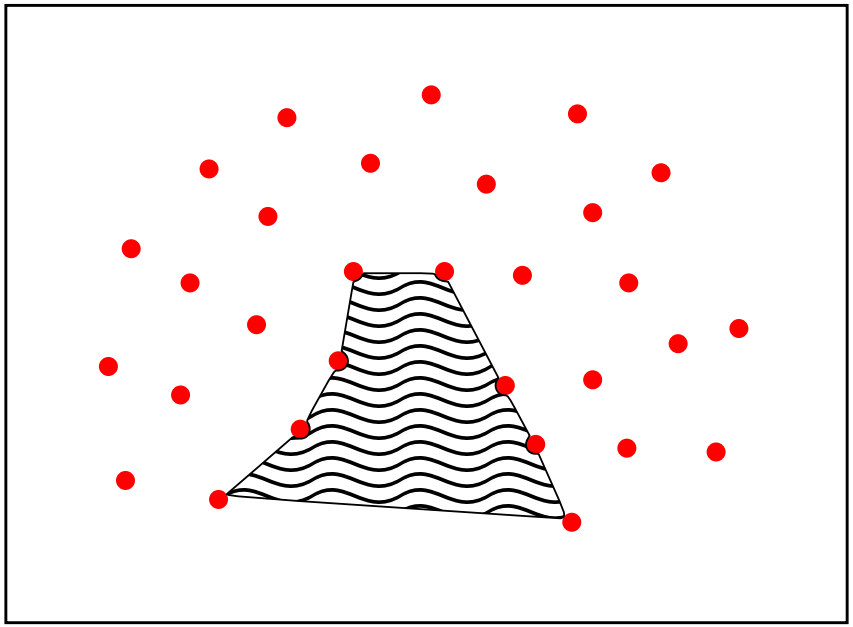
Now let us define this level as our lower α-level. We can choose a high α-level as our upper level... e.g. the convex hull.



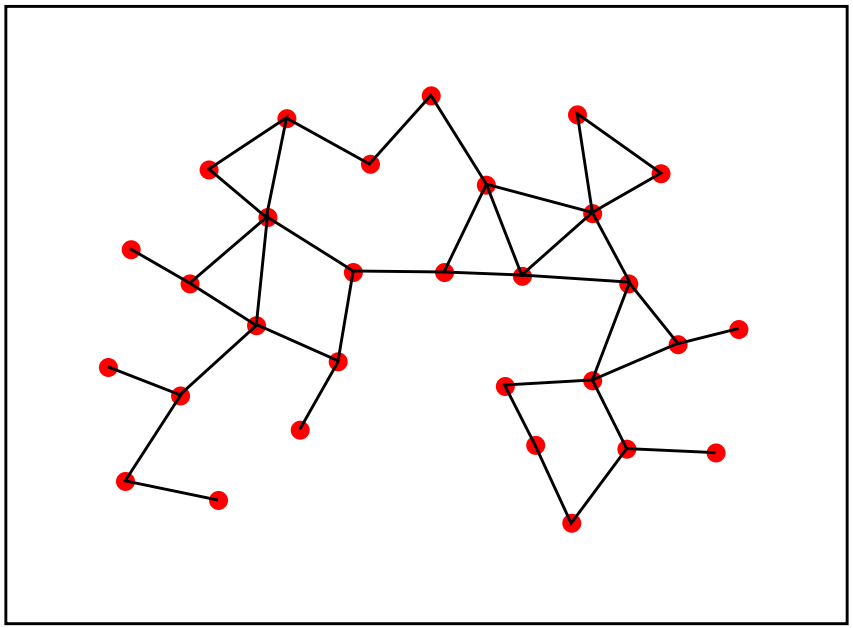
If we cluster the space between those two α-shapes we obtain cavities.



