Chem-Bio Informatics Journal, Vol. 2, No. 4, pp.102-118(2002) ***Review***

Peach-Grape system - a high performance simulator for biomolecules

Yuto Komeiji* and Masami Uebayasi

IBRF/RICS, National Institute of Advanced Industrial Science and Technology, AIST Central 6, Tsukuba, 305-8566

**E-mail: y-komeiji@aist.go.jp*

(Received October 22, 2002; accepted November 27, 2002; published online December 30, 2002)

Abstract

A high performance system for the molecular dynamics simulation of biological molecules was constructed by combining a software package, Peach, with a special-purpose computer, Grape. The resultant simulator "Peach-Grape system" was used to analyze several important biological molecules including the Hin/DNA complex, the *trp*-Repressor/DNA complex, and Calmodulin. In addition to those simulations performed by the Peach-Grape system, other simulation studies of biomolecules by special-purpose computers are briefly reviewed.

Key Words: MD, simulation, Peach, Grape, protein, DNA

Area of Interest: Molecular Computing

1. Introduction

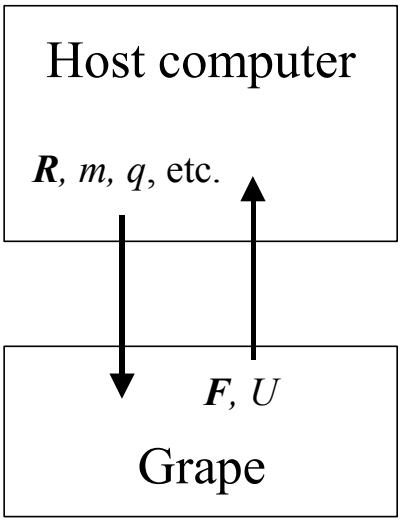
The Peach-Grape system is a high-performance simulator of biological molecules. Peach stands for "Program for Energetic Analysis of bioCHemical molecules," a software package for the molecular dynamics (MD) simulation of biological molecules. Grape stands for "GRAvity piPE," a family of special hardware for *N*-body problems. We constructed the Peach-Grape system in 1995 by combination the Peach software and the Grape hardware [1] [2]. Since then we have applied the system to the MD simulations of several important biological molecules.

Recently, we published our last paper dealing with the system [3]. Hence, here we would like to review and summarize our studies with the Peach-Grape system, as well as related studies with other kinds of special computers for MD. We will focus on those papers reporting practical MD studies on biological molecules, rather than those related to the hardware system development.

2. GRAPE: special hardware for *N*-body problems

In this section, we first describe an outline of the Grape hardware system, and then summarize studies on Grape and related computers specialized for MD.

Fig. 1. Grape hardware system.



Grape is connected to a host computer (a workstation or PC). The host sends to Grape the coordinates (*R*), mass (*m*, or charge *q*),and other parameters of the particles. Then Grape computes force (*F*) and potential (*U*) and sends them back to the host.

2.1 Outline of Grape

As its name "GRavity piPE" indicates, Grape was originally developed as a special computer for use on gravitational problems by the group of D. Sugimoto at the University of Tokyo [4] [5]. In the Sugimoto laboratory, computer simulations of gravitational *N*-body systems were conducted to analyze cosmological problems. In such simulations, the Newtonian equation (1) of motion for an *N*-body system is solved numerically.

$$m_i \frac{d^2\mathbf{R}_i}{dt^2} = -m_i \sum_{j \neq i}^{N} \frac{m_j}{R_{ij}^3}\left(\mathbf{R}_i - \mathbf{R}_j\right) \qquad (1)$$

Here, $\mathbf{R}_{i(j)}$ and $m_{i(j)}$ are the position vector and mass of particle *i(j)*, respectively. The right-hand-side of equation (1) is the gravitational force, whose computational cost is $O(N^2)$ and is the most CPU-time consuming part of the simulation procedure. Grape computes **only** the gravitational force and potential, whose mathematical simplicity is the reason for Grape's hardware simplicity. Because the hardware is simple, Grape can be made faster than ordinary computers at a lower cost.

In brief, Grape is a hardware accelerator. The Grape hardware system is illustrated in Fig. 1. Grape is connected to a host computer, usually a PC or workstation. The host sends the position and mass of each particle to Grape. Then, Grape computes the gravitational force and energy

rapidly and returns the computed values to the host. The host takes care of all miscellaneous tasks such as time integration, file input and output, and so on. Thus, assuming that the performance of Grape is much faster than that of the host, the total performance of the system is accelerated without sacrificing precision. For example, Grape accelerated MD simulations of several solvated proteins by 20- to 200-fold [1].

The advantages of the Grape system have been recognized in many fields of numerical simulations [6] [7] [8], including MD simulation. In the next subsection, we focus on those special computers for MD simulation.

2.2 Grape and related special computers for molecular dynamics

A number of special computers for MD simulation have been developed since 1992. Usually the computation of electrostatic energy is the most time consuming part of an MD simulation. It is also known that the functional form of electrostatic energy is the same as that of gravitational energy. Equation (2) represents the motion of a Coulombic system, and is similar to equation (1), except that the sign of the right-hand-side is opposite and $m_{i(j)}$ is replaced by $q_{i(j)}$, the particle charge.

$$m_i \frac{d^2\mathbf{R}_i}{dt^2} = q_i \sum_{j \neq i}^{N} \frac{q_j}{R_{ij}^{\ 3}} \left(\mathbf{R}_i - \mathbf{R}_j\right) \qquad (2)$$

Thus, due to the similarity of the gravitational and Coulombic problems, the success of the gravitational simulation by GRAPE soon drew the attention of MD scientists and promoted the development of special computers for MD.

To review the special computers for MD, we would like to use a “family tree” (Fig. 2) to clarify the relationship among them. Table 1 summarizes the publications on MD simulations of biological molecules analyzed by those MD computers. Papers reporting only system construction and evaluation are excluded from the list.

We classified the computers into three generations (Fig. 2). The first generation computers, (1) Grape2A and (2) Wine-1, were not used practically, but their architectures have served as the prototypes for the succeeding computers. The second generation computers (3) MD-engine and (4) MD-grape (nearly equal to (5) ITL-md-one) were constructed with custom LSIs developed by adapting the architectures of the first generation computers. ITL-md-one and MD-engine have been used for practical simulations of biological molecules. ITL-md-one was the Grape we used to construct the Peach-Grape system. The third generation computers (6) MD-engine II and (7) MDM are massive parallel computers for MD.

