JCTC Journal of Chemical Theory and Computation

Essential Dynamics: A Tool for Efficient Trajectory Compression and Management

Tim Meyer,[†,▼] Carles Ferrer-Costa,[†,▼] Alberto Pérez,[†] Manuel Rueda,[†] Axel Bidon-Chanal,[‡] F. Javier Luque,[‡] Charles. A. Laughton,[§] and Modesto Orozco*[,†,||,⊥,#]

Institut de Recerca Biomèdica Barcelona, Parc Científic de Barcelona, Josep Samitier 1-5, Barcelona 08028, Spain, Departament de Fisicoquímica, Facultat de Farmacia, Universitat de Barcelona, Avgda Diagonal 643, Barcelona 08028, Spain, School of Pharmacy and Centre for Biomolecular Sciences, University of Nottingham, Nottingham NG7 2RD, U.K., Departament de Bioquímica i Biologia Molecular, Diagonal 645, Barcelona 08028, Spain, Computational Biology Program, Barcelona Supercomputing Center, Jordi Girona 31, Edifici Torre Girona, Barcelona 08028, Spain, and Bioinformatics Structural Node, Instituto Nacional de Bioinformática, Josep Samitier 1-5, Barcelona 08028, Spain

Received November 21, 2005

Abstract: We present a simple method for compression and management of very large molecular dynamics trajectories. The approach is based on the projection of the Cartesian snapshots collected along the trajectory into an orthogonal space defined by the eigenvectors obtained by diagonalization of the covariance matrix. The transformation is mathematically exact when the number of eigenvectors equals $3N-6$ (N being the number of atoms), and in practice very accurate even when the number of eigenvectors is much smaller, permitting a dramatic reduction in the size of trajectory files. In addition, we have examined the ability of the method, when combined with interpolation, to recover dense samplings (snapshots collected at a high frequency) from more sparse (lower frequency) data as a method for further data compression. Finally, we have investigated the possibility of using the approach when extrapolating the behavior of the system to times longer than the original simulation period. Overall our results suggest that the method is an attractive alternative to current approaches for including dynamic information in static structure files such as those deposited in the Protein Data Bank.

Introduction

Recent advances in algorithms, force-fields, and computer power have greatly promoted the use of molecular dynamics (MD) simulations to gain deeper insight into the structural and dynamical behavior of biomolecules. MD is becoming a standard tool even for experimental groups, and trajectories are being collected for larger systems and for longer simulation times. A search of the *pubmed* server using the keyword "molecular dynamics" found 1608 entries for the period 1992−1994 and 6865 citations for 2002−2004. In addition, while 5 years ago *state-of-the-art* MD typically provided trajectories that covered around 10 ns for biomolecular systems of the size of 100-residue proteins or 12-mer DNAs, today the length of such simulations has increased by nearly 1 order of magnitude, and some groups are turning their attention to far larger systems such as the nucleosome or the ribosome.[1,2] The net result of all this

* Corresponding author e-mail: modesto@mmb.pcb.ub.es.
† Institut de Recerca Biomèdica Barcelona.
‡ Universitat de Barcelona.
§ University of Nottingham.
|| Departament de Bioquímica i Biologia Molecular.
⊥ Barcelona Supercomputing Center.
Instituto Nacional de Bioinformática.
▼ These authors contributed equally to this work.

10.1021/ct050285b CCC: $33.50
Published on Web 02/14/2006

activity is a huge increase in the quantity and quality of available MD data, and how these data can be efficiently stored and retrieved are becoming issues of concern.

The trajectory collected in a MD run consists of a very large file (or a sequence of smaller ones) containing a series of 'snapshots'—the coordinates of the system—over the simulation time. The integration algorithm provides the coordinates of all the atoms in the system every 1−2 fs, but data are output to file much less frequently (typically every 1 ps). Despite this, long trajectories of large systems generate huge data files (many gigabytes in size) that can place a severe burden on disk storage and data transfer systems, because the process of data analysis (which nowadays will usually take much longer than the MD simulation that generated the data) will typically require frequent and high-speed access to the data, very often from remote locations. Additionally, it is increasingly the case that data generated by one research group for one purpose is seen as potentially valuable to another group for another purpose, so questions of enabling the efficient archival and remote retrieval of the data become important. Examples of projects that are facing this issue include the ABC-database (http://max.chem.wesleyan.edu/), the BiosimGrid (www.biosimgrid.org) project, and the MODEL (htpp://mmb.pcb.ub.es/MODEL) project, which involve (i) generating and managing hundreds of very large trajectories for different systems (our group generates nearly 1Tb of trajectory data every month through the MODEL project) and (ii) processing, analyzing, and making available to the scientific community both the analysis and the 'raw' data itself.

In this paper we will present a method that exploits the concept of essential dynamics for the compression and management of large MD trajectories. The method allows a dramatic reduction in the size of the files with no significant loss of quality in the results. Furthermore, the reduction of noise implicit to the use of the method helps in the interpretation of the essential features of the trajectory. The algorithms presented here have been tested using a series of MD trajectories taken from our MODEL database as well as with a very long trajectory of a 28-mer DNA duplex.