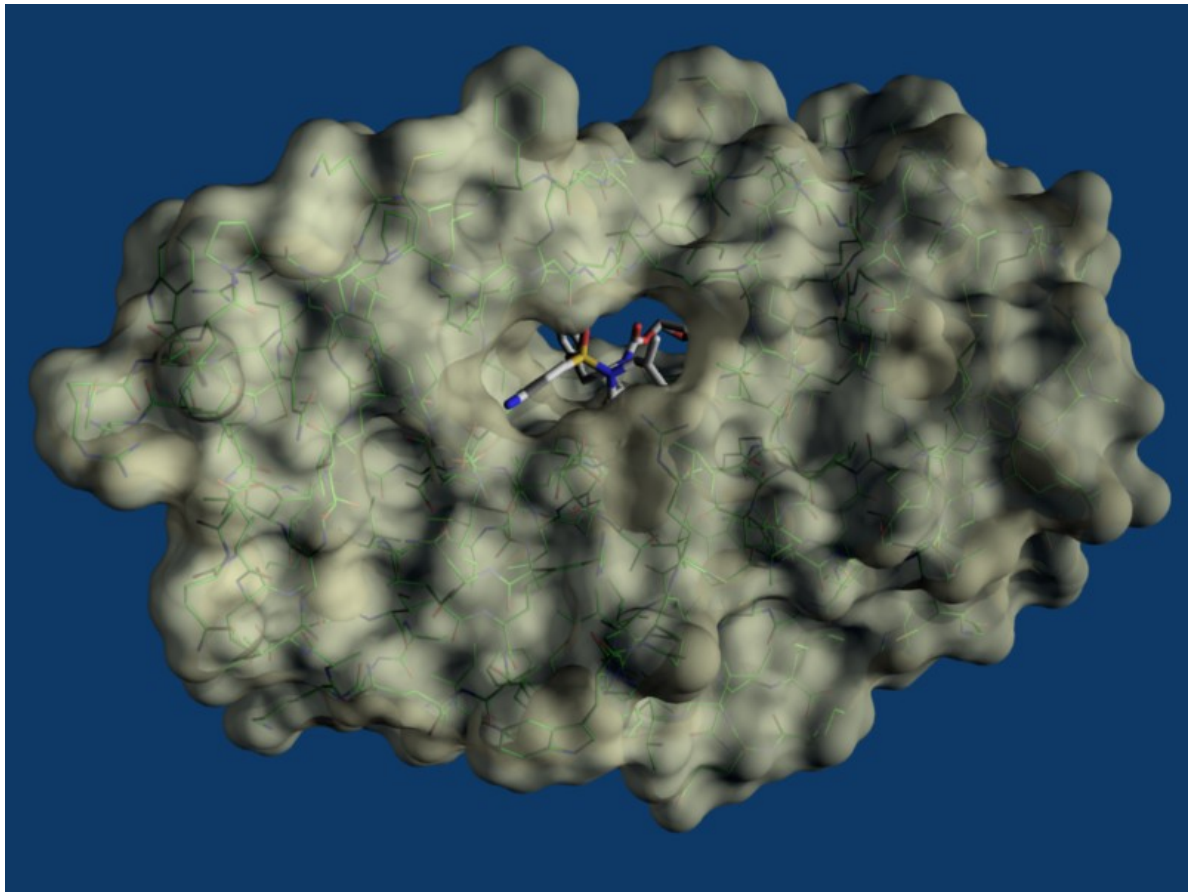
Of course, in a 'real-case' application there will be a huge number of such cavities. . .



