THE ECEPP PACKAGE FOR CONFORMATIONAL ANALYSIS OF POLYPEPTIDES

DANIEL R. RIPOLL[1], ADAM LIWO[2] AND CEZARY CZAPLEWSKI[2,3]

[1]*Cornell Theory Center, Cornell University, Ithaca, NY 14853–3801, U.S.A.*

[2]*Faculty of Chemistry, University of Gdansk, Sobieskiego 18, 80–952 Gdansk, Poland*

[3]*Baker Laboratory of Chemistry and Chemical Biology, Cornell University, Ithaca, NY, 14853–1301, U.S.A.*

Abstract: We present here the ECEPPAK (developed in the laboratory of prof. H.A. Scheraga, Cornell University) and ANALYZE packages for the conformational search of polypeptides that is based on the ECEPP/3 force field. The functions of the program include energy calculation and minimization and global conformational search using the Electrostatically–Driven Monte Carlo (EDMC) method. The search can be constrained using experimental information e.g., the distance constraints from NMR measurements. The sister program, ANALYZE, allows the user to classify the conformations by means of cluster analysis and fit the statistical weights of the conformations to best fit the experimental observables. The package is extensively parallelized, which allows the user to carry out the conformational search even of comparatively large polypeptides in real time.

Keywords: peptides and proteins, conformational analysis, empirical force fields, global optimization

1. Introduction

Theoretical conformational analysis of polypeptides and small proteins plays an increasingly greater role in structural studies of these molecules [1]. The prediction of the structure of new proteins and other biological macromoleculesis the final goal of computational molecular biology and biophysics. The reason for this is that the function of a protein is entirely dependent on its structure. Thus, the knowledge of three–dimensional structure of proteins is necessary for rational interpretation of

their mechanism of action. Experimental techniques, such as X–ray crystallography and NMR spectroscopy provide only a few hundred new structures a year, compared to thousands of new amino–acid sequences. In the energy–based methods, the native structure of a protein is sought as a global minimum of its potential energy function [2, 3].

The situation is in a way more complex in the case of oligopeptides which, in general, exist in an ensemble of conformations [4–6], though conformational search becomes far less expensive, because the molecules are much smaller than proteins. The most important practical application of the conformational analysis of oligo– and polypeptides is searching for conformational properties of possible lead compounds in drug design. Virtually all pharmaceutical companies use molecular modeling in the design of new drugs,which saves expensive chemicals and time required for otherwise blind testing.

It should also be noted that even the techniques considered as the experimental methods for structure determination use theoretical conformational analysis as a part of the process. Most commonly, the interproton distances and coupling constants provided by NMR measurements are used as restraints in conformational search. In the X–ray crystallography, the structure is also obtained by simulated annealing of an energy function constrained by the collected experimental data.

A good search technique is required, both in order to find the global energy minimum of a protein or to find all representative conformers of a peptide. Many different methods have been designed for this purpose and all of them are based on empirical force fields to represent the energy hypersurface of polypeptide chains [2, 3]. Among the first applied to biomolecules is the ECEPP force field (Empirical Conformational Energy Program for Peptides and Proteins) [7–10]. The search techniques most commonly used are based on molecular dynamics, Monte Carlo methods, and genetic algorithms. The last two were implemented into the ECEPP force field. Apart from complete conformational analysis, it is also necessary to develop tools for the classification of conformations, comparison with experimental data, and estimation of the populations of the conformations of the obtained ensembles based on the experimental data.

In this article we describe our ECEPPAK and ANALYZE packages, the first of which does the conformational search using the Monte Carlo with Minimization (MCM) [11, 12] and the Electrostatically–Driven Monte Carlo (EDMC) methods [13–18]. This program is an extended version of the ECEPP program available from QCPE.Classification of conformations and comparison with experimental data is carried out using the program ANALYZE.

2. Molecular systems

The package treats single polypeptide chains. The molecule is input as a sequence of amino–acid residues terminated on both ends by the end groups. These building blocks are stored in a resident database file, which is copied during installation. The database can be further enhanced by the user. The following

extensions were made with respect to the QCPE version of the ECEPP/2 program:

1. The old set of ECEPP input files has been replaced by a more flexible file structure.
2. The main input file contains now a series of cards that define the type of run and parameters.
3. The residue data file has been enhanced. This file contains the ECEPP/3 residues and other non–standard ones. There are 72 residues (including N–methyl residues), and new end groups defined.