Basic Approaches

Essential dynamics (ED) is a very powerful analysis technique[3−6] which exploits principal component analysis to identify the nature and relative importance of the essential deformation modes of a macromolecule from MD samplings. Accordingly, the original Cartesian covariance matrix which contains the atomic positional fluctuations in all 3 coordinate axis about the average structure is diagonalized to obtain a set of eigenvectors and eigenvalues. The eigenvectors describe the nature of deformation movements in Cartesian space, whereas the eigenvalues represent the amount of variance explained by each movement. The eigenvectors define a complete and orthogonal basis set, and accordingly any snapshot in the trajectory can be exactly reproduced in this new $3N-6$ basis set (N is the number of atoms in the system; see eq 1)

$$\{R\}_{x,y,z} \rightarrow \{P\}_{\nu} \tag{1}$$

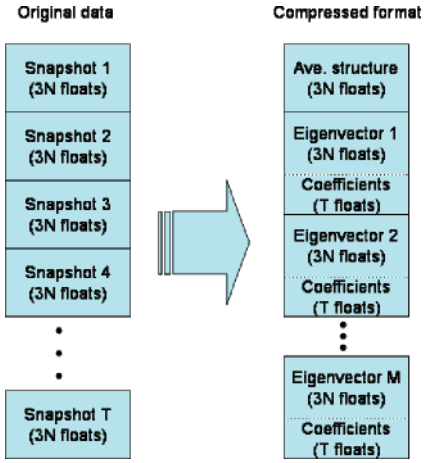

Figure 1. Schematic representation of the data structure of original and compressed trajectories.

where R stands for the original Cartesian (x,y,z) coordinates and P stands for the projections in the $3N-6$ eigenvectors (υ), which are defined to maximize the amount of variance explained in a descending order. Of course, the reverse process is also possible: the original data can be recovered by back-projection from the eigenvectors space to the Cartesian one.

For proteins and nucleic acids the number of important eigenvectors (i.e., those needed to explain 95−99% of the total variance) is much smaller than $3N-6$. If just M eigenvectors describe, say, 95% of the total variance (see below), then projections of the original Cartesian coordinates along the set of important eigenvectors contain nearly all of the original information in a much more compressed way, and it is still possible to regenerate Cartesian coordinate data ($\{R_j\}^n_{x,y,z}$) by back projection (eq 2), though the reconstituted coordinates are no longer exact.

$$\{R_j\}_{x,y,z} \rightarrow \{P_j\}^M_{\nu} \rightarrow \{R_j\}^n_{x,y,z} \tag{2}$$

The opportunities for data compression are obvious. For a set of T snapshots ($T > 3N-6$), the original trajectory file will contain $3NT$ coordinates. This will be transformed (see Figure 1) into a set of M eigenvectors, each of size $3N$, plus M sets of T coefficients—total $M(3N+T)$. For a typical current MD trajectory, reasonable values of N, M, and T would be 500, 50, and 2000, respectively. This would translate into compression of the data to 5.8% of its original size. The question then is what is the cost of this compression—i.e., what is the error between $\{R_j\}^n_{x,y,z}$ and $\{R_j\}_{x,y,z}$.

Possibilities for further data compression also exist. By the quasi-harmonic approximation, the modes of deformation (eigenvectors) associated with the largest eigenvectors are expected to show the lowest frequencies of motion. If the coefficients associated with these modes vary slowly with time (compared to the original snapshot sampling rate), then it should be possible to reduce the M sets of T coefficients to a more sparsely sampled set and regenerate intermediate values by a process of interpolation. Again, the question we seek to address here is to what extent this procedure is useable with 'real life' data.

As a partial aside, we also investigate the utility of this process for data extrapolation, rather than interpolation. We examine to what extent a set of *M* eigenvectors, chosen to be able to capture the dynamic behavior of a system to within a defined tolerance during the time period *T*, are able to continue to represent the system for times beyond *T*. This is not a new idea, forming as it does the basis of the approach of Essential Dynamics,[3−6] but here we provide a detailed analysis based on a wide range of representative systems, of the reliability of this approach.

Practical Derivation of Projections

Covariance matrices were created from equally spaced snapshots collected during long MD trajectories (see above). Following our previous studies,[7] unless specifically noticed, at least $3N-6$ snapshots were collected for each system. Time spacing for data collection ranged from 1 to 10 ps. Once the covariance matrix was defined, eigenvalues were computed using standard algebraic procedures which avoid memory-costly inversion procedures. The percentage of total variance explained by each essential movement is determined according to eq 3, where λ is the set of eigenvalues (in the same distance[2] units in which the covariance matrix is created) and *N* is the number of atoms in the macromolecule. We then determined the minimum number of eigenvectors needed to account for a given amount of variance (generally 95% or 99%), defining an "important space" of *M* eigenvectors which represent the main global movements.

$$\%\ \mathrm{var}_i = 100\frac{\lambda_i}{\sum_{i=1}^{3N-6}\lambda_i} \quad (3)$$

Following *Ptraj* implementation in the AMBER suite of programs,[8] we first derive the eigenvalues using Pal, Walker, and Kahan method,[9,10] whose computational cost scales with the square of the number of atoms. The Arnoldi-Lanczos[9,10] method is then used to find pairs of eigenvectors/eigenvalues in the reduced space (dimension *M*) corresponding to a given amount of variance (determined from the eigenvalues). This latter method is more efficient than the PWK one when a small number of eigenpairs needs to be determined. Finally, the original Cartesian coordinates are projected using the reduced space of eigenvectors (eq 3), which is not strictly exact since $M \ll 3N-6$.

