High Throughput Convolution for Protein Docking

Hourai, Y.
HPC team, CBRC, AIST
Nukada, A.
JST CREST, Univ. of Tokyo

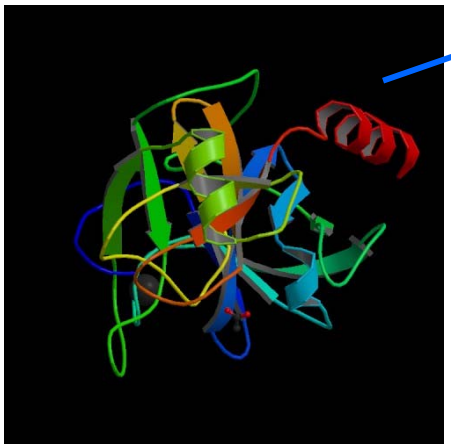
CBRC

National Institute of Advanced Industrial Science and Technology
AIST

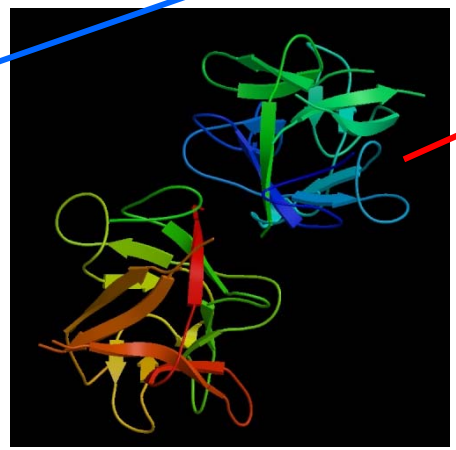
Protein Docking

- **>30,000 protein structures are in PDB database.**
  - Interactions among them are determined by experiments.
    - Yeast two-hybrid essay, mass spectrometry, etc.
- **Structure of a complex in 3D-space**
  - Experiments
    - X-ray crystallography, NMR, etc.
    - Difficult to determine.

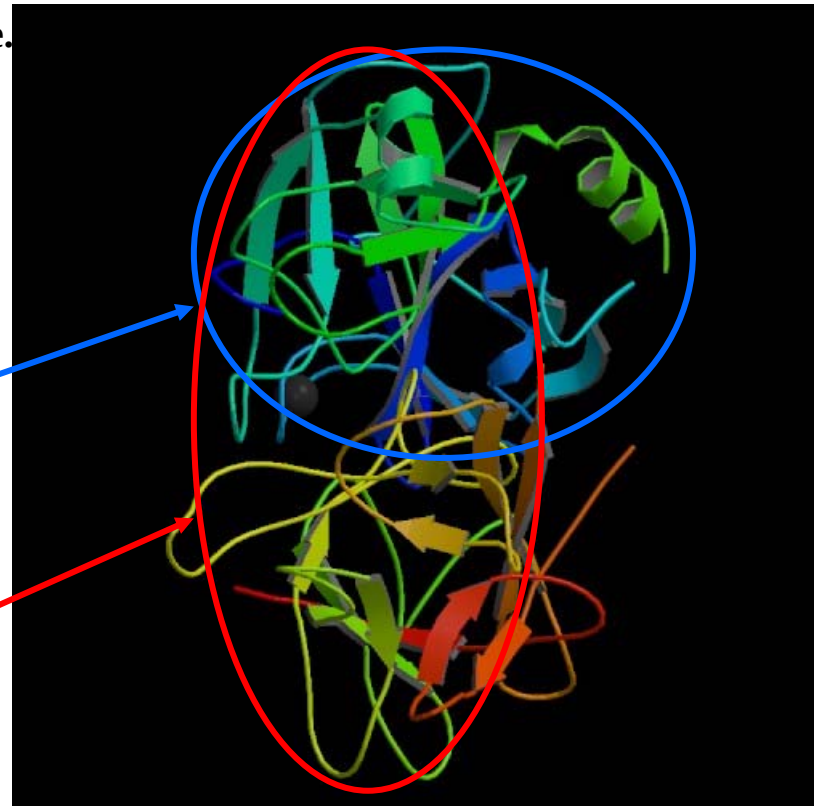
By computation



Porcine trypsin
(1qqu)



Soybean trypsin inhibitor
(1ba7)

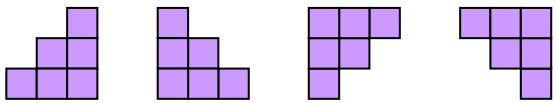


Complex (1avx)

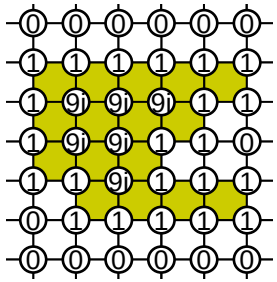
make a complex without destroying original structures.