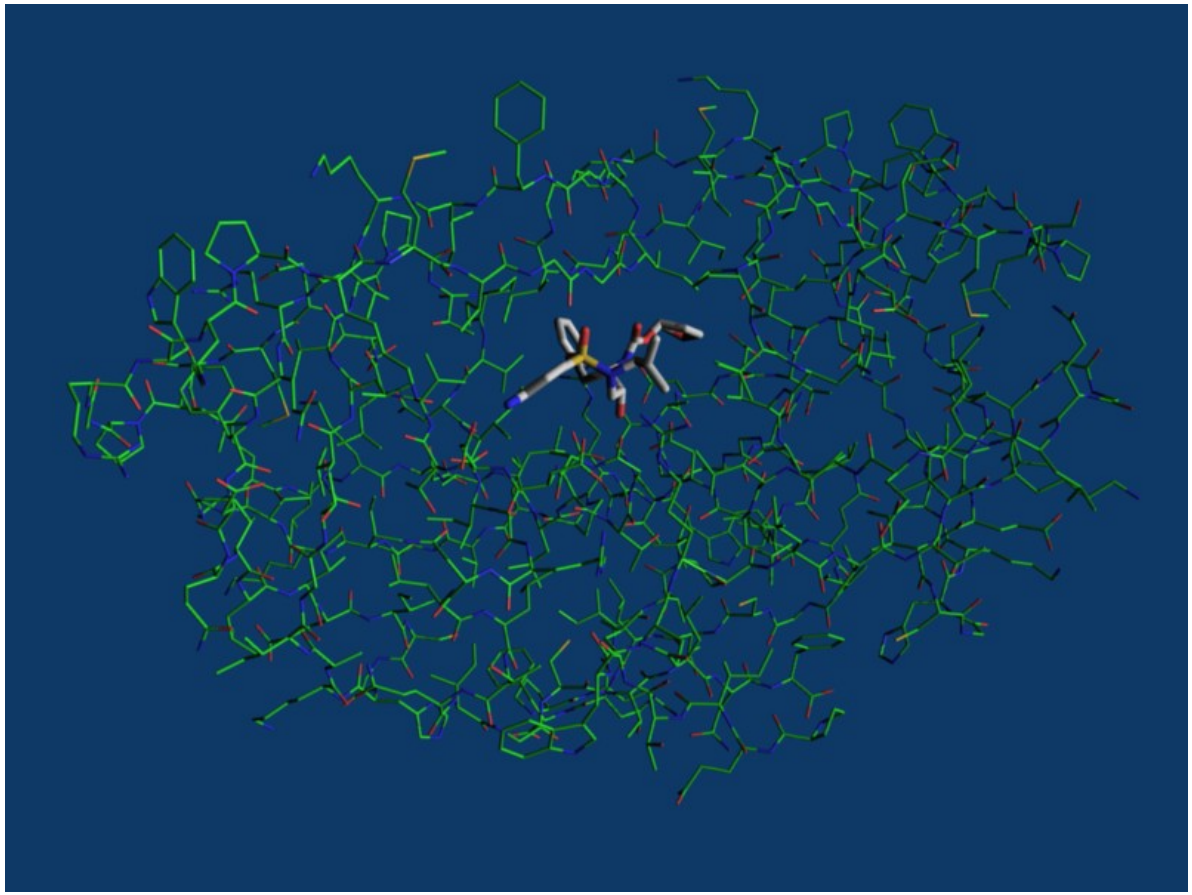
So the last step is to filter out all cavities with a volume below a given threshold.