   Among the changes introduced in the residue database file are:

   a) Data on loop closing pairs was added. The program uses a general treatment for these pairs, not only for disulfide bridges.

   b) The data base includes N–methyl residues.

   c) Hydration atom types were added in the description of atoms.

   d) Description of 1–4 interactions is included in a more general format.

   e) The atom labels C′ as well as NP in PRO and HPRO were replaced by C and N, respectively, to increase the compatibility with PDB format.

   f) Atom types of protons in COOH groups (Asp, Glu, N–Me–Asp, N–Me–Glu and carboxyl–end terminal) were changed to type 1, (as in ECEPP/3, no H–bonding allowed).

   g) Hydration parameters for different surface and volume models have been included.

Table 1. *Amino–acid residues included in the ECEPPAK data base*

Residue	*ECEPP LIST No.*	*ECEPP KIND*	*3–letters code*	*1–letter code*[a]
Alanine	1	–1	ALA	A
Aspartic acid	2	–2	ASP	D
Cystine	3	–3	CYS	C_
Glutamic acid	4	–4	GLU	E
Phenylalanine	5	–5	PHE	F
Glycine	6	6	GLY	G
Histidine–δ	7	–7	HIS	H
Isoleucine	8	–8	ILE	I
Lysine	9	–9	LYS	K
Leucine	10	–10	LEU	L
Methionine	11	–11	MET	M
Asparagine	12	–12	ASN	N
Proline–down	13	13	PRO	P
Glutamine	14	–14	GLN	Q
Arginine	15	–15	ARG	R
Serine	16	–16	SER	S
Threonine	17	–17	THR	T

Valine	18	–18	VAL	V
Tryptophan	19	–19	TRP	W
Tyrosine	20	–20	TYR	Y
Cysteine	21	–21	CYX	C
Hydroxyproline–down	22	–22	HPD	P<
Norleucine	23	–23	NOR	N<
Ornithine	24	–24	ORN	O
Histidine–ε	25	–26	HIE	H–
Benzyl–aspartate	26	–30	BZD	B<
Ornithine$^+$	27	–25	OR+	O+
Histidine$^+$	28	–27	HI+	H+
Lysine$^+$	29	–28	LY+	K+
Arginine$^+$	30	–29	AR+	R+
Aspartic acid$^-$	31	–31	AS–	D–
Glutamic acid$^-$	32	–32	GL–	E–
Proline–up	33	13	PRU	P%
Azetidin	34	13	AZE	P*
Hydroxyproline–up	35	–22	HPU	P>
Tyrosine$^-$	36	–36	TY–	Y–
γ–Aminobutyric acid	37	–33	ABU	Z<
Aminoisobutyric acid	38	–38	AIB	Z>
Serinola	39	–39	SLA	S<
allo–isoleucine	40	–40	AIL	I*
γ–Aminobutyric acid loop	41	–41	ASU	U<
Sillyxarin'	42	–42	SXY	X
Sllxrayin	43	–43	SLX	X*
Glutamic acid loop	44	–44	GLP	E_
Lysine loop	45	–45	LYP	K_
Diaminobutyric acid loop	46	–46	DAB	B_
Glycine loop	47	47	GYP	G_
Leucine loop	48	–48	LEP	L_
Aspartic acid loop	49	–49	ASX	D_
N–methyl alanine	51	–51	M–A	@A
N–methyl aspartic acid	52	–52	M–D	@D
N–methyl cystine	53	–53	M–C	@C_
N–methyl glutamic acid	54	–54	M–E	@E
N–methyl phenylalanine	55	–55	M–F	@F
Sarcosine	56	–56	SAR	@G
N–methyl histidine	57	–57	M–H	@H
N–methyl isoleucine	58	–58	M–I	@I
N–methyl lysine	59	–59	M–K	@K
N–methyl leucine	60	–60	M–L	@L
N–methyl methionine	61	–61	M–M	@M
N–methyl asparagine	62	–62	M–N	@N
N–methyl glutamine	64	–64	M–Q	@Q