Inter- and Extrapolation of Trajectories

Another goal of this study is to explore the possible use of the preceding procedure to interpolate trajectories, i.e., to estimate a trajectory sampled at a time interval t' from one originally collected with a time interval t and t' and $t' < t$ (eq 4)

$$\{R_j\}_{x,y,z}(t) \rightarrow \{P_j\}_{\nu}^{M}(t) \rightarrow \{P_j\}_{\nu}^{M}(t') \rightarrow \{R_j\}_{x,y,z}^{n}(t') \quad (4)$$

where *t* stands for the time used for storage of the original data, which is used to derive eigenvectors and projections, and t' is the new time spacing ($t'<t$) used to build up the new trajectory.

To this end, we explored the goodness of a simple linear interpolation scheme where a Gaussian noise (Θ) may be introduced to include some stochastic nature in the trajectories (eq 5). The use of the Gaussian noise (always set to define a standard deviation around 10%) helps to reduce an excessive correlation between the interpolated and the original points

$$\{P_j\}_{\nu}^{M}(tt + tt') = \{P_j\}_{\nu}^{M}(tt) + tt'\frac{\{P_j\}_{\nu}^{M}(tt + t) - \{P_j\}_{\nu}^{M}(tt)}{t} + \Theta(tt + tt') \quad (5)$$

where *tt* and *tt*+*t* are trajectory times at which trajectory points were originally collected and *tt*+*tt*′ stands for new times at which the trajectory is interpolated under the constraint that *tt*+*tt*′ pertains to the interval from *tt* to *tt*+*t*.

As discussed above, the eigenvectors obtained from a portion of a trajectory can be used to extrapolate the behavior of the system forward in time, allowing the rapid generation of very long pseudoharmonic trajectories. We explored the validity of this extrapolation scheme by projecting the Cartesian coordinates of a portion of trajectory $t \rightarrow t+\Delta t$ into the set of important eigenvectors obtained from a previous portion of the same trajectory (for example $t\text{-}\Delta t \rightarrow t$). The back-projection procedure generates a new set of Cartesian coordinates, which are then compared with the original ones and with those obtained when the process is repeated using eigenvectors obtained from the same portion of the trajectory (i.e., the period $t \rightarrow t+\Delta t$).

Simulation Details

The compression procedure was first examined using a long (70 ns) trajectory of the DNA duplex d(AAGCATTTTCACG-CATGAGTGCACAGAA). The simulation system contains a total of nearly 82 000 atoms (including water and counterions) and constitutes, to our knowledge, the longest DNA fragment ever simulated over this time scale in explicit solvent. It is therefore an excellent example with which to illustrate the possibilities of the compression procedure and to gain insight into the accuracy of the interpolation and extrapolation schemes outlined above.

The performance of the approach presented here was also examined using 10 ns trajectories of a small set of proteins, which a priori should have more complex dynamics than nucleic acids.[6,11] These were taken from our MODEL database (http://mmb.pcb.ub.es/MODEL). The selected set (PDB entries 1ark, 1cei, 1sro, 2gb1, 3ci2, and 4icb) contains examples of all-α, α+β, and all-β small globular proteins. Finally, we analyzed a long MD simulation (100 ns) of a medium-size protein (PDB entry 1idr) to evaluate the behavior of the method when dealing with long trajectories.

Simulations were performed in all cases using the AMBER parm98[12] force field and the TIP3P[13] model for water and suitable equilibration protocols (from 0.2 to 1 ns). All trajectories were performed in the isothermic-isobaric ensemble (298 K and 1 atm) using Particle Mesh Ewald[14] and truncated octahedral periodic boundary conditions. For the DNA simulation an integration time step of 1 fs was used in conjunction with SHAKE applied to bonds involving

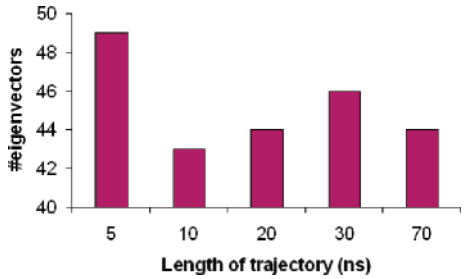

Figure 2. Number of eigenvectors needed to represent 95% of the variance in the trajectory of d(AAGCATTTTCACGCATGAGTGCACAGAA)$_2$ for different simulation times.

hydrogens,[15] while protein simulations were performed using a 2 fs integration time step and SHAKE for all covalent bonds. Simulations were performed using AMBER8.0[8] and NAMD2.5[16] programs on the MareNostrum Cluster (PowerPC64/Myrinet) at the Barcelona Supercomputing Center (details: http://www.bsc.es). All simulations were inspected to verify the lack of equilibration artifacts and that the structural parameters are similar to those determined experimentally. The analysis was performed using a modified version of the *Ptraj* module of AMBER8.0[8] and *in-house* software.