Table 1. Application of special-purpose computers for MD to simulate biological molecules *.

Computer	Program	Molecule	Outline of MD	Reference
(1) Grape2A [9]				None reported.
(2) Wine-1 [10]				None reported.
(3) MD-engine [11][12]	AMBER	Ras p21 protein.	Temperature separation.	Oda et al., 1996 [13].
		HIV-1 Protease.	Modeling of ES-complex. Catalytic process.	Okimoto et al., 1999, 2000, 2001. [14][15][16]
		Ras p21 protein.	Inhibition mechanism.	Futatsugi & Tsuda, 2001 [17].
		DMPC bilayer.	Orientational correlations of lipid molecules.	Takaoka et al., 2000 [18].
		DMPC bilayer.	Initial process of membrane fusion.	Ohta-Iino et al., 2001 [19].
(4) MD-grape [20]				None reported.
(5) ITL-md-one [1,][21]	PEACH	Histidine containing Protein.	Comparison of boundary conditions, ensembles, and methods for the electrostatic.	Komeiji et al., 1997 [2].
		Hin-recombinase/ DNA complex.	Artifact of cutoff. Conformational change upon DNA binding.	Komeiji et al., 1999 [22][23].
		TrpR/DNA complex.	DNA recognition via H-bonds.	Suenaga et al., 2000 [24].
		Sex pheromone binding protein.	Modeling of the apo-form.	Nemoto et al., 2001 [25].
		Calmodulin.	Conformational change upon Calcium binding.	Komeiji et al., 2002 [3].
	PRESTO	Ala 15mer peptide.	Helix-coil transition.	Takano et al., 1999 [26].
(6) MD-engine II [12]				None reported.
(7) MDM [27][28]				None reported.

* This list contains papers on MD simulations of biological molecules (peptides, proteins, polynucleotides, and lipids) published in international journals up to August 2002. The following papers are **excluded** from the list: (1) papers on non-biological molecules, (2) papers reporting only system description or benchmark tests, and (3) conference abstracts.

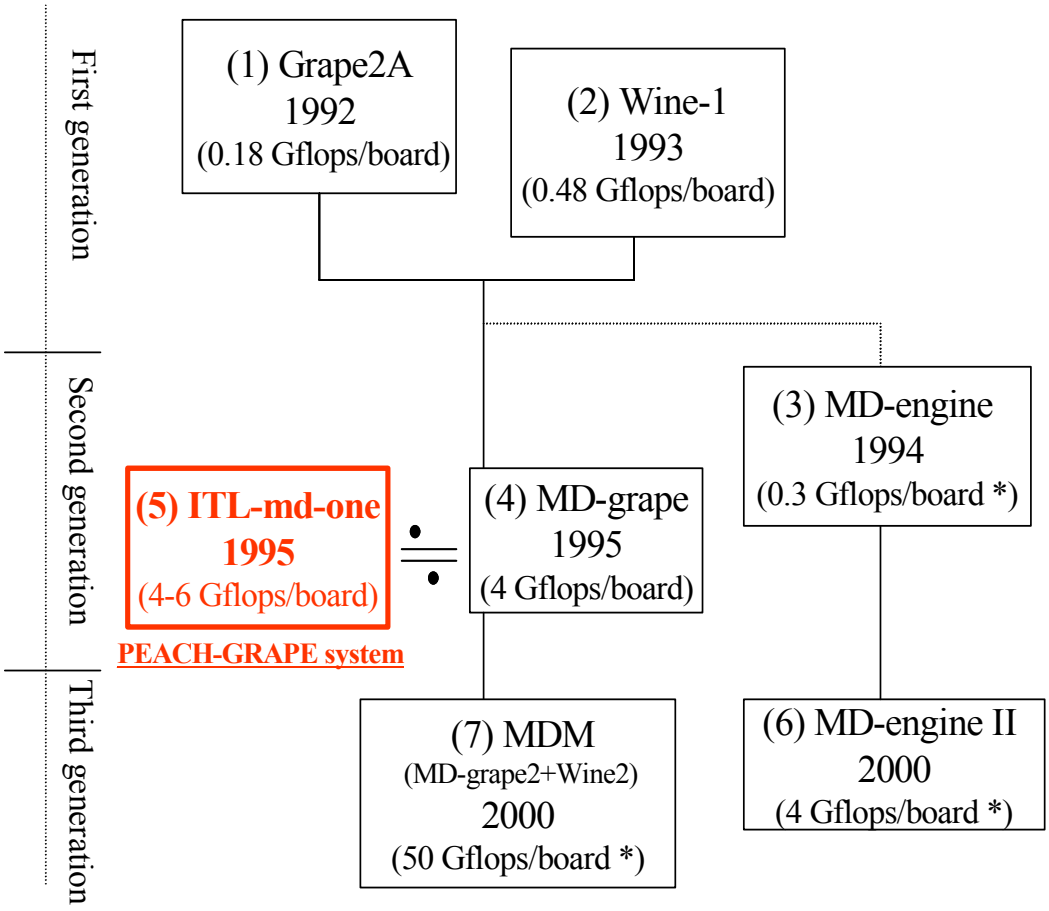
(1) Grape2A was the first Grape-type computer specialized for MD simulation and was developed in the Sugimoto laboratory. Grape2A was designed for computation of Coulomb, Van Der Waals (VDW), and Ewald-real space interactions. It was constructed by using LSIs available on the market, and the peak speed was 0.18 Gflops.

(2) Wine-1 (Wave space INtegrator for Ewald method) was constructed in the Sugimoto laboratory for computation of the Ewald-k space summation. The peak speed was 0.48 Gflops.

Grape2A and Wine-1 were the ancestors of special-purpose computers for MD because their basic architectures were adopted for use in succeeding MD computers.

The architectures of Grape2A and Wine-1 were made into a Large Scale Integrated Circuit (LSI) named the MD-chip; these chips were used to construct a new parallel computer named **(4) MD-grape**. MD-grape was constructed by a joint effort of the Sugimoto Laboratory and Information Technology Laboratory (ITL Corp.). MD-grape has been sold by the ITL Corp. under the name of **(5) ITL-md-one**. MD-grape and ITL-md-one were nearly equivalent machines, but they used different interface libraries.

Fig. 2. Family tree of MD computers.



See Table 1 for references to these computers.
* Multiple boards can be used to construct a highly parallel-machine.