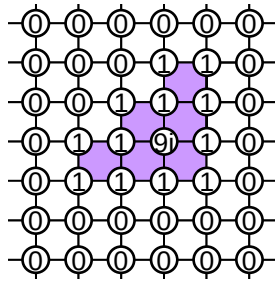
Shape Complementary Search

- Shape complementary search is the first step from computational approach.
- Evaluation of docking orientation
  - Proteins are discretized to NxNxN cubic grids assuming proteins have rigid body.
  - Score functions are based on shape complementarity, desolvation energy, and electrostatics. [FtDock, ZDock]
- Search
  - Exhaustively to avoid overlooking.
    - Scores by translational move is computed by complex (or real number) convolution.
    - Search is iterated for multiple angles.
      - e.g. angle steps in 3D vs. #non-isomorphic rotational moves
        - Δ=15° → 3600 rot.
        - Δ=6° → 54000 rot.



Rotation Δ=90° in 2D





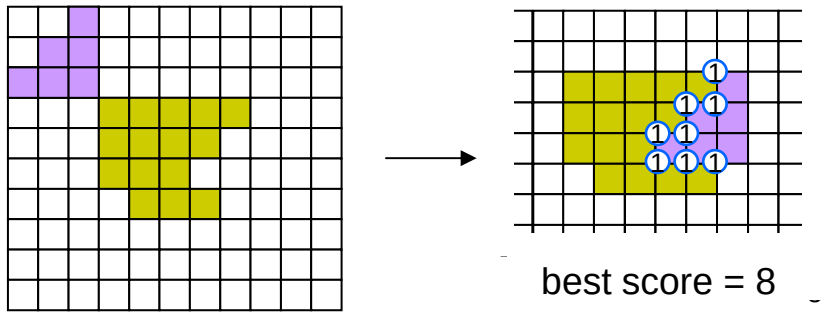
Local match score:

core – core = 9i × 9i = -81

surface – surface = 1 × 1 = +1

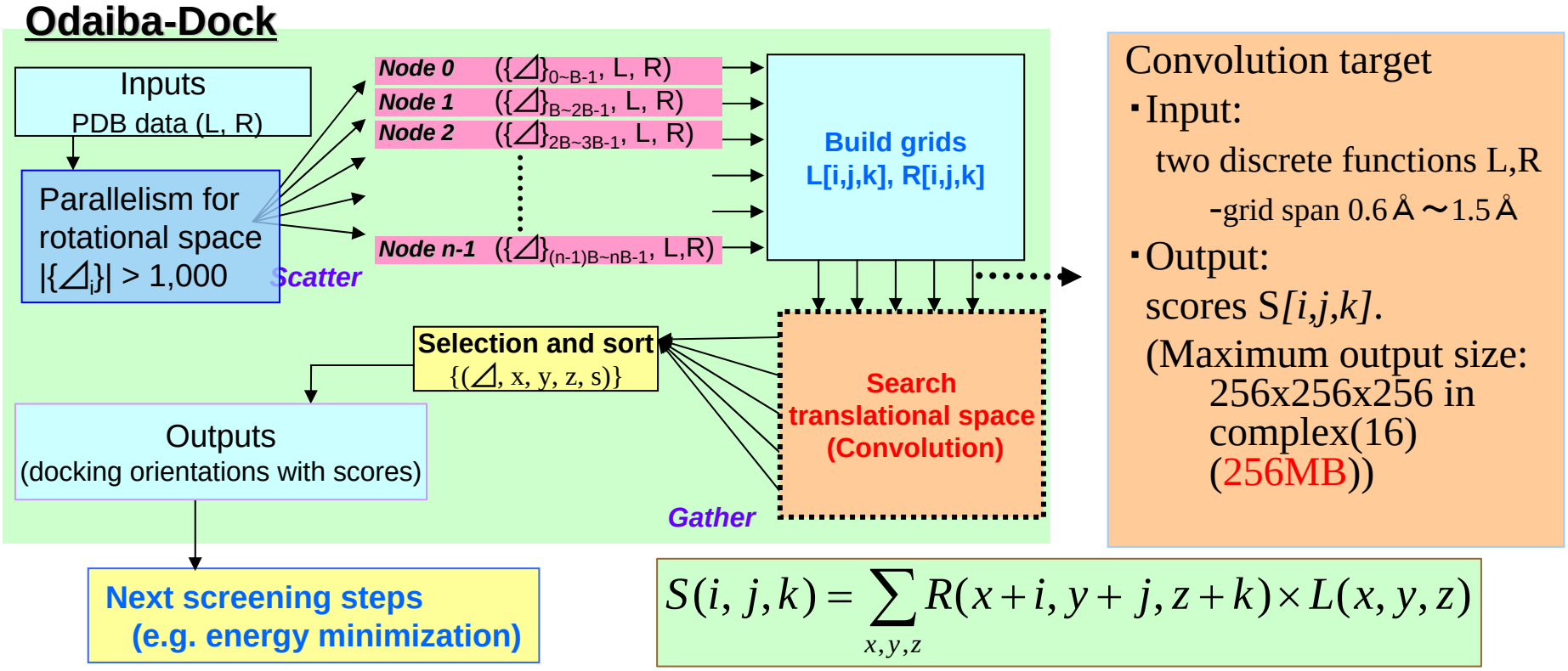
$$S(k,l) = \sum_{i,j} R(i,j) \times L(i+k, j+l)$$