Results and Discussion

DNA Simulations. Our previous studies[6,7,18] have shown that DNA has a quite simple dynamical behavior, which can be represented to a high degree of accuracy by a limited number of eigenvectors. Thus, though the 28-mer duplex considered here has around 1600 atoms, only 44 (220) essential modes are able to capture 95% (99%) of the variance in the 70 ns trajectory of the duplex. These numbers remain quite constant if shorter simulation times are considered, thus revealing that the complexity of conformational space sampled does not increase with the length of the simulation time (see Figure 2).

The average all-atom RMSd between the MD-averaged structure and the 7000 collected snapshots is around 3.0 ± 0.8 Å, with the largest deviations being around 6.5 Å (see Figure 3). When the projection→back-projection procedure is performed using only the first eigenvector (which explains 29% of variance), the RMSd is reduced to 2.5 ± 0.7 Å, and the largest RMSd is close to 6 Å. When the importance space is expanded to the first 5, 44, and 220 eigenvectors (76%, 95%, and 99% of the variance, respectively), the average RMSd is reduced to 1.5 ± 0.2 Å, 0.68 ± 0.06 Å, and 0.30 ± 0.02 Å, and the largest RMSd is below 2 Å when 5 eigenvectors are used and very close to the average error in the other two cases (see Figure 3). Such small errors are impossible to obtain by just taking "representative structures" obtained from clustering analysis of the bidimensional RMSd space. Not surprisingly, the coordinates generated with the compression procedure lead to helical parameters similar to the original ones for the 95% and 99% cutoff levels (see Table 1). The similarity is maintained when the helical

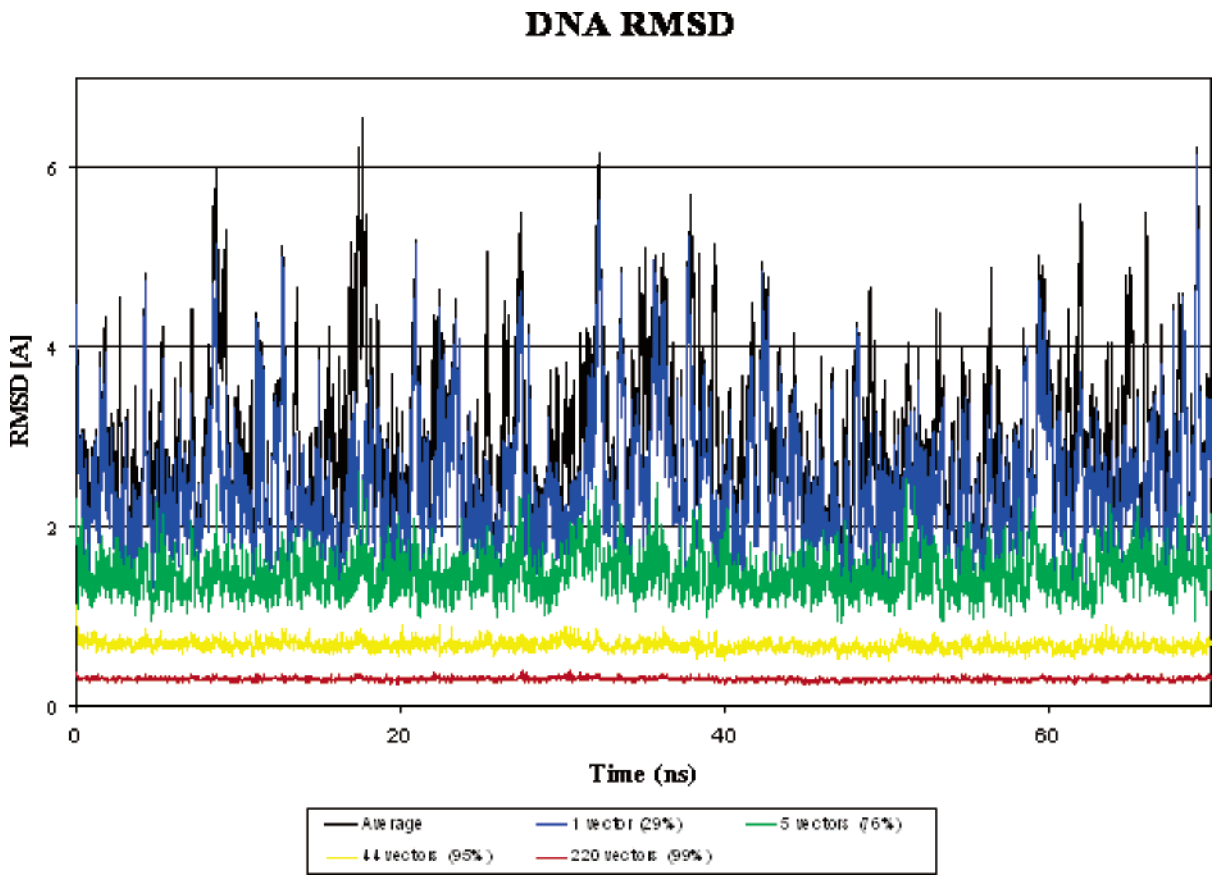

Figure 3. RMSd (in Å) between the real DNA-trajectory and coordinates generated using the projection→back-projection procedure with a different number of eigenvectors (1: blue; 5: green; 95% variance: yellow; 99% variance: red). The results obtained when no eigenvectors are used (average structure) are displayed, for reference, in black.