(7) MDM (Molecular Dynamics Machine) is a massive parallel version of MD-grape. MDM has been developed by T. Ebisuzaki's group at RIKEN. The main purpose for the development of the MDM is to analyze protein dynamics and folding. In 2000, MDM won the Gordon Bell Prize for peak performance by attaining a speed of 1.4 Tflops. A few articles on the MDM have been published in conference proceedings, which describe the architecture and performance of the system and give some demonstrative simulations [27] [29] [30]. To our knowledge, however, no practical simulation of biological molecules has been so far published in the literature.

Yet another family of MD computers has been developed by a joint effort of the Taisho Pharmaceutical Co. and Fuji-Xerox Co. Their computer, **(3) MD-engine**, is not a Grape machine (only those computers developed by Sugimoto and coworkers are called "Grape"). The basic architecture of the LSI used in MD-engine was also based on Grape2A and Wine-1, but MD-engine was greatly optimized for the purpose of molecular simulation. With MD-engine it is possible to use conventional methods of molecular simulation such as the nonbonded pair list, the minimum

image method, the treatment of the excluded atoms, a rectangular box rather than a cube, pressure ensemble, and so on. Then, the developers wrote a patch for Amber [32], a popular software package for biomolecular simulations, to enable its use with MD-engine. Thus, MD-engine has been used with Amber for scientific MD simulations of biological molecules (Table 1). Stimulated by the success of MD-engine, a new model named **(6) MD-engine II** has been developed by Fuji Xerox Co. Applications of MD-engine II to biological molecules have not yet been published.

Thus, a number of special computers for MD have been developed. In the following section, we will give a detailed account of our studies performed with (5) ITL-md-one.

3. Construction of the Peach-Grape system

In this section, we give the outline of the Peach-Grape system, briefly explain the Peach software, and then describe the functions of the Grape hardware. The Grape we used was ITL-md-one (computer 5 in Fig. 2), but it was nearly equivalent to MD-grape (computer 4). Hence, we will refer to ITL-md-one as "Grape" hereafter. See references [1] [2] for full documentation of the Peach-Grape system.

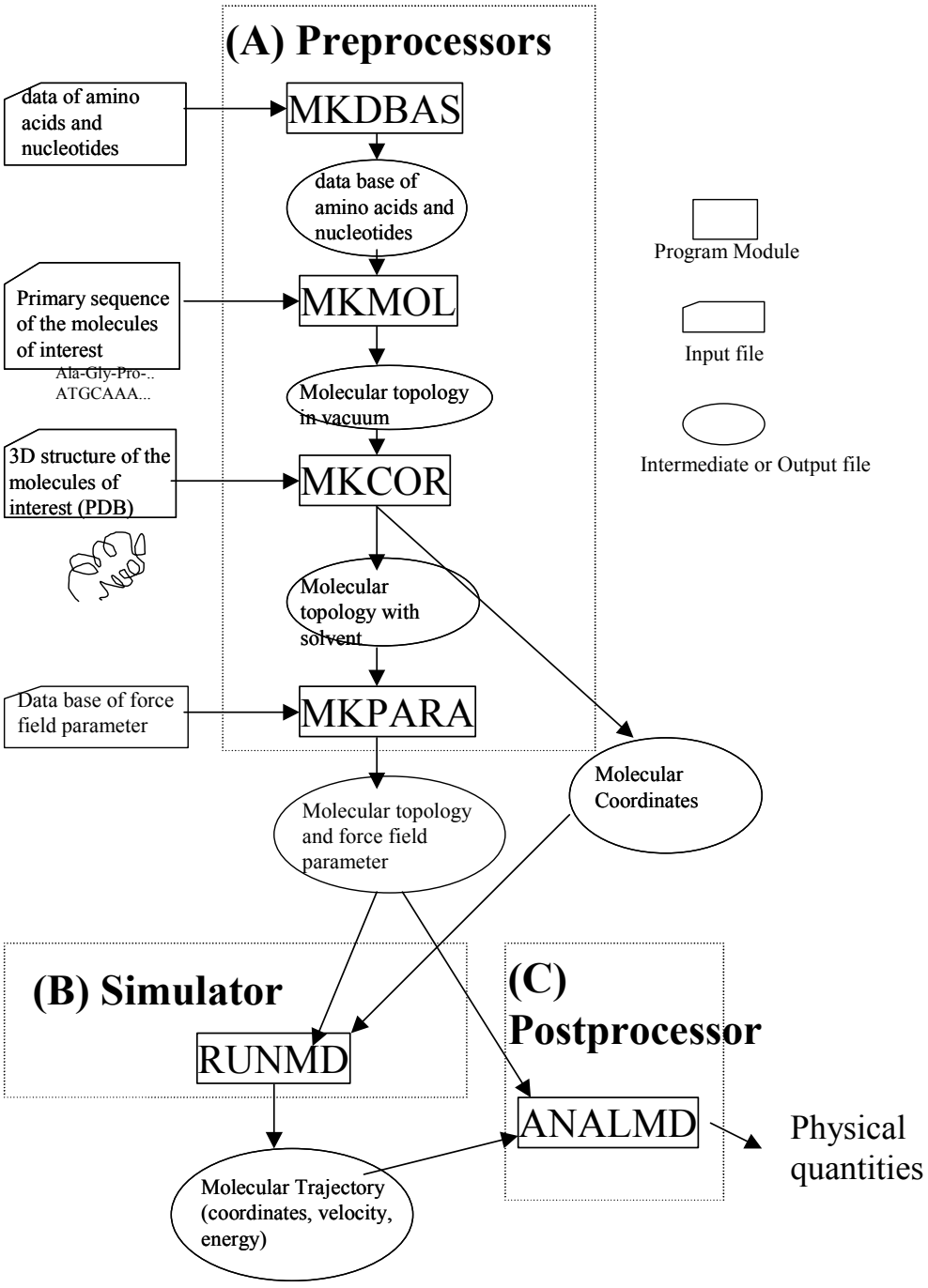
Peach is a software package for the MD simulation and Energy Minimization (EM) of biological molecules. Similar to conventional program packages such as Amber [32], Peach comprises several program modules, among which data are transmitted via intermediate files (Fig. 3). The program modules are roughly categorized into three segments: (1) preprocessors, which prepare input files for simulation, a (2) simulator, which performs EM and MD simulation, and a (2) postprocessor, which analyzes the output files of the simulation. Features of Peach 3.8, the latest version, are summarized in Table 2.

Table 2. Features of Peach Ver. 3.8 (2002).