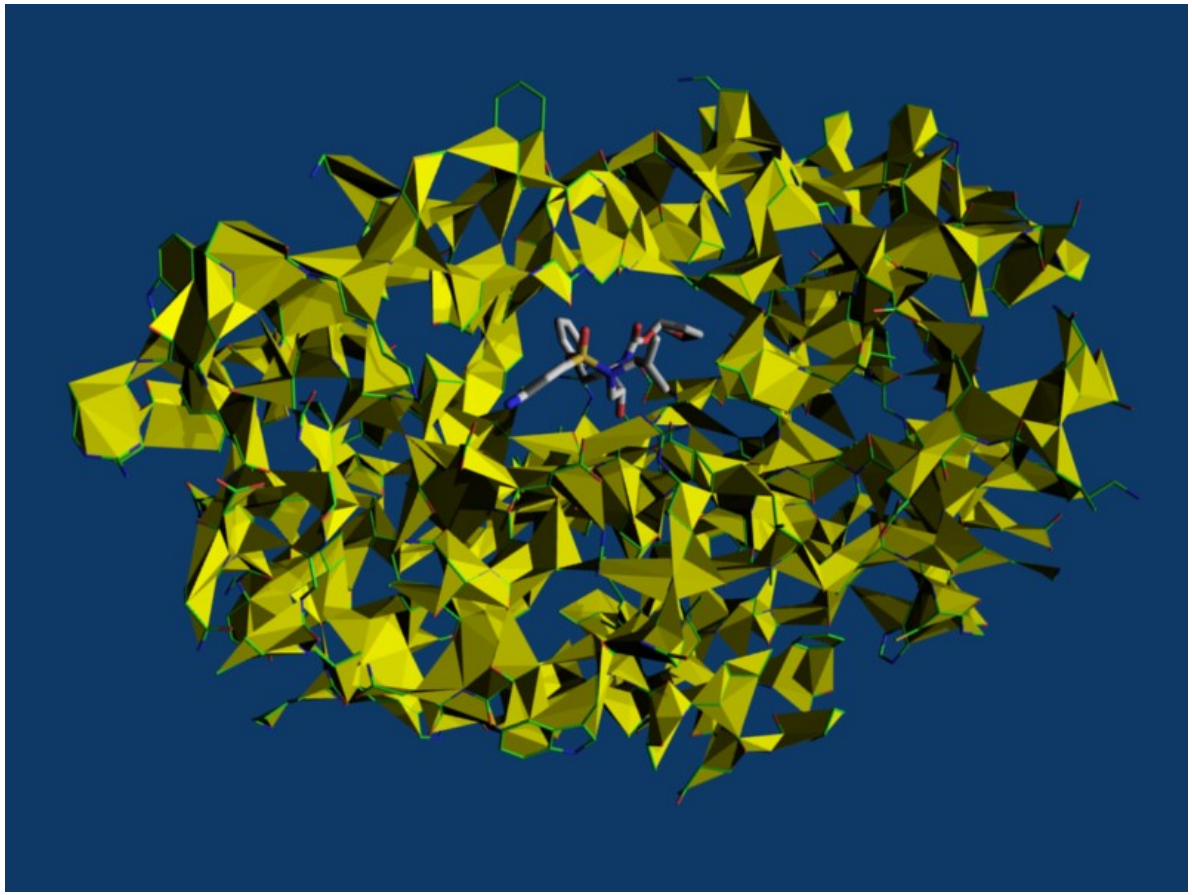
Here you can see the shape for a very low α-level. In that case we would not obtain reliable cavities. For an arbitrary set of points it is not trivial to choose an appropriate α-level, but luckily we are dealing with chemical structures and due to the normal range of bond lengths, we also know the valid range of lower α-levels. Let's have a look on a 3D example...



Here we have a HIV protease (1HPV)