convolution (correlation)



Example of translational search in 2D

OdaibaDock: Target Problem & Strategy



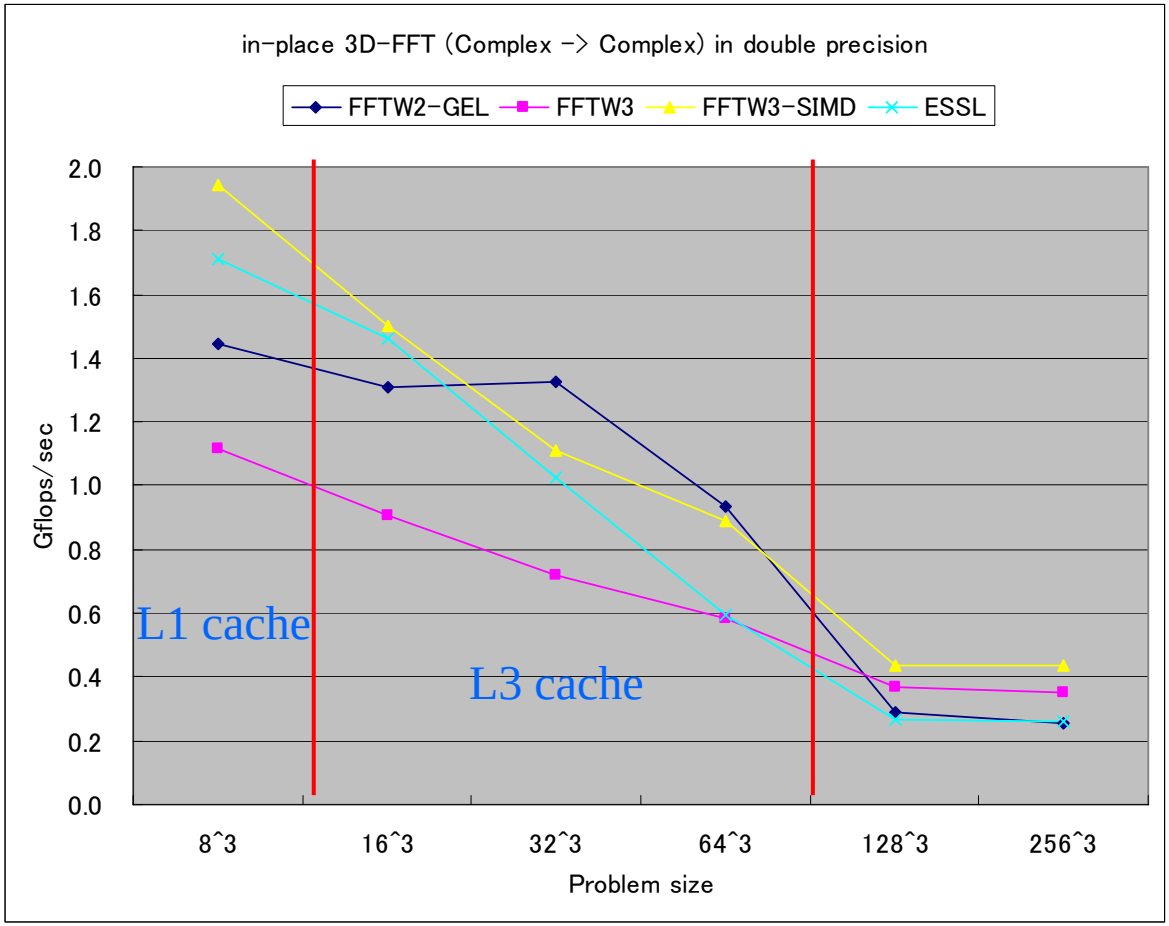
$$S(i,j,k) = \sum_{x,y,z} R(x+i, y+j, z+k) \times L(x,y,z)$$