Force fields	Amber84, 86, 94, 96, 99 [a], and OPLS.
Interactions	Bond, Angle, Torsion, VDW [b], H-bond [b], and Electrostatic (Cut-off, Direct summation [b], Ewald summation [b], and PPPC [a] [33]).
Time integration	V-Verlet and RESPA [34].
Solvent model	Shell, Box, and Droplet.
Boundary condition	Periodic and Free.
Ensemble	NVE and NTV (Velocity scaling, Nose-Hoover [35], and Zhang [a] [36]).
Bond constraint	RATTLE [a].
Energy minimization	Steepest descent, Conjugate gradient, and Quenched dynamics.
Language	Fortran90.
Parallelization	MPI [a].
OS	UNIX, including LINUX.

[a] These features are relatively new to Peach and were not used in the simulations performed by Grape.
[b] These interactions are computed by Grape (only Peach 3.0a or before).

Fig. 3. Program modules of PEACH.



In Peach, the 'RUNMD' module (Fig. 3B, simulator) uses Grape for computation of the electrostatic (Coulomb and Ewald) and VDW interactions (See legend to Table 2). Potential energy functions of the interactions computed by Grape are as follows:

Van der Waals (VDW): $$\sum_{i>j}\left(\frac{a_{ij}}{R_{ij}^{12}}-\frac{b_{ij}}{R_{ij}^{6}}\right)$$

Coulomb: $$\sum_{i>j}\frac{q_i q_j}{R_{ij}}$$

Ewald-real-space: $$\sum_{i>j}\frac{q_i q_j}{R_{ij}}\mathrm{erfc}\left(\frac{R_{ij}}{\eta}\right)$$

Ewald-k-space:

$$\frac{1}{2}\sum_{\mathbf{k}}\frac{\exp\left(-\left(\frac{\pi\eta k}{L}\right)^2\right)}{\left(\frac{\mathbf{k}}{L}\right)^2}\left\{\left(\sum_i q_i\cos\frac{2\pi\mathbf{k}\bullet\mathbf{r}_i}{L}\right)^2+\left(\sum_i q_i\sin\frac{2\pi\mathbf{k}\bullet\mathbf{r}_i}{L}\right)^2\right\}$$

The analytical gradients of the above potentials (forces) are also computed by Grape. See reference [2] for details of the implementation.

4. Application of PEACH-GRAPE system

In this section, we review important MD simulations of biological molecules performed by the Peach-Grape system. Among our simulation studies listed in Table 1, those of (1) Hin recombinase/*hixL* complex, (2) *trp*-Repressor/Operator complex, and (3) Calmodulin (CaM) are presented in this review.

The simulations were performed using several features of the Peach-Grape system (Table 2). All the studies reviewed in this article were conducted by using Peach ver. 3.0a (2000) or older, because the capability to use Grape was abandoned after Peach ver. 3.3 (2000). All the simulations were conducted in an explicit solvent, the system size being ca. 10,000-40,000 particles. The simulations were performed in a periodic boundary condition with the Ewald summation of the electrostatic interaction. RESPA, a multiple time-step-method, was used to save computation time without sacrificing accuracy. The simulations were conducted for a few nanoseconds. Amber94 and Amber96 force fields were used [37]. NVE ensemble and Nose-Hoover's NTV ensemble were employed. The computation time required was 0.5-10 s per 1 fs MD.

4.1 Hin/*hixL* complex

The Hin-recombinase peptide/*hixL* DNA complex is a molecular complex formed by the C-terminal 52mer peptide of the Hin-recombinase protein and the half site *hixL* (13 bp), the cognitive DNA. The crystal structure (Fig. 4A) showed that the peptide recognizes the DNA via its helix-turn-helix (HTH) motif inserted in the major groove and by two termini inserted in the minor groove. The *hixL* DNA adopted a conformation close to the canonical B-form, but the minor groove was wider than would be expected from its AT-rich sequence. Some experiments showed that the deletion of the two N-terminal residues resulted in complete loss of the binding

activity, whereas deletion of the eight C-terminal residues resulted only in a reduced affinity.

The Hin/*hixL* DNA complex is regarded as a model system of DNA recognition by a protein. MD simulations were performed to characterize the conformational change upon formation of the peptide/DNA complex [22] [23].

We adopted a simple strategy to analyze the conformational dynamics. The strategy and the simulation results are illustrated in Fig. 4. All the simulations were started from the crystal structure (Fig. 4A). Simulations of the free peptide (B), the free DNA (C), and the complex (D) were conducted in an explicit solvent. The conformational dynamics of the peptide and DNA in the free and complexed forms were compared.

The peptide showed a drastic conformational difference between the free form and the complexed form. The N-terminal region and HTH motif of the peptide were stabilized largely by the interaction with the DNA. The C-terminus was also stabilized by the complex formation, but only marginally.

In contrast to the peptide, the effect of the bound peptide on the DNA conformation was small. However, the bound peptide was suggested to enlarge the minor groove and to slightly distort the DNA from the canonical B-form toward the A-form. Such distortion of the DNA upon DNA-recognition has been found to be important in some other DNA-binding proteins [38].

The simulations also gave important insight into the DNA-peptide interaction. The central major groove was rigidly fixed by the interaction with the HTH motif. The specific interaction between the peptide and the DNA should be mostly governed by the N-terminus and the HTH motif via numerous H-bonds, including solvent-mediated ones. Many of the H-bonds in the DNA-peptide interface existed in a dynamic equilibrium among several binding sites. The C-terminus should participate in non-specific DNA binding by loosely associating with the minor groove through the electrostatic interaction (see [23] for details).