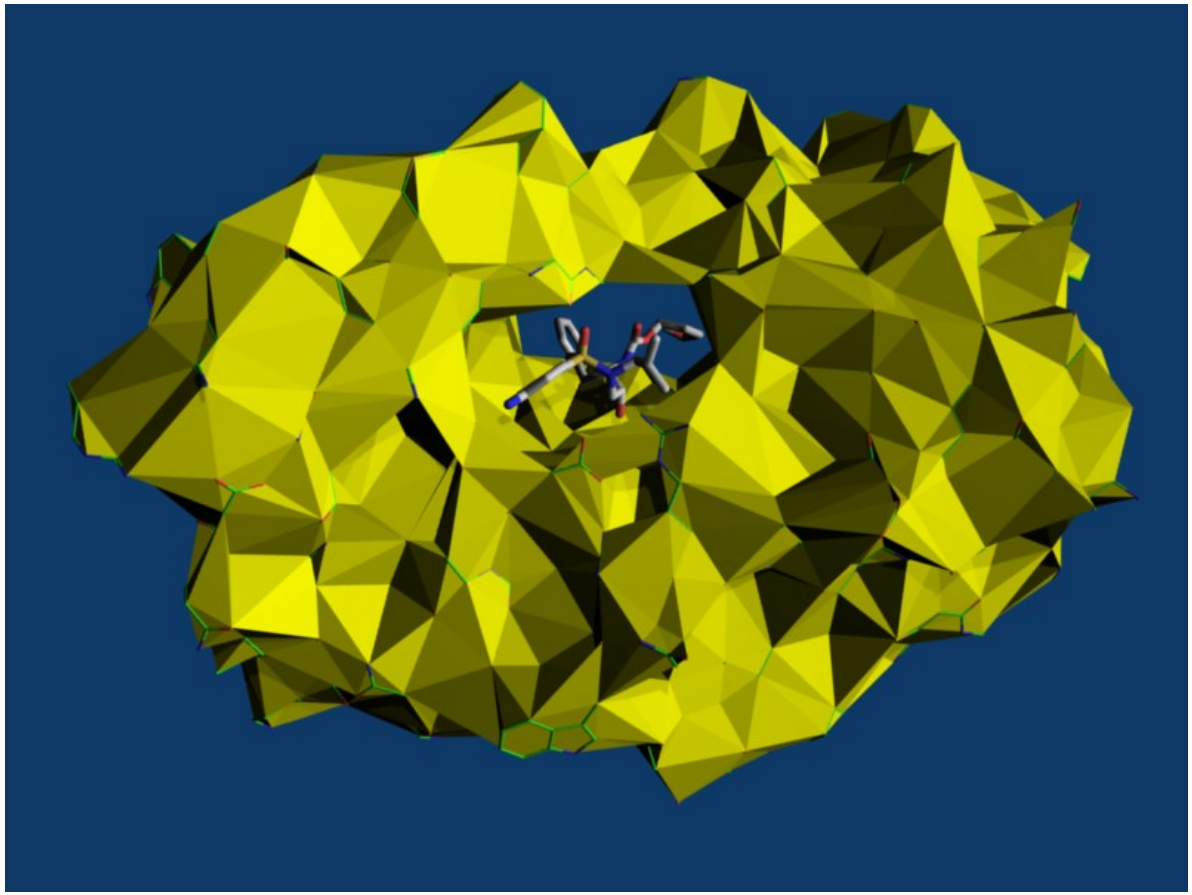


Here we have a HIV protease (1HPV)
First let us see how it looks if we apply a too small α-level in 3D...