N–methyl arginine	65	–65	M–R	@R
N–methyl serine	66	–66	M–S	@S
N–methyl threonine	67	–67	M–T	@T
N–methyl valine	68	–68	M–V	@V
N–methyl tryptophan	69	–69	M–W	@W
N–methyl tyrosine	70	–70	M–Y	@Y
N–methyl BMT	71	–71	BMT	@Z
N–methyl ornithine	72	–72	MOR	@O

a "@" is used to indicate N-methyl residues."_" is generally used to indicate a bridging residue (e.g., C_ indicates a half-cystine). "+" and '–' are used to indicate a charged residue (e.g., K+ indicates a charged lysine residue).

The amino–acid residues and end groups included in the ECEPPAK data base are listed in Tables 1 and 2, respectively.

3. Force field

The ECEPP/3 [10] force field has been implemented. It assumes rigid valence geometry of polypeptide chains. The total conformational energy, E_{tot}, is expressed as as sum of electrostatic energy, E_{es}, nonbonded energy, E_{nb}, torsional energy, E_{tor}, and loop–distortion energy, E_{loop}, by Equation (1).

$$E_{tot} = \sum_{i<j}\left[\frac{q_i q_j}{\varepsilon\, r_{ij}} + \frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{6}}\right] + \sum_{\text{H-bonded pairs}}\left[\frac{q_i q_j}{\varepsilon\, r_{ij}} + \frac{C_{ij}}{r_{ij}^{12}} - \frac{D_{ij}}{r_{ij}^{10}}\right] + \\ + \sum_{\text{torsional angles}}\sum_{n} V_{ni}\left[1+\cos(n\phi_i - \gamma_{ni})\right] + \sum_{ij,loop} k_{ij}(d_{ij,loop} - d^{\circ}_{ij,loop})^2, \qquad (1)$$

where r_{ij} is the distance between atoms i and j, A_{ij}, B_{ij}, C_{ij}, and D_{ij} are pair–specific constants in the nonbonded potentials, ε is the relative dielectric permittivity, q_i is the partial charge of atom i, ϕ_i is ith torsional angle, n is the multiplicity of a torsional energy term, V_{ni} is the torsional constant of multiplicity n characteristic of ith angle and γ_{ni} is the respective phase angle, $d_{ij,loop}$ is a distance within a loop (between bonded or 1,3–nonbonded atoms), $d^{\circ}_{ij,loop}$ is the corresponding "strainless" value of this distance, and k_{ij} is the force constant.

The solvation energy can be represented using two alternative models: the solvent–accessible surface model, in which the solvation energy is expressed as a sum of contributions from solvent–exposed surfaces of each atom [19] [Equation (2)] and the volume model, in which the solvation energy is expressed in terms of solvation–shell volumes around each atom. In the second case, an approximation has been made to express the solvation–shell volumes in terms of pairwise contributions [20, 21] [Equation (3)].

$$E_{solv} = \sum_i \sigma_i A_i, \qquad (2)$$

$$E_{solv} = \sum_i C_{ij} \exp\left(\frac{-r_{ij}^2}{B_{ij} R_{ij}^2}\right), \tag{3}$$

where A_i is the solvent-accessible surface of atom i, σ_i is the solvation energy per unit surface area of this atom, C_{ij}, B_{ij}, and R_{ij} are the constants characteristic of the volume–solvation contribution from atom pair i and j, r_{ij} is the distance between these atoms.