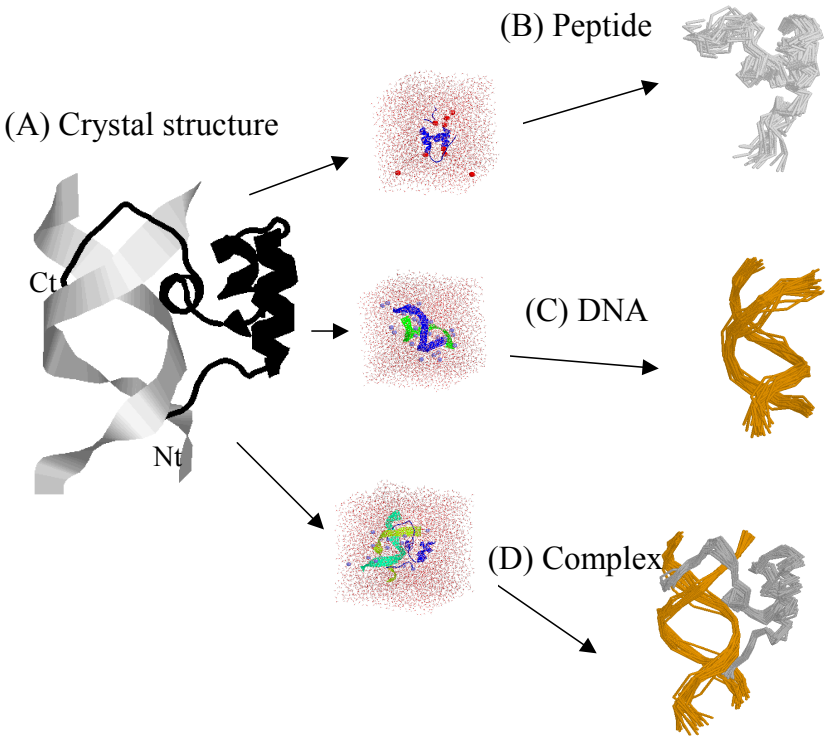
The conformational changes associated with the complex formation are schematically summarized in Fig. 5. Then, the implications from the MD simulations were compared with available experimental data. The simulation results were consistent with available experimental data (Table 3). Namely, the N-terminus is the most important in DNA recognition, whereas the C-terminus was only supportive. Also, the bound peptide enlarges the minor groove and slightly distorts the DNA from the B to the A form.

Thus, in summary, the high precision MD simulations of the Hin/*hixL* complex and its components showed detailed conformation changes associated with the complex formation.

Table 3. Correspondence between MD and experiments about the complex formation.

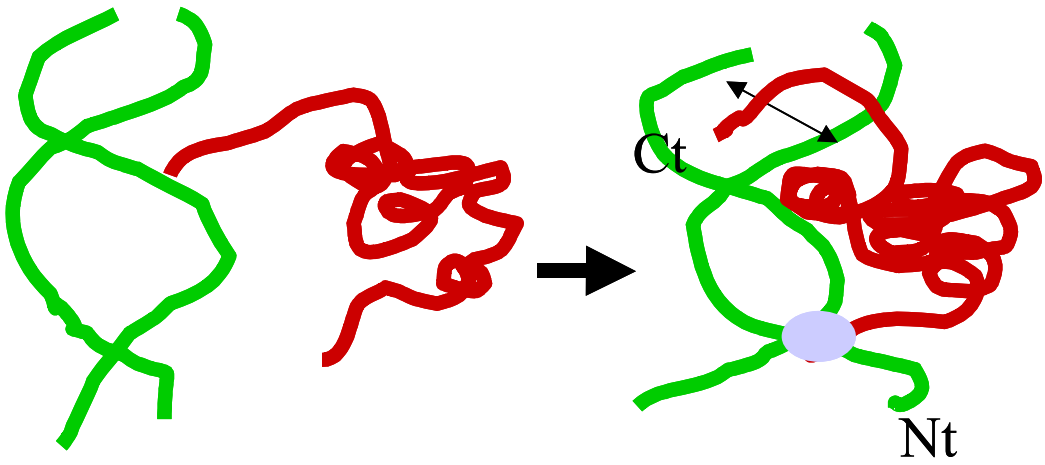
	MD	Experiment
PEPTIDE	N-terminus was firmly attached to DNA, while C-terminus was only loosely associated	Deletion of the N-terminus resulted in complete loss of DNA recognition, whereas that of the C-terminus did not [39] [40].
DNA	The minor groove was widened and the DNA conformation was slightly distorted from B to A.	The DNA in the crystal structure showed unusually wide minor groove [41].

Fig. 4. Strategy and results of the simulations of the Hin/hixL complex.



The crystal structure of the hin/hixL complex was used as the initial structure. The peptide, DNA, and complex were simulated in solvent boxes with counter ions. The simulated structures were superimposed. Reproduced from reference [23] with permission from the publisher.

Fig. 5. Schematic summary of the simulations the Hin/*hixL* complex.



Appendix: Artifacts of the electrostatic cutoff

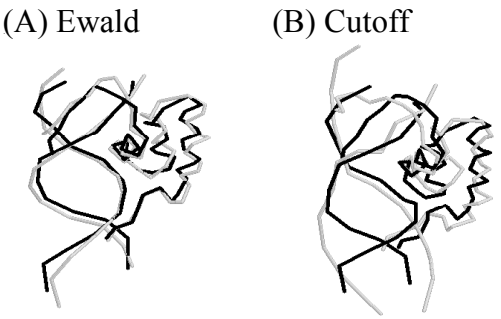
The most prominent feature of the Peach-Grape system is the use of a 'no cutoff" method for computation of the electrostatic interaction. Hence, we compared the cutoff and 'no cutoff' (Ewald summation) methods [22] prior to the production MD runs presented above.

The cutoff artifact was first demonstrated by comparing the crystal and simulated structures (Fig. 6). The simulated trajectory fluctuated within the realm of the crystal structure in the Ewald simulation (A), but the initial structure was largely destroyed in the cutoff simulation (B). This destruction was more prominent in the DNA portion of the complex, presumably because of the highly charged nature of the molecule.

Next, the temperature separation was examined. It is now generally known that solute and solvent have different temperatures when the abrupt cutoff of the electrostatic interaction is used [13]. As shown in Table 4, no temperature separation was observed in the Ewald method, but the separation was prominent in the cutoff method. Unlike the usual cases, the solute had a higher temperature than the solvent. Also, the temperature of the ions was far above the average of 300 K. This should be attributable to the charge of the ions and the solute molecules; specifically, the abrupt cutoff gave more noise to the ions and solutes than to the solvent. Because the total temperature was fixed at 300 K, the relative temperature of the solvent was lower than that of the solute and ions.

In summary, the Ewald method was shown to produce a far more stable and equilibrated trajectory than the cutoff method does, both from structural and thermodynamic viewpoints.

Fig. 6. Comparison of the Ewald and cutoff methods.



Simulated structures (grey) were superimposed to the crystal structure (black). Reproduced from reference [22] with permission from the publisher.

Table 4. Comparison of temperatures (K) between Ewald and Cutoff methods.

	Ewald	Cutoff
Solute	302.1	331.1
Ion	301.1	486.2
Solvent	301.0	297.0

Reproduced from reference [22] with permission from the publisher.

4.2 *trp*-Repressor/Operator complex

The Peach-Grape system was used to simulate another polypeptide/DNA complex: the *trp*-Repressor/Operator complex [24]. The H-bonds in the protein-DNA interface were investigated in detail.