Table 1. Average Helical Parameters (with Standard Deviations) Associated with the 70 ns Trajectory of DNA Studied Here

helical parameter	original	99% cutoff	95% cutoff
shift	-0.05 ± 0.1	-0.05 ± 0.1	-0.05 ± 0.1
slide	-0.42 ± 0.2	-0.42 ± 0.2	-0.42 ± 0.1
rise	3.34 ± 0.03	3.34 ± 0.03	3.36 ± 0.03
tilt	-0.22 ± 0.6	-0.21 ± 0.6	-0.18 ± 0.2
roll	3.49 ± 1.0	3.49 ± 0.9	3.54 ± 0.9
twist	32.3 ± 0.6	32.3 ± 0.6	32.3 ± 0.6

analysis is performed at the base pair level (see Table S1 in Supporting Information).

As expected, the neglect of fast intramolecular vibrations in the projection→back-projection process generates some deviations of bond lengths and angles from the optimum values and eventually to some incorrect van der Waals contacts. However, these alterations do not affect key intramolecular interactions such as stacking or hydrogen-bond (see Table S2 in Supporting Information). In any case, these artifacts can be easily eliminated by a few cycles of geometry optimization without any significant structural alteration (the average RMSd before and after the optimization is 0.03 ± 0.01 Å). In summary, we can conclude then that for most practical purposes in the field of nucleic acids simulations, original and compressed files provide the same structural information. However, the size of the compressed files is **1.4%** (95% variance cutoff) and **6.4%** (99% variance cutoff) that of the original ones.

In the preceding analysis, the attempt has been made to approximate an entire trajectory as a single set of major eigenvectors perturbing a single average structure. We hypothesized that the use of different reference coordinates and different eigenvector sets for separate sections of the whole trajectory might increase the accuracy of the procedure, especially in cases where radically different conformational states are sampled. However, for the 70 ns trajectory of DNA considered here no relevant gain (average reduction of the RMSd error of the compressed trajectory of 0.03 Å) was obtained when the trajectory was divided in 7 blocks of 10 ns and the projection→back-projection process was repeated using eigenvectors/eigenvalues and average structures were computed for each separate block. Clearly, this situation might change for systems with a less well equilibrated trajectory.

As discussed above, important eigenvectors theoretically define low-frequency movements. This opens up the possibility of using a very aggressive compression procedure where essential movements are recorded for intervals much larger than the original coordinate collection rate, and Cartesian coordinates for intermediate points are regenerated when required by linear interpolation of the projections along the set of important eigenvectors. Results in Table 2 show that the interpolation scheme introduces an additional error in the structures obtained after the projection→back-projection procedure. However, if the interpolation is used within reasonable limits this error might be acceptable for many applications (around 0.1−0.2 Å when 1 ps data is interpolated up to 10 ps samplings). Remarkably, the size of the compressed trajectory is **0.1%** (95% cutoff) that of the original Cartesian one. Further improvements might be made if different spacings were used for collecting data for low- and high-frequency modes, but the investigation of this point falls outside the scope of this article.

Table 2. RMSd (in Å) between Original (Cartesian) and Projected→Back-Projected Coordinates (Determined Using 95% Variance Cutoff) Using Different Interpolation Schemes[a]

interpolation level (spacings)	RMSd (Cartesian)	ΔRMSd
1 ps → 1 ps	0.59 ± 0.04	
2 ps → 1 ps	0.70 ± 0.05	0.11
5 ps → 1 ps	0.72 ± 0.06	0.13
10 ps → 1 ps	0.79 ± 0.08	0.20
20 ps → 1 ps	0.89 ± 0.13	0.40

[a] In all cases the lost of quality (determined as the increase in RMSd relative to that obtained with no interpolation (1ps→1ps)) is indicated. Interpolations were carried out considering a Gaussian noise defined by a standard deviation of 10% the width of the bin.

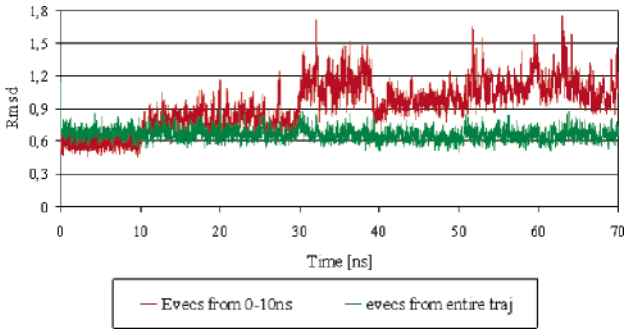

Figure 4. RMSd (in Å) between the real DNA-trajectory and coordinates generated using the projection→back-projection procedure when eigenvectors/eigenvalues and reference structures are obtained using only the first 10 ns of trajectory data. A reference profile is included (in green) indicating the expected errors when eigenvectors/eigenvalues and reference structures are obtained from analysis of the whole trajectory.