Table 2. *End groups included in the ECEPPAK data base*

End group	*ECEPP LIST No.*	*ECEPP KIND*	*3–letters code*	*1–letter code*
Amino — H_2	1	1	H2N	H
Amino — H_3^+	2	2	H3N	H+
Amino–CH_3	3	3	CH3	M
Amino–$COCH_3$	4	–4	ACE	A
Formyl	5	–5	FYL	F
End–Pro, cis–H	6	–6	CHP	P–
End–Pro, trans–H	7	–7	THP	P
End–H_2^+–Pro	8	–8	AHP	P+
Pyroglutamic	9	–9	PGL	G
Amino (cyclizing)	10	10	HN–	H_
Carboxyl — COOH	11	–11	CXH	O
Carboxyl — O$^-$	12	12	OCC	O–
Carboxyl–CH_3	13	13	CCC	L
Carboxyl–NH_2	14	–14	NCC	N
Carboxyl–$NHCH_3$	15	–15	NME	C
N, N–dimethyl	16	–16	DME	D
Methyl ester	17	–17	MES	T
Ethyl ester	18	–18	EES	E
Amino–t–Boc	19	–9	BOC	B
Carboxyl (cyclizing)	20	20	CXL	O_
Mpa (half S–S)	21	–21	MPA	R_
Dmp (half S–S)	22	–22	DMP	D_
Cpp_{ax} (half S–S)	23	–23	CPP	C_
Carboxyl–CH_2F	24	24	CHF	S
Oca_{ax} (half S–S)	25	–25	OCA	A_
Oca_{eq} (half S–S)	26	–26	OCE	E_
Sca_{ax} (half S–S)	27	–27	SCA	S_
Sca_{eq} (half S–S)	28	–28	SCE	T_
Cpp_{eq} (half S–S)	29	–29	CPE	F_
Dansyl	30	–30	DAN	W
Carboxyl (dummy)	31	31	CXX	X
Amino–cynamonic	32	–32	CYN	Y

4. The Electrostatically Driven Monte Carlo (EDMC) method for conformational search

A brief description of the Electrostatically–Driven Monte Carlo method (EDMC) is as follows [13–16, 18]. Starting from an arbitrary conformation, a single local energy minimization is carried out, and then a new conformation is generated by perturbing the dihedral angles of the energy–minimized starting conformation. The energy of the perturbed conformation is subsequently minimized. If the new energy–minimized conformation is similar in shape and in energy to the starting conformation, it is discarded. Otherwise, the energy of the new conformation is compared with the energy of the parent conformation. If the new energy is lower, the new conformation is accepted unconditionally, otherwise the Metropolis criterion is applied in order to accept or reject the new conformation. If the new conformation is accepted, it replaces the starting one; otherwise another perturbation of the parent conformation is tried. The process is iterated, until a sufficient number of conformations have been accepted. When the dihedral angles are perturbed in a random way, the method is called Monte Carlo with Minimization, which has been found to be a very efficient method in locating the lowest–energy conformations of small peptides [11, 12]. This method can be made yet more efficient, if a fraction of the dihedral–angle perturbations is aimed at optimizing the alignment of local peptide–group dipoles with the electrostatic field of the remaining part of the molecule; this modification defines the EDMC method. Details of the procedure can be found in the references cited [13–16, 18].

To treat the case of cyclic peptides and the cases in which distance constraints are included, the EDMC method was augmented with a constrained–sampling algorithm developed in an earlier work [22]. This algorithm assures that the constraints (coming from ring closure requirements or user–imposed) significantly distorted while changing the backbone dihedral angles. First, it makes a perturbation of the selected dihedral angle(s) and then, keeping the perturbed angles fixed at the new values, it adjusts the remaining angles using a least–squares procedure so as to satisfy the constraints. Without applying the above–mentioned least–squares procedure, the dominant contributions to energy would frequently come from the harmonic ring–closure potentials and energy minimization would first attempt to reduce these terms at the expense of the other energy contributions. As a result, the minimization could either be trapped in a high–energy local minimum or restore the initial conformation. In both cases the perturbation would be rejected. Therefore, the constrained–sampling algorithm helps to maintain a reasonably high conformation–acceptance rate for cyclic molecules or in the cases in which distance–restraints are applied [22, 23].

5. Including experimental restraints

A distance–restraint energy term can be included in the calculations. The algorithm used in this program represents a modification of the one originally implemented by Vasquez and Scheraga [24]. The functional form of this term is:

$$E_{restr} = \begin{cases} w_e \Sigma_{ij \in \kappa} w_{ij} (d_{ij}^2 - d_{ij}^{up})^2 & \text{if } d_{ij} > d_{ij}^{up} \\ w_e \Sigma_{ij \in \kappa} w_{ij} (d_{ij}^2 - d_{ij}^{low})^2 & \text{if } d_{ij} < d_{ij}^{low} \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

κ is the set of all atom pairs included in distance–restraint list, d_{ij} is the distance d_{ij}^{up} and d_{ij}^{low} are the upper and lower bound of this distance, respectively, w_e is a factor that weights the distance energy term with respect to other energy terms (such as electrostatic, torsional, etc.), and w_{ij} is the weight of the corresponding distance restraint.