The crystal structure of this molecular complex is shown in Fig. 7A. The H-bond pattern in the protein-DNA interface is also presented, including the water-mediated H-bonds (Fig. 7B).

It was a large complex, and a solvent box as large as 75 x 75 x 75 Angstrom^3 was necessary to solvate it (not shown). In total, 39,956 atoms were included in the simulation. In the preparation of the initial structure, all the crystal water molecules were deleted. Then, water molecules were generated randomly within the box. A stable MD trajectory was generated for 800 ps, and the trajectory was subjected to analysis of the H-bond pattern within the protein/DNA interface.

We examined whether the crystallographic H-bonds, including the water-mediated ones, would reform in MD. The H-bond patterns in the MD structure were compared with those in the X-ray and NMR structures.

As shown in Table 5, most of the H-bonds present in the X-ray and NMR structures were reproduced by our MD simulations. The role of water-mediated H-bonds in DNA recognition has recently been attracting the attention of scientists [42]. The fact that the water-mediated H-bonds formed spontaneously in our simulation indicates their importance in DNA recognition.

Fig. 7. Crystal structure of the *trp*-repressor/operator complex.

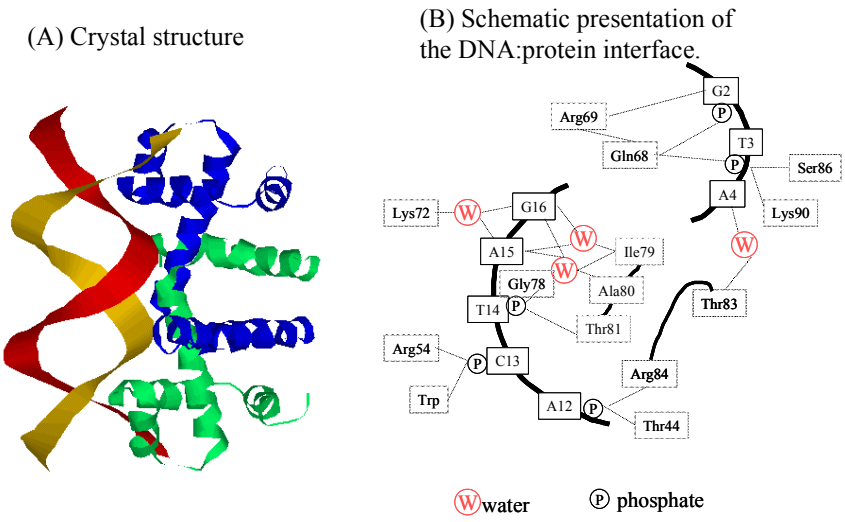


Figure (B) was reproduced from reference [24] with permission from the publisher.

Table 5. H-bond pattern in the repressor-operator interface.

Residue	X-ray	NMR	MD
Arg69	G2	G2, T2	G2
Lys72	G16W	G16W	A15W, G16W
Ile79	G16W	A15W, G16W	A15W, G16W
Ala80	A15W, G16W	A15W	A15W, G16W
Thr83	A4W	A4W	A4W

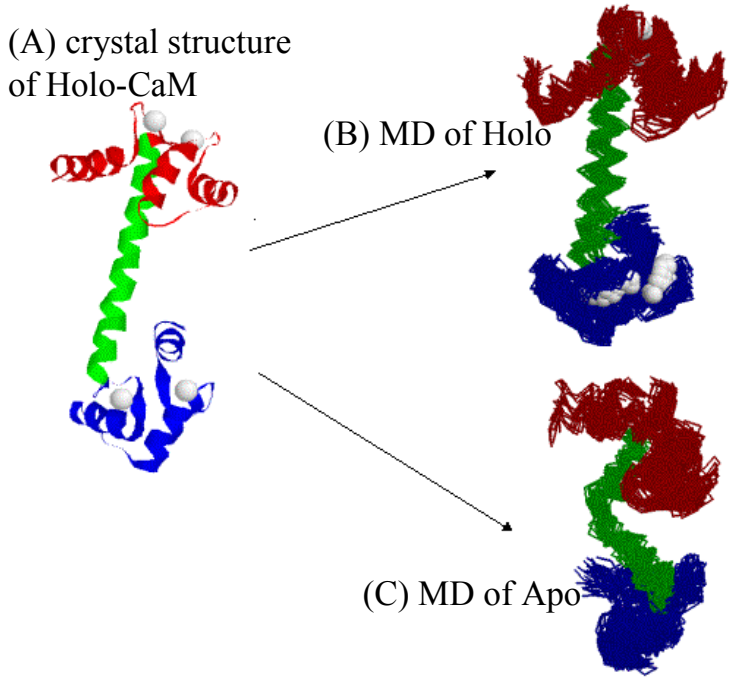
Water-mediated H-bonds were labeled with 'W.' Reproduced from reference [24] with permission from the publisher.

4.3 Calmodulin

The final study performed by using the Peach-Grape system was the analysis of the Ca^{2+}-dependent conformational change of Calmodulin (CaM) [3].

Calmodulin (CaM) is a ubiquitous protein mediating signal transduction by Ca^{2+} ions. A molecule of the Ca^{2+}-free form of CaM (Apo-CaM) binds four Ca^{2+} ions, and the resultant fully bound form (Holo-CaM) changes its conformation in order to interact with various target proteins. In crystal structures, Holo-CaM adopts a dumbbell-like shape (Fig. 8A). CaM comprises three domains: the N-terminal lobe (N-lobe, red), the C-terminal lobe (C-lobe, blue), and the central helix (green).

Fig. 8. Strategy and results of the simulations of Calmodulin.



Simulations were conducted in explicit solvation (not shown).
Reproduced from reference [3] with permission from the publisher.

We investigated whether MD simulation could reproduce the Ca^{2+}-dependent conformational change of CaM. Again, we adopted a simple strategy. The strategy and the results are illustrated

in Fig. 8. We started from the X-ray crystal structure of Holo-CaM (Fig. 8A). Two simulations were performed, with and without the bound Ca^{2+} ions. The former was the trajectory of Holo-CaM (Fig. 8B), while the latter was intended to simulate Apo-CaM (Fig. 8C). Both were performed in an explicit solvent (not shown).

The simulations resulted in striking differences between Holo and Apo forms. The central helix was almost straight in the former but was largely bent in the latter. This difference was all the more interesting, because the central helix was not in direct contact with the Ca^{2+} ions. Hence, this was one of the rare cases in which MD reproduced an allosteric conformational change induced by ions.