Since eigenvectors/eigenvalues describe the movements performed by a molecule along a section of its trajectory, for very long equilibrium trajectories the important movements sampled by a system in the period t-Δt→t should be identical to those sampled in the period t→$t+\Delta t$. Accordingly, in the limit of perfectly equilibrated trajectories the set of eigenvectors/eigenvalues obtained using the sampling collected in the period $[t\text{-}\Delta t,t]$ might be used to extend the trajectory to $[t,t+\Delta t]$. This is the basis of the method known as Essential Dynamics,[3−6] which is of particular interest because MD (or Monte Carlo) simulations in the essential space can be computationally very efficient. However, for the method to be reasonable it is necessary that the set of eigenvectors/eigenvalues obtained from the trajectory over the period $[t\text{-}\Delta t,t]$ can also provide an accurate representation of the essential movements for the trajectory over the period $[t,t+\Delta t]$. To investigate this point we computed the important eigenvectors for the first 10 ns of the trajectory and then used these for the projection→back-projection procedure over intervals 10−20, 10−30,...,10−70 ns. As Figure 4 reveals, the errors related to the use of eigenvectors obtained from a previous segment can be twice as large as those obtained

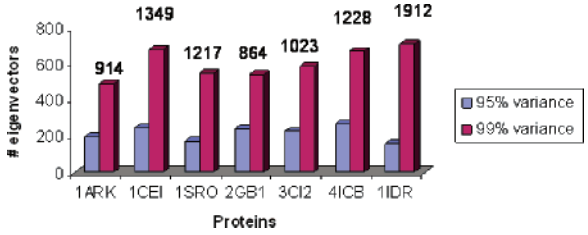

Figure 5. Number of eigenvectors needed to represent 95% or 99% of variance in the protein trajectories considered here. The number of atoms in each protein is displayed on top of the histogram bars.

when the projection→back-projection procedure is performed for the entire trajectory. The particularly mediocre performance of the approach in capturing the behavior of the system during the 30−40 ns period suggests that during this time the system underwent a form of conformational change that was not present in the 0−10 ns period and so was not captured effectively by any eigenvector used. Overall, the discrepancy between extrapolated and real trajectories increases with time, suggesting a time-dependent degradation in the quality of the eigenvectors outside their region of origin. Obviously the errors can be reduced by taking longer simulation periods for the determination of the eigenvectors. For example, if eigenvectors are obtained using the 0−30 ns simulation data, the error in the predicted coordinates over the 30−70 period reduces to 0.8−0.9 Å. In conclusion then, caution is needed in the use of essential dynamics to extend trajectories, since low-frequency movements, which are not well represented in a short-time simulation, are important to trace deformation in distant periods of time

Protein Simulations. The number of eigenvectors needed to represent the essential dynamics of proteins is larger than that of DNA duplexes of similar size. Thus, we need around 200 and between 500 and 700 essential modes to represent 95% and 99% of the variance, respectively (see Figure 5), in other words 3−5 times more eigenvectors than needed for a DNA molecule of similar size. The number of important eigenvectors does not dramatically increase when longer simulation times are used, or when the size of the protein increases (see Figure 5 obtained for 1idr). In any case, the number of important eigenvectors is still much smaller than the number of degrees of freedom of the proteins (between 2600 and 5700 for the proteins considered here), suggesting that compression should be an effective approach to reducing the size of the files.

The average all-atoms RMSd between the MD-averaged structure and the collected snapshots are between 1.2 and 2.4 Å, with the largest point deviations being above 4 Å (see as example Figure 6). When the projection→back-projection procedure is performed using only the first eigenvector (which explains between 20 and 35% of variance), the RMSd between original and back-projected conformations is reduced to 1−3 Å for the 7 proteins considered. When the space is expanded to consider the first 10 eigenvectors (around 50−60% of the total variance) the RMSd is similar or less than 1 Å for all the proteins (see as example Figure 6). The error is reduced to around 0.3 Å (10 ns trajectories) or 0.5 Å (100 ns trajectories) when the

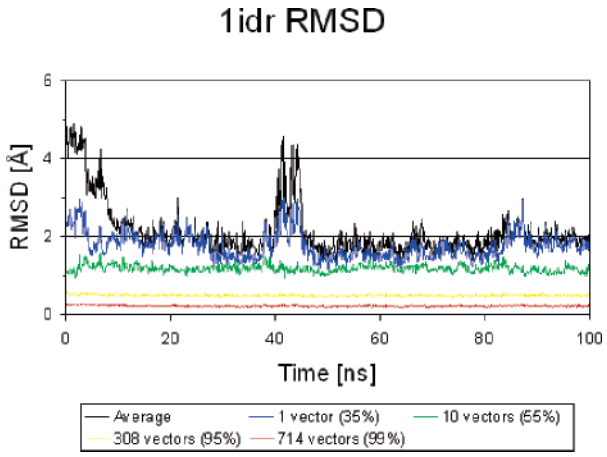

Figure 6. RMSd (in Å) between the real 100 ns trajectory of 1idr and coordinates generates using the projection→back-projection procedure with different number of eigenvectors (1: blue; 10: green; 95% variance: yellow; 99% variance: red). For reference, the results obtained when no eigenvectors are used (average structure) are also displayed (black).