6. Fitting X–ray and NMR structures to rigid ECEPP geometry

As pointed out in Section 3, ECEPP assumes rigid valence geometry of the polypeptide chain, the bond lengths and angles being assigned standard values. This causes problems with using X–ray or NMR structures as input conformations for the calculations. Conformations generated by ECEPP/3 using the dihedral angles calculated directly from X–ray or NMR structures usually give large Residual Mean Square Deviations (RMSDs) from the experimental structure. To improve the fit, a series of minimizations of a target function composed of a pseudo–energy term (that prevents overlaps; [25]) and an RMSD term is carried out. The resulting structure can be used as a starting conformation.

7. Clustering

Calculations with the EDMC method produce a large number of (typically a few thousand) conformations. An analysis of such a large set is difficult without applying appropriate tools. For this purpose, cluster analysis has been implemented. Clustering is carried out with the minimal–spanning tree or minimum–variance algorithm [26]. The distance between two conformations (required in the clustering algorithm) is defined as the Root–Mean–Square Deviation (RMSD) between user–specified atoms.

8. Calculation of the statistical weights of the conformations by fitting the theoretical to the experimental NMR data

Nuclear Magnetic Resonance (NMR) and particularly Nuclear Overhauser Effect (NOE) spectroscopy are useful tools for conformational studies of peptides and proteins [27, 29]. A usual procedure is to convert the NOE intensities into interproton distances and implement the latter in a conformational–search method, such as molecular dynamics or EDMC (see Section 5) simulations as distance restraints [27, 29]. While such a procedure is justifiable in the case of proteins, which occur in a well–defined conformation, its application to flexible polypeptides that occur in multiplicity of conformations is not straightforward. In the last case the experimental observables should rather be regarded as conformational averages

[4, 30]. We have therefore developed an approach that fits the weights of the conformations generated by the EDMC or molecular dynamics methods so as to minimize the difference between the measured NOE intensities and coupling constants and the average computed observables [31]. This approach is implemented in the ANALYZE package and uses the MORASS program [32, 33] to compute theoretical NOE integral intensities. The program solves the system of Bloch differential equations for the cross–relaxation of a system of interacting proton spins.

The theoretical NOE intensities are averages over all conformations of the ensemble:

$$\upsilon_{kl} = V_0 \sum_{i=1}^{NC} x_i \upsilon_{ikl}, \quad k,l = 1,2,\ldots,NP , \tag{5}$$

$$x_i \geq 0, \quad i = 1,2,\ldots,NC , \tag{6}$$

$$\sum_{i=1}^{NC} x_i = 1 , \tag{7}$$

where $\bar{\upsilon}_{kl}$ is the integral intensity of the NOE between protons k and l averaged over all conformations, υ_{ikl} is this intensity for conformation i, x_i is the statistical weight (fraction) of the ith conformation, V_0 is a scaling factor, and NP and NC are the number of protons and the number of conformations, respectively.

The vicinal NH—C^{α}H coupling constants corresponding to the ith conformation can be calculated from the empirical Bystrov–Karplus relationship [Equation (8)]:

$$J_{ik} = a_{0k} + a_{1k} \cos\theta_{ik} + a_{2k} \cos^2 \theta_{ik} , \tag{8}$$

where J_{ik} is the coupling constant of kth angle and ith conformation and θ_{ik} is the corresponding angle.

As in the case of NOE intensities, the coupling constants must be averaged over conformations:

$$\overline{J_k} = \sum_{i=1}^{NC} x_i J_{ik} . \tag{9}$$

Thus, the average NOE intensities and the average coupling constants are functions of the weighs x_1, x_2, ..., x_{NC}. The weights could therefore be determined by least–squares fitting of the calculated NOE intensities and coupling constants to the corresponding experimental quantities, as given by Equation (10):