The gyration radii (R_g) were computed from the trajectories and then compared with those from the SAXS experiments. The experimental and computed R_g values are listed in Table 6. The experimental values differ from paper to paper, but all of the SAXS results showed that the R_g of Holo-CaM is larger than that of Apo-CaM. The relative difference in R_g was reproduced by the MD trajectories. Thus, the simulations were consistent with the Ca^{2+}-dependent elongation of the CaM molecule observed in SAXS experiments.

In summary, the MD trajectories of CaM showed important aspects of this molecule.
(1) A large conformational difference exists between Holo-CaM and Apo-CaM.
(2) The bound Ca^{2+} ions stabilize the structure of Holo-CaM.
(3) The central helix is inclined to bend, but the bound Ca^{2+} ions interfere with extensive bending via an allosteric effect.

In a living body, Apo-CaM is activated by Ca^{2+} ions, and the resultant Holo-CaM activates various enzymes. Our study suggested that the main role of the Ca^{2+} ions is fixation of the protein conformation. Namely, in the absence of Ca^{2+} ions, the protein should be too flexible to make specific interactions with other proteins.

Table 6. R_g values (Angstrom) for CaM determined by experiments and MD.

	Holo-CaM	Apo-CaM	R_g(Holo)- R_g(Apo)
SAXS [a]			
Seaton et al. [43]	21.5±0.2	20.6±0.6	0.9
Heidorn & Trewhella [44]	21.3±0.2	19.6±0.1	1.7
Matsushima et al. [45]	21.5±0.3	20.9±0.3	0.6
Kataoka et al. [46]	20.2±0.1	19.5±0.1	0.7
X-ray Crystallography			
Chattopadhyaya et al. [47]	21.9		
MD			
This study	21.9±0.4	20.4±0.4	1.5

Reproduced from reference [3] with permission from the publisher.

5. Summary

We have reviewed our studies related to the Peach-Grape system. The system was shown to perform high-precision MD simulations of biological molecules at a reasonable cost. The system was used to perform practical simulations of several proteins and DNA/protein complexes and gave significant insight into their conformational dynamics and molecular recognition. Thus the Peach-Grape project was fruitful.

Nevertheless, in the field of biological simulation, the volume of scientific results obtained by Grape and related computers is rather small (Table 1) compared with the money and human effort spent in the development of those computers. We also know that many of those computers were sold but not effectively used.

To improve this situation, we would like to bring three important points for the effective use of the special-purpose computers. These points should be considered before obtaining a special-purpose computer:

(1) Is it possible to use your choice of algorithm?

Special purpose computers are fast because they are customized for certain purposes. Therefore, they can handle only a limited number of algorithms.

(2) Is the Price/Performance ratio really lower than ordinary computers?

Do not rely too much on the value of Flops. Fast algorithms such as Particle-mesh-Ewald [48] or Fast multipole method [49] could perform MD faster than a brute force type algorithm. Hence, you have to carefully estimate the computation time required for each step of MD with and without the special computer, by an algorithm of your own choice.

This is a tough test for the special-purpose computers because inexpensive PC clusters are available nowadays. Yet one advantage of the special-purpose computers over the PC clusters may be the smaller space and less electricity per performance needed to set and drive them.

(3) Can you develop a program?

You may have to modify your simulator program to implement the special computer. The development could take weeks or months. Support from the hardware vendor is of critical importance for such program development. It is best if you can use a program supplied by the vendor.

We have stopped working on further improvements of the Peach-Grape system mainly because we have shifted our interest from classical MD to quantum MD. Peach is now used as an ordinary MD package independent of Grape [50], and also as a platform for development of a new *ab initio* MD simulation method [51]. In conclusion, we would like to emphasize that special-purpose computers should be evaluated solely by the scientific work they produce, not by their computational performance.

The Peach software is available for free via Internet [52].

We thank following coauthors of the original papers of the Peach-Grape system: Professor Daiichiro Sugimoto (now in the University of Air), Dr. Makoto Taiji (now in RIKEN), and Dr. Toshiyuki Fukushige of the University of Tokyo, Mr. Ryo Takata, Mr. Akihiro Shimizu, and Mr. Keiji Itsukashi of ITL Corp., Professor Ichiro Yamato, Ms. Chieko Yatsu, Dr. Atsushi Suenaga (now in RIKEN), and Dr. Toshiyuki Meguro of Tokyo University of Science, and Dr. Hiroshi Yokoyama, Dr. Kazuaki Harata, Dr. Yutaka Ueno, and Dr. Tadashi Nemoto of AIST. We are grateful to the late Professor Peter Kollman of the University of California at San Francisco for giving us permission to redistribute Amber force field files.

References

[1] Y. Komeiji, H. Yokoyama, M. Uebayasi, M. Taiji, T. Fukushige, D. Sugimoto, R. Takata, A. Shimizu, and K. Itsukashi, in Pacific Symposium on Biocomputing '96 (L. Hunter & T. Klein eds.), pp. 472-487, World Scientific, Singapole (1995).

[2] Y. Komeiji, M. Uebayasi, R. Takata, A. Shimizu, K. Itsukashi, and M. Taiji, *J. Comput. Chem.*, **18**, 1546-1563 (1997).

[3] Y. Komeiji, Y. Ueno, and M. Uebayasi, *FEBS Lett.*, **521**, 133-139 (2002).

[4] T. Ito, J. Makino, T. Ebisuzaki, and D. Sugimoto, *Comput. Phys. Commun.*, **60**, 187- 194 (1990).

[5] D. Sugimoto, Y. Chikada, J. Makino, T. Ito, T. Ebisuzaki, and M. Umemura, *Nature*, **345**, 33-35 (1990).

[6] J. Makino, E. Kokubo, and M. Taiji, *Publ. Astron. Soc. Jpn.*, **45**, 349-360 (1993).

[7] Y. Ohno, J. Makino, I. Hachisu, M. Ueno, T. Ebisuzaki, D. Sugimoto, and S. K. Okumura, *Publ. Astron. Soc. Jpn*., **45**, 377-392 (1993).

[8] P. Hut and J. Makino, *Science*, **283**, 501-505 (1999).

[9] T. Ito, T. Fukushige, J. Makino, T. Ebisuzaki, S. K. Okumura, D. Sugimoto, H. Miyagawa, and K. Kitamura, *PROTEINS*, **20**, 139-148 (1992).

