APPENDIX

Inner border tracing algorithm in Roll

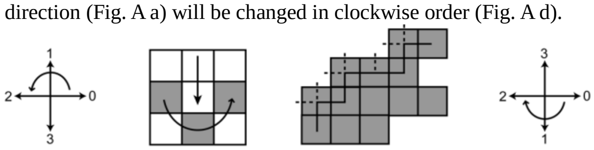
The inner border tracing algorithm is used to search the boundary of a specific region. A schematic illustration of this algorithm in 4-connectivity (one grid has four neighbourhood grids) is shown in Fig. A. The algorithm is as follows:

1) The search begins at a grid with the smallest column coordinate of all grids with the smallest row coordinate. Define a variable, dir_{pre}, which stores the direction of the previous move along the border from the previous border grid to the current border grid and initialises dir_{pre} equal to the downward direction.

2) Search the 3×3 neighbourhood of the current grid in the anticlockwise direction, beginning with a neighbourhood search from the grid position in the direction = (dir_{pre} + 3) mod 4 in anticlockwise order (Fig. A b). The first grey grid found is a new boundary element. Then, update dir_{pre} value.

3) Repeat step 2 until returning to the beginning position.

With this inner border tracing algorithm, the boundary of a region can be obtained efficiently. However, for Roll, this algorithm should be adjusted as the probe surface is the outer border of the protein. Therefore, the search direction (Fig. A a) will be changed in clockwise order (Fig. A d).



(a) (b) (c)

(d)

Fig. A. Inner border tracing algorithm. (a) Original search direction. The numbers 0 – 3 represent the four search directions. (b) Neighbourhood grid search order. The grey grids represent the region the boundary of which will be searched. The white grids represent the free grids. (c) Boundary tracing of a 4-connectivity region. The dashed lines show the search attempts in the border tracing of this region. (d) Search direction in Roll. The directions 1 and 3 are reversed.

Four direction controllers

From the inner border tracing algorithm, we deduced four direction controllers: *up controller*, *right controller*, *down controller* and *left controller* (upper part in Fig. B) to control the rolling direction.

Here, we use *right controller* as an example to explain how these controllers function. Assuming the probe sphere rolled in an orderly manner through points A, B and C and will roll toward to point D, the rolling direction from B to C is rightward ($dir_{pre} = 0$) so the *right controller* is adopted. For the next roll, according to the right controller, the probe sphere will first attempt to roll upwards. However, this will not be a successful move (Fig. C b), so the second attempt (*i.e.*, rightward) is performed, which is a successful move as shown in Fig. C (c). Once complete rolling is finished, the value of dir_{pre} is be updated to "0" for the next step.

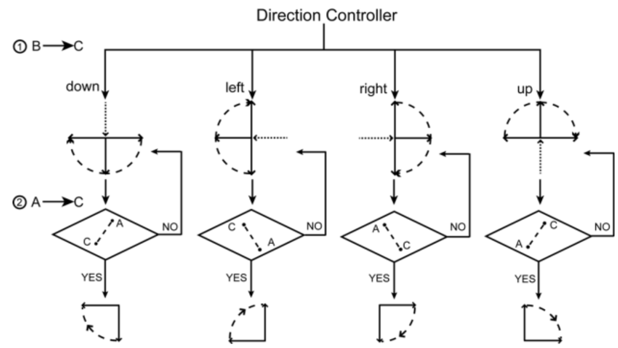


Fig. B. The upper part shows the four original direction controllers. The dotted lines represent the dir_{pre} direction. The solid lines represent the three possible rolling directions. The dashed lines represent the rolling order. Each direction controller has three possible directions and the rolling order is clockwise. The lower part shows the four efficient direction controllers that omit some directions when the position relationship between point A and C is satisfied according to the criteria in the decision boxes.

Efficient rolling

For each controller in Fig. B, there are three possible directions of movement. In some cases the number of possible directions can be reduced to roll the probe sphere more efficiently. We used the same example as shown in Fig. C. Normally, two attempts are necessary for one successful roll (Figs. C b and c). However, the positional relationship between point A and point C is helpful to avoid these unnecessary attempts. As shown in Fig. C (b), point C is at the bottom right of point A. In this case, upward movement is omitted (Fig. C e). Therefore, only one movement attempt is needed (*i.e.*, rightward).

The final direction controllers are shown in the lower part of Fig. B. For one roll of the probe sphere, the first step is to choose one controller from the four according to the rolling direction from the previous point B to the current point C. Then, in terms of the positional relationship between points A and C, unnecessary rolling attempts will be omitted.

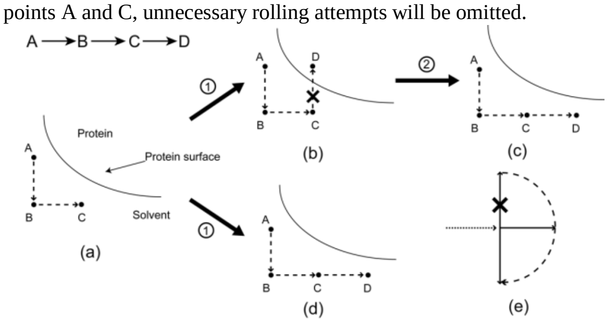


Fig. C. Right controller. (a) Three consecutive rolling positions A, B and C. (b) Failed roll. (c) Successful roll with two attempts. (d) Successful roll with only once attempt. (e) Remove one search direction in *right controller* as point C is at the bottom right of point A.

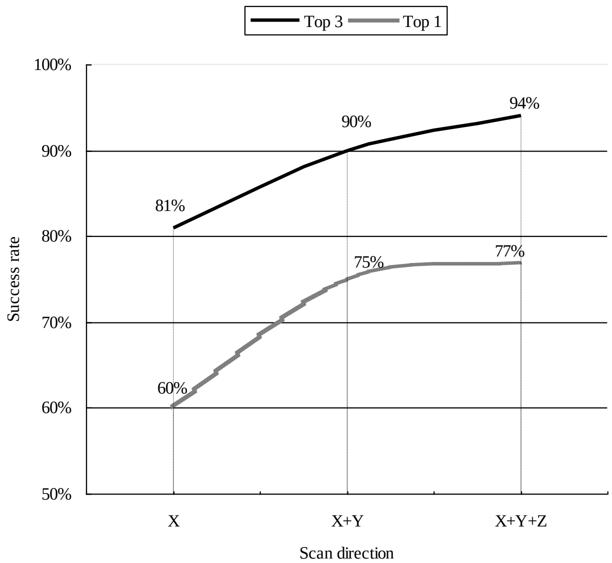


Fig D. Success rates of different rolling directions for the 48 bound structures. The shadow line represents the success rates for Top1 and the solid line represents the success rates for Top3. The success rates were greatly improved from one direction namely (*x*) direction to two direction (*x*)+(*y*); however, from (*x*)+(*y*) to (*x*)+(*y*)+(*z*), the success rates slow down to increase.