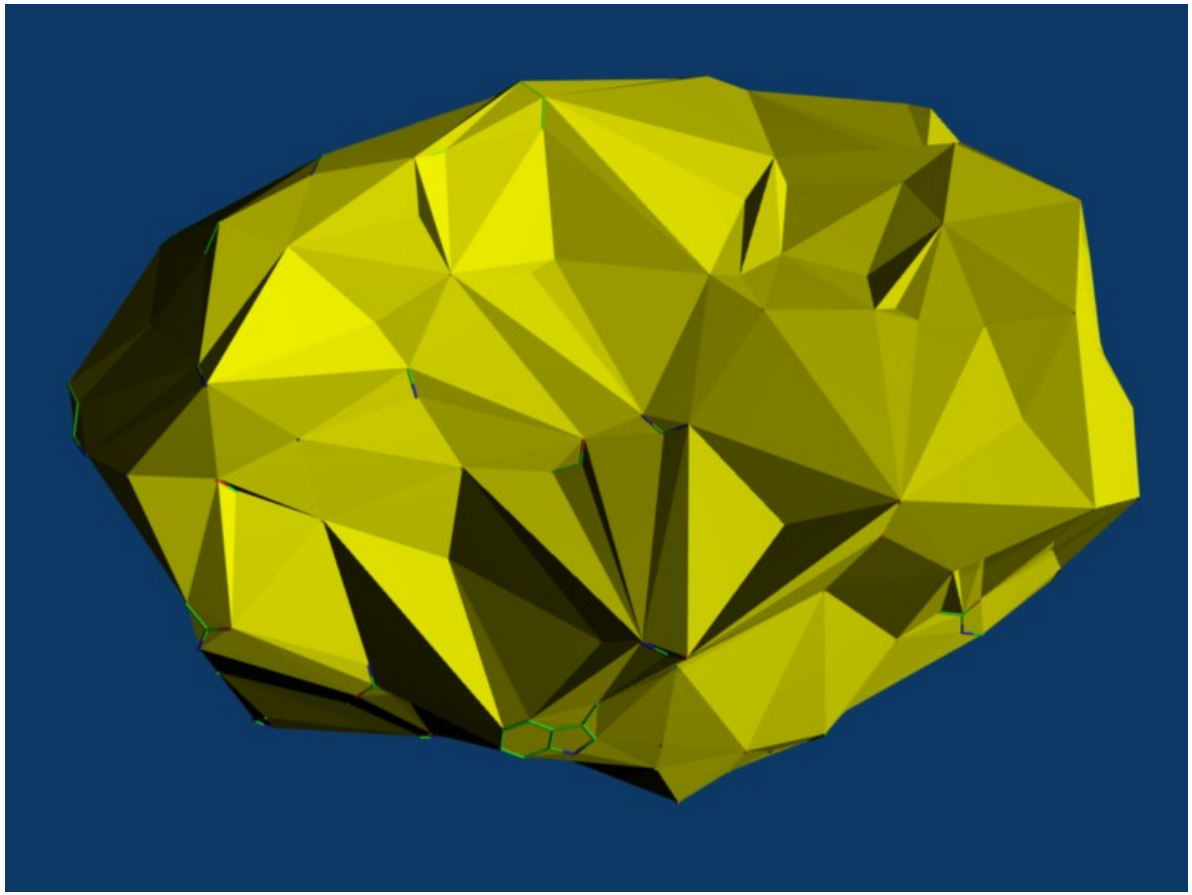


$\alpha = 1.2$ Å

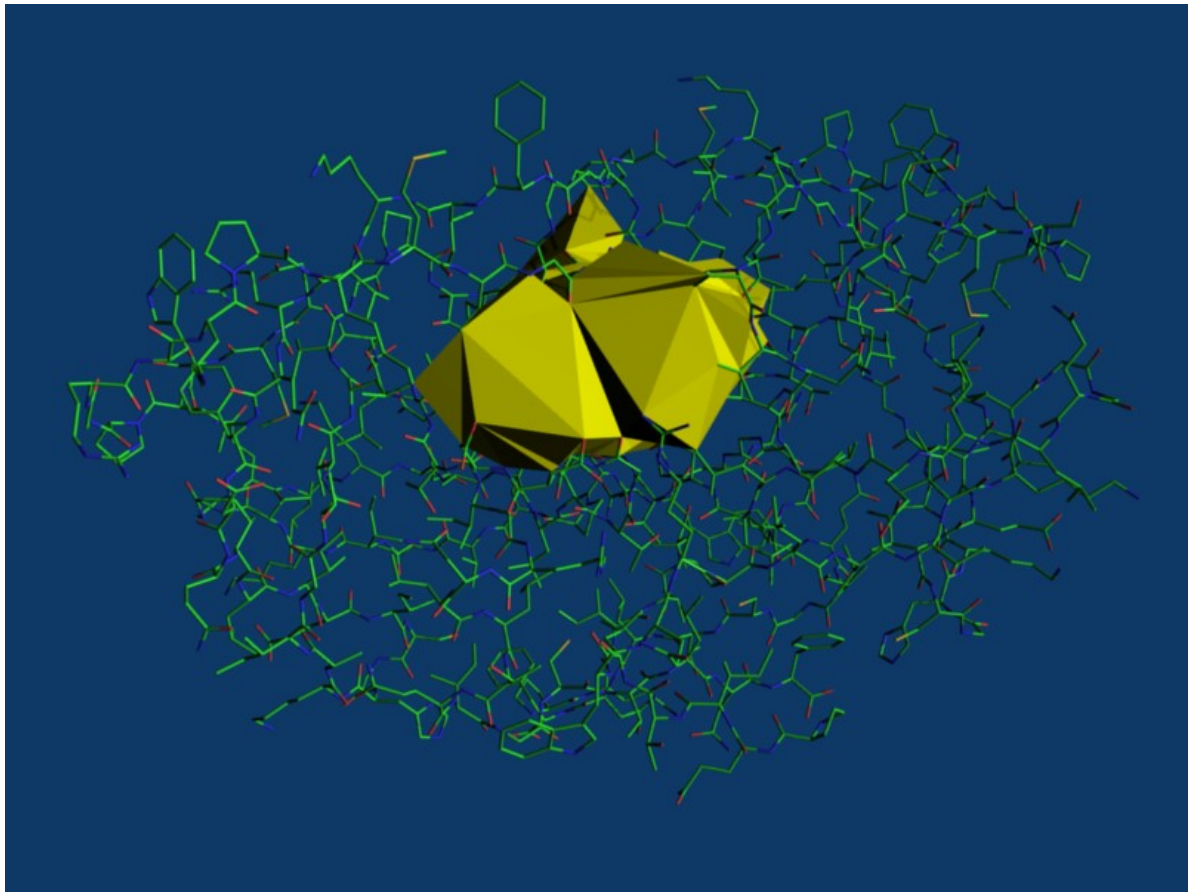
If you have read the help on -pa2 you already know the default of 2.5 Å and you also know, that you should never get below 2.2 Å



$\alpha = 2.6$ Å
This would be an appropriate lower level.



$\alpha = 8.0$ Å
This is the default for the upper level.



Here you can see the difference volume between $\alpha = 2.6$ Å and $\alpha = 8.0$ Å

```
fconv -pa2 1CIL.pdb
```

Of course we are using 1CIL again, so download it if necessary. Use the command shown in the box to obtain an automatic MOL2-cavity for default α-levels. If you have a look at the result you will notice that only those atoms are written out which are part of the cavities surface...

```
fconv -pa2 1CIL.pdb --r=4
```

Use the new command to also additionally write out atoms within 4 Å range to the surface atoms and watch the result. As you will see there are no bonds between the written atoms. But we can also write out full residues with bonds...

```
fconv -pa3 1CIL.pdb
```

Use the new command to write the detected cavity as PDB file with full residues (If you want to have MOL2, convert it with `-f`).

```
fconv -pa3 1CIL.pdb --r2=2.3
```

The new command changes the lower α-level to 2.3 Å resulting in a fine grained cavity which looks slightly better than our cavity with default values. But be careful, if you use 2.2 Å you will not only get two additional cavities on the surface, but also some intra-protein space.

```
fconv -pa3 1CIL.pdb --r3=10
```

Increasing the upper level to 10 Å results in an additional cavity which is larger than the first one, while the latter does not change so much).

```
fconv -pa3 1CIL.pdb --s2=1CIL_ETS_263_A.mol2
```

The algorithm considers only protein atoms, but if you e.g. have a cofactor you can supply it by option `--s2=` and fconv will use it as part of the protein. If you extract the ligand with `fconv -l 1CIL.pdb`, you will get 1CIL_ETS_263_A.mol2 and can use the command shown in the box. As expected, fconv no longer detects a cavity, as it is now occupied by the ligand.

- There will be a Part 4, but I've currently not chosen a topic...

Thanks go out to...

- Prof. Dr. Gerhard Klebe
- fconv bug reporters