[10] T. Fukushige, J. Makino, T. Ito, S. K. Okumura, T. Ebisuzaki, and D. Sugimoto, *Publ. Astronom. Soc. Jpn.*, **45**, 361-375 (1993).

[11] S. Toyoda, H. Miyagawa, K. Kitamura, T. Amisaki, E. Hashimoto, H. Ikeda, A. Kusumi, and N. Miyagawa, *J. Comput. Chem*., **20**, 185-199 (1999).

[12] http://www.fujixerox.co.jp/nbc/esradd/MDEngineII/

[13] K. Oda, H. Miyagawa, and K. Kitamura, *Mol. Simul*., **16**, 167-177 (1996).

[14] N. Okimoto, T. Tsukui, M. Hata, T. Hoshino, and M. Tsuda, *J. Am. Chem. Soc*., **121**, 7349-7354 (1999).

[15] N. Okimoto, T. Tsukui, K. Kitayama, M. Hata, T. Hoshino, and M. Tsuda, *J. Am. Chem. Soc*., **122**, 5613-5622 (2000).

[16] N. Okimoto, K. Kitayama, M. Hata, T. Hoshino, and M. Tsuda, *J. Mol. Struct.: Theochem*., **543**, 53-63 (2001).

[17] N. Futatsugi and M. Tsuda, *Biophys. J*., **81**, 3483-3488 (2001).

[18] Y. Takaoka, M. Pasenkiewicz-Gierula, H. Miyagawa, K. Kitamura, and A. Kusumi, *Biophys. J.*, **79**, 3118-3138 (2000).

[19] S. Ohta-Iino, M. Pasenkiewicz-Gierula, T. Takaoka, H. Miyagawa, K. Kitamura, and A. Kusumi, *Biophys. J*., **81**, 217-224 (2001).

[20] T. Fukushige, M. Taiji, J. Makino, T. Ebisuzaki, and D. Sugimoto, *Astrophys. J*., **468**, 51-61 (1996).

[21] http://www.gazogiken.co.jp/md_one.html

[22] Y. Komeiji and M. Uebayasi, *Mol. Simul*., **21**, 303-324 (1999a). http://www.tandf.co.uk/

[23] Y. Komeiji and M. Uebayasi, *Biophys. J*., **77**, 123-138 (1999b).

[24] A. Suenaga, C. Yatsu, Y. Komeiji, M. Uebayasi, T. Meguro, and I. Yamato, *J. Mol. Struct*., **526**, 209-218 (2000).

[25] T. Nemoto, M. Uebayasi, and Y. Komeiji, *CBIJ*., **2**, 32-37 (2002).

[26] M. Takano, T. Yamato, J. Higo, A. Suyama, and K. Nagayama, *J. Am. Chem. Soc*., **121**, 605-612 (1999).

[27] T. Narumi, R. Susukita, T. Koishi, K. Yasuoka, H. Furusawa, A. Kawai, and T. Ebisuzaki, SC2000, Dallas (2000).
[28] http://mdm.riken.go.jp/
[29] T. Narumi, R. Susukita, T. Ebisuzaki, G. McNiven, and B. Elmegreen, *Mol. Simul.*, **21**, 401-415 (1999).
[30] T. Narumi, R. Susukita, H. Furusawa, T. Ebisuzaki, in Proc. 5th Int. Conf. on Signal Processing, Beijing, pp. 575-582 (2000).
[31] T. Narumi, A. Kawai, and T. Koishi, SC2001, Denver (2001).
[32] P. K. Weiner and P. A. Kollman, *J. Comput. Chem*., **2**, 287-303 (1981).
[33] M. Saito, *Mol. Simul*., **8**, 321-333 (1992).
[34] M. B. Tuckerman, B. J. Berne, and G. J. Martyna, *J .Chem. Phys.*, **97**, 1990-2001 (1992).
[35] S. Nose, Prog. *Theor. Phys. Suppl*., **103**, 1-46 (1991).
[36] F. Zhang, *J. Chem. Phys*., **106**, 6102-6106 (1997).
[37] W. D. Cornell, P. Ceiplak, C. I. Layly, I. R. Gould, K. M. Merz, D. M. Ferguson, D. C. Spellmeyer, T. Fox, J. W. Caldwell, and P. A. Kollman, *J. Am. Chem. Soc*., **117**, 5179-5197 (1995).
[38] C. A. Bewley, A. M. Gronenborn, G. M. Clore, *Annu. Rev. Biophys. Biomol. Struct*., **27**, 105-131 (1998).
[39] J. P. S. Sluka, S. J. Horvath, A. C. Glasgow, M. I. Simon, and P. D. Dervan, *Biochemstry*, **29**, 6551-6561 (1990).
[40] D. P. Mack, J. P. Sluka, J. A. Shin, J. H. Griffin, M. I. Simon, and P. B. Dervan, *Biochemisty*, **29**, 6561-6567 (1990).
[41] J.- A. R. Feng, R. C. Johnson, and R. E. Dickerson, *Science*, **263**, 348-355 (1994).
[42] J. Schwabe, *Curr. Opin. Struct. Biol*., **7**, 126-134 (1997).
[43] B. Z. Seaton, J. F. Head, D. M. Engelman, and F. M. Richards, *Biochemistry*, **24**, 6740-6743 (1985).
[44] D. B. Heidorn and J. Trewhella, *Biochemistry*, **27**, 909-915 (1988).
[45] N. Matsushima, Y. Izumi, T. Matsuo, H. Yoshino, T. Ueki, and Y. Miyake, *J. Biochem*., **105**, 883-887 (1989).
[46] M. Kataoka, J. F. Head, B. A. Seaton, and D. M. Engelman, *Proc. Natl. Acad. Sci. USA*, **86**, 6944-6948.
[47] R. Chattopadhyaya, W. E. Meador, A. R. Means, and F. A. Quiocho, *J. Mol. Biol*., **228**, 1177-1192 (1992).
[48] T. A. Darden, D. M. York, L. G. Pedersen, *J. Chem. Phys*., **98**, 10089-10092 (1993).
[49] L. Greengard, *Science*, **265**, 909-914 (1994).
[50] Y. Komeiji, M. Haraguchi, and U. Nagashima, *Parallel Comput*., **27**, 977-987 (2001).
[51] Y. Komeiji, T. Nakano, K. Fukuzawa, Y. Ueno, T. Nemoto, M. Uebayasi, D. G. Fedorov, and K. Kitaura, submitted (2002).
[52] http://staff.aist.go.jp/y-komeiji/