Time complexity

Direct : $O(N^6)$

FFT : $O(N^3 \log N)$

3D-FFT Performance (fftw, essl)



Output Size 8KB 64KB 512KB 4MB 32MB 256MB

- fftw-gel:
  fftw-gel.2.1.5 provided by Stefan Kral (Vienna Univ.) and Bob Walkup (IBM Watson)
- ESSL:
  The benchmark results are provided by people at IBM Japan.
- fftw3:
  fftw 3.1 with compiler optimization for Double Hummer (xlc)
- fftw3-simd:
  modified fftw-3.1 [-enable-sse2] (SSE2 instructions are replaced with Double Hummer intrinsic functions.)

Improved memory access in 3D-FFT

Not open to the public
for the time being

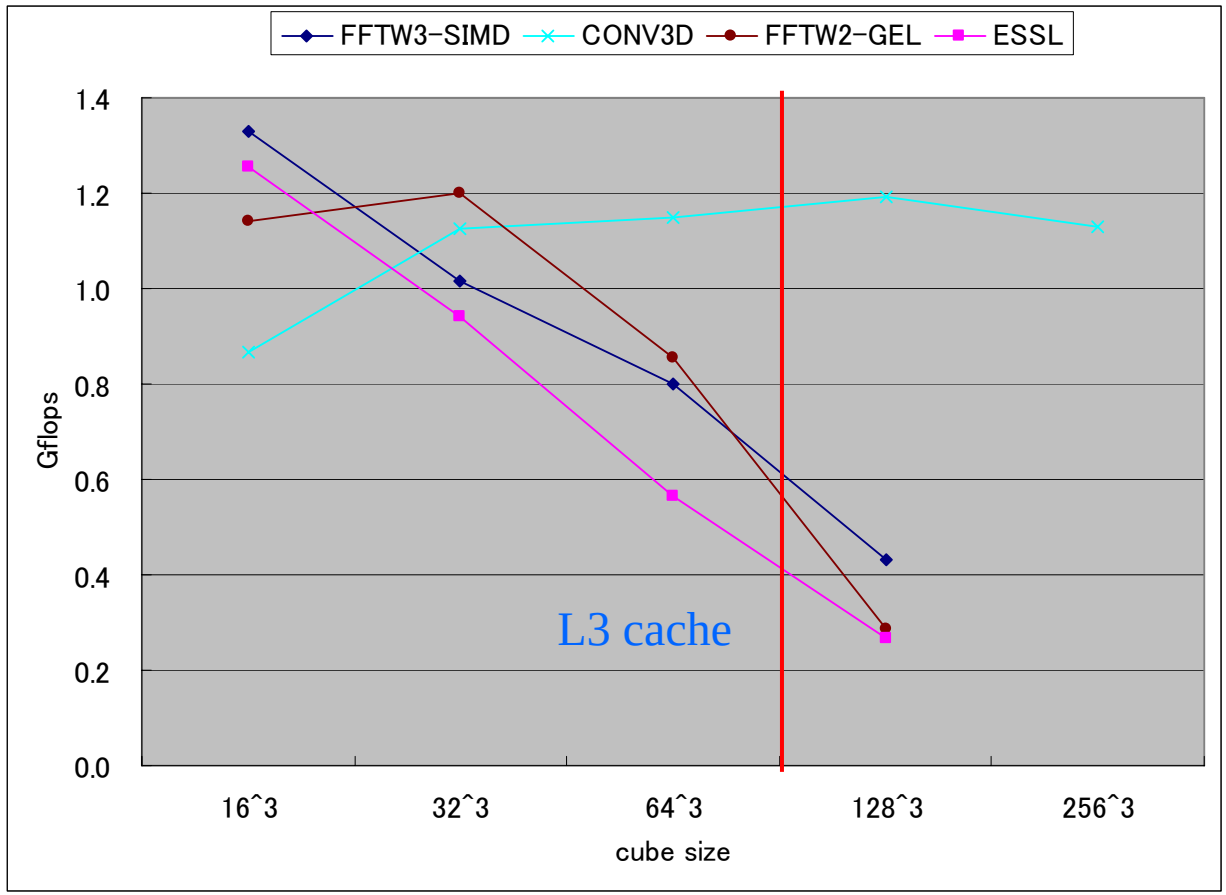
1D-FFT kernel for 3D-Convolution

Complex → Complex (Double Precision)

Not open to the public
for the time being

3D-Convolution Performance

(Complex → Complex)



Output Size	64KB	512KB	4MB	32MB	256MB

Why CONV3D fast?

Not open to the public
for the time being

Summary

Not open to the public
for the time being