Table 3. RMSd (in Å) between the Original Trajectories and Those Obtained after the Projection→Back-Projection Using a Single Reference Coordinate and the Set of Eigenvectors Necessary To Explain 95 or 99% of the Variance[a]

	95% cutoff		99% cutoff	
protein	RMSd	file size	RMSd	file size
1ark	0.36	8.5	0.15	20
1cei	0.36	7.8	0.16	20
1sr0	0.45	6.0	0.20	18
2gb1	0.29	10.0	0.13	22
3ci2	0.36	8.6	0.16	21
2icb	0.33	8.8	0.14	22
1idr	0.50	5.1	0.22	14

[a] The size of the trajectory file obtained after the projection procedure (% of the original file) is also indicated.

important eigenvectors are defined using a 95% variance cutoff and to around 0.1 (10 ns trajectories) and 0.2 (100 ns trajectory) Å when the 99% variance cutoff is used (see Table 3). As found for DNA, small geometrical errors arising from the neglect of high frequency movements can be easily corrected with a simple minimization protocol (between 20 and 50 energy minimization steps) without alteration of the global structure (RMSd below 0.03 Å).

As noted above, the compression method is exact when all the eigenvectors are considered. However, its computational efficiency should increase as trajectory behaves more harmonically. Thus, we can expect that for trajectories following irreversible transitions "nonequilibrium" trajectories the method will be less accurate. In a practical test we compare two 10 ns segments of a trajectory, the first showing a fast irreversible transition, and the second corresponding to an equilibrium trajectory (see Figure S1 in Supporting Information). Since the first trajectory is dominated by the irreversible transition, the total number of eigenvectors needed to explain a given variance threshold decreases (for example for 99% variance a reduction of 200 eigenvectors;

Table 4. RMSd (in Å) between Original (Cartesian) and Projected→Back-Projected Coordinates (Determined Using 95% Variance Cutoff) Using Different Interpolation Expansions for 1idr (Similar Relative Values Were Obtained for the Other Proteins)[a]

interpolation level	RMSd (Cartesian)	ΔRMSd
1 ps → 1 ps	0.51	
2 ps → 1 ps	0.68	0.17
5 ps → 1 ps	0.83	0.32
10 ps → 1 ps	0.93	0.42
20 ps → 1 ps	1.04	0.53

[a] In all cases the lost of quality (determined as the increase in RMSd from that obtained with no interpolation (1ps→1ps)) is indicated. Interpolations were carried out considering a Gaussian noise defined by a standard deviation of 10% the width of the bin.

see Table S3 in Supporting Information), and the RMSd between the real and compressed files slightly increases (for example for 99% variance from 0.15 to 0.19 Å). However, when the same number of eigenvectors is considered, the performance of the method is almost identical for the two trajectories (see Table S3 in Supporting Information).

In summary, the compression procedure provides a set of coordinates that is nearly indistinguishable (for most purposes) from the original ones. Very interestingly, the size of the compressed files is on average (see Table 3) 8% (95% variance cutoff) and 20% (99% variance cutoff) that of the original trajectories. The reduction becomes more evident for longer trajectories. Additional savings of disk space can be obtained by adding the interpolation procedure outlined above; however, it introduces an additional error which can be too large when it is performed between snapshots too far apart in time. Our results suggest (see Table 4) that a 5→1 ps expansion seems a good compromise between the reduction in the size of the files and the loss of quality in the generated coordinates. Note that this interpolation procedure reduces to 1/5 the size of the projection data, which is the only part of the compressed format which depends on the length of the trajectory.

The use of multiple reference conformations and associated eigenvectors/ eigenvalues is not justified for short (10 ns) trajectories and leads to only a modest increase in the performance of the method for long trajectories. In fact, using 10 sets (10 ns each) of references structures and eigenvectors/ eigenvalues, the RMSd error between original and compressed conformations for the 100 ns trajectories was reduced by 0.1 Å (for both 95 and 99% variance limits). We expect that the multiple-reference strategy may be more effective for more complex trajectories showing large variations in the average structure.

Finally, the use of eigenvectors obtained in a short trajectory fragment to describe the movements in more distant regions of the trajectory leads to non-negligible errors in the back-projected coordinates with respect to the real ones and also to those generated by the usual compression procedure (see Figure 7). It is then clear that both the use of extrapolation techniques based on the sampling of essential movements defined in short simulation times must be done with caution.

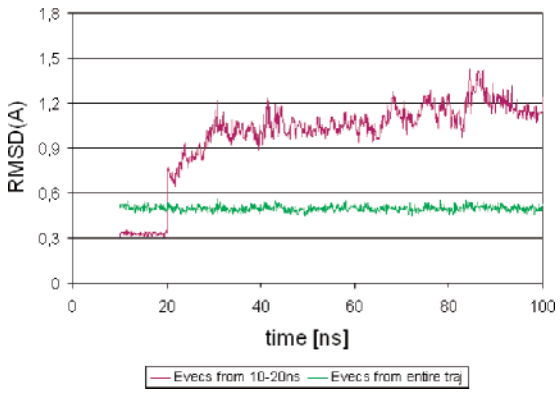

Figure 7. RMSd (in Å) between the real 1idr-trajectory and coordinates generated using the projection→back-projection procedure when eigenvectors/eigenvalues and reference structures are obtained using only the first 10 ns of trajectory data. A reference profile is included (in green) indicating the expected errors when eigenvectors/eigenvalues and reference structures are obtained from analysis of the whole trajectory.

Conclusions

We find that a data compression method based on principal component analysis can work remarkably well with MD trajectory data, permitting files to be reduced to typically less than one tenth of their original size with very acceptable levels of approximation. We would suggest a file format based on this approach configured as follows (Figure 1). The file would begin with the coordinates of the time-averaged structure from the trajectory (3*N* floating point numbers). Next would come the first eigenvector (again 3*N* floats). Next would come the *T* coefficients of this eigenvector over the trajectory. The format then repeats: the second eigenvector then the second time series of coefficients, then the third, etc. The advantage of this approach is that, if one imagines the data being transmitted from one place to another, transmission may be interrupted at any point according to the accuracy required for the regenerated Cartesian coordinates.

Further work remains to be done concerning the possibility for further data compression by allowing interpolation. While in theory the major eigenvectors should relate to low frequency modes, which should be able to be accurately recreated by interpolation between sparse samplings, in practice this is not really the case. A good example of this can be seen in our recent work[19] contrasting the dynamical behavior of a DNA duplex in simulations undertaken with an implicit solvation model compared with those undertaken (as here) with explicit solvent. As Figure 4 in ref 19 shows, the effect of solvent is to contaminate the low-frequency modes with high frequency 'noise' from solvent−solute collisions. In future work we will address the question of whether it is possible to optimize interpolation schemes by a careful analysis of this phenomenon,

Acknowledgment. This work was supported by the Instituto Nacional de Bioinformática (INB-Genoma España), the Spanish Ministry of Education and Science (BIO2003-06848 and SAF2002-04282), and the Barcelona Supercomputing Center.

Supporting Information Available: Analysis of the performance of the compression procedure to reproduce helical parameters stacking and hydrogen bonding in DNA simulations and of the quality of the method to reproduce "nonequilibrium" MD simulations of proteins. This material is available free of charge via the Internet at http://pubs.acs.org.

References

(1) Bishop, T. C. *J. Biomol. Struct. Dyn.* **2005**, *22*, 673−686.

(2) Sanbonmatsu, K. Y.; Joseph, S.; Tung, C. S. *Proc. Natl. Acad. Sci*. **2005**, *102*, 15854−9

(3) Amadei, A.; Linsen, A. B. M.; Berendsen, H. J. C. *Proteins* **1993**, *17*, 412.

(4) Groot, B. I. de; Hayward, S.; van Aalten, D. M. F.; Amadei, A.; Berendsen, H. J. C. *Proteins* **1998**, *31*, 116.

(5) Wlodek, S. T.; Clark, T. W.; Scott, L. R.; McCammon, J. A. *J. Am. Chem. Soc.* **1997**, *119*, 9513.

(6) Orozco, M.; Pérez, A.; Noy, A.; Luque, F. J. *Chem. Soc. Rev.* **2003**, *32*, 350−364.

(7) Noy, A. Meyer, T.; Rueda, M.; Ferrer, C.; Valencia, A.; Perez, A.; de la Cruz, X.; López, J. M.; Luque, F. J.; Orozco, M. *J. Biomol. Struct. Dyn.* **2005**, in press .

(8) AMBER8.0 Computer Program. Case, D. A. et al. University of California, San Francisco, 1999.

(9) Anderson, E.; Bai, Z.; Bischof, C.; Demmel, J.; Dongarra, J.; Du Croz, J.; Greenbaum, A.; Hammarling, S.; McKenney, A.; Ostrouchov, S.; Sorensen, D. *LAPACK Users' Guide*, 2nd ed.; SIAM: Philadelphia, PA, 1995.

(10) Lehoucq, R. B.; Sorensen, D. C.; Yang, C. *ARPACK Users' Guide: Solution of Large Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*; SIAM: Philadelphia, PA, 1997.

(11) Morreale, A.; de la Cruz, X.; Meyer, T.; Gelpí, J. L.; Luque, F. J.; Orozco, M. *Proteins* **2004**, *58*, 101−109.

(12) Cornell, W. D.; Cieplak, P.; Bayly, C. I.; Gould, I. R.; Merz, K.; Ferguson, D. M.; Spellmeyer, D. C.; Fox, T.; Caldwell, J. W.; Kollman, P. A. *J. Am. Chem. Soc.* **1995**, *117*, 11946−11975.

(13) Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R. W.; Klein, M. L. *J. Chem. Phys.* **1983**, *79*, 926−935.

(14) Darden, T. A.; York, D. M.; Pedersen, L. G. *J. Chem. Phys*. **1993**, *98*, 10089−10092.

(15) Ryckaert, J. P.; Ciccotti, G.; Berendsen, H. J. C. *J. Comput. Phys.* **1977**, *23*, 327−341.

(16) Phillips, J.C.; Braun, R.; Wang, W.; Cumbart, J.; Tajkhorshid, E.; Villa, E.; Chipot, C.; Skeel, R.D.; Kale, L.; Schulten, K. *J. Comput.Chem.* **2005**, *26*, 1781−1802.

(17) Ciccotti, J. P. G.; Berendsen, H. J. C. *J. Comput. Phys*. **1977**, *23*, 327−341.

(18) Pérez, A.; Blas, J. R.; Rueda, R.; López-Bes, J. M.; de la Cruz, X.; Orozco, M. *J. Chem. Theory Comput.* **2005**, *1*, 790−800.

(19) Sands, Z. A.; Laughton, C. A. *J. Phys. Chem. B* **2004**, *108*, 10113−10119.

CT050285B