$$\min \Phi(V_o, x_1, x_2, \ldots, x_{NC}, a_{o1}, a_{11}, a_{12}, \ldots, a_{NJ}) =$$
$$\sum_{(k,l)\in\kappa} w_{kl}\left[v_{kl}^{\exp} - \overline{v_{kl}}(V_o, x_1, x_2, \ldots, x_{NC})\right]^2 +$$
$$w_J \sum_{k=1}^{N\theta}\left[J_k^{\exp} - \overline{J_k}(x_1, x_2, \ldots, x_{NC})\right]^2 + \qquad (10)$$
$$\sum_{l=1}^{NJ} \frac{1}{\sigma_{a_{ol}}^2}(a_{ol} - a_{ol}^o)^2 + \frac{1}{\sigma_{a_{1l}}^2}(a_{1l} - a_{1l}^o)^2 + \frac{1}{\sigma_{a_{2l}}^2}(a_{2l} - a_{2l}^o)^2,$$

where κ is the set of all signals considered, w_{kl} is the weight of the intensity of the NOE between protons k and l, w_J is the weight of the coupling–constant term, $N\theta$ is the number of angles for which the coupling constants were determined, NJ is the number of the sets of the constants in the Bystrov–Karplus equation, a^o_{kl} denotes the "standard" value of a_{kl} in the Bystrov–Karplus equation, $\sigma_{a_{kl}}$ is its estimated standard deviation. Including the last sum accounts for the fact that the values of the coefficients in Equation (8) are uncertain within the limits determined by their standard deviations.

Minimization of Φ of Equation (10) usually results in the predominance of only a few conformations, while the weights of the remaining ones are close to zero. At the same time, the experimental quantities are usually overfitted. In order to prevent this, the maximum entropy approach [34] has been implemented. The resulting functional is expressed by Equation (11):

$$\Psi(V_o, x_1, x_2, \ldots, x_{NC}) = \Phi(V_o, x_1, x_2, \ldots, x_{NC}) + \alpha \sum_{i=1}^{NC} x_i \log x_i . \qquad (11)$$

The entropy term reaches its global minimum, if the statistical weights of all conformations are equal. This can be regarded as the reference state, in which no information about the preference of individual conformations is provided. Weight differentiating comes only from the Φ term that includes experimental information. Therefore a common procedure is to choose the coefficient at the entropy term, α, so that the weighted χ^2 value be equal to the number of observations [34], which is equivalent to the requirement that the mean errors in the fitted quantities be comparable with the error estimates. In data analysis it is a natural approach, because the expected "misfit" measures should be equal to the estimated experimental inaccuracy; if the agreement between theory and experiment is better, one starts fitting the noise.

Minimization of Ψ is carried out using the Secant Unconstrained Minimization Solver (SUMSL) routine [35]. Minimization of Φ (which is a sum of squares) is carried out using the Marquardt method [36].

9. The ECEPPAK program

9.1. Functions of the program

ECEPPAK performs the following calculations:

1. A single energy evaluation.
2. A single energy minimization.
3. Energy evaluation of multiple input conformations.
4. Energy minimization of multiple input conformations.
5. Conformational search using the EDMC method [13].
6. Produce an energy map for a pair of dihedral angles.
7. Carry out an RMS deviations analysis.
8. Carry out variable target function calculations for structure determination using distance restraints.

9.2. Structure of input file

The general input to the program is given through a file with a set of instructions. The program uses a parser to read these instructions. The parser reads and interpret the first 78 characters of a line. No distinction is made between lower–case or upper–case letters. The symbols "#" and "!" are used to indicate the beginning of a comment. When any of this symbols are encountered, the parser will ignore the rest of the line. Instructions related to a given procedure are associated into the so called "data groups". A "data group" is identified by a main keyword which contains the symbol "$" as the first character, i.e. $EDMC, $CNTRL. Also the keyword $end or $END, should be present, indicating the end of the data group. Any word included between the main keyword and $end, is considered an instruction. The following is an example of a data group:

```
$CNTRL
runtyp=Energy
$end
```

The following list contains the data groups already defined in ECEPPAK:

1. $BOUNDS — contains the weights of distance constraints corresponding to pairs of atoms of specified types;
2. $BOUND_DEF — contains the control data for a distance–constrained run;
3. $BRIDGE — contains the positions of covalent links (e.g., disulfide bridges);
4. $CNTRL — the control data of the run;
5. $DIST_CONST — contains the distance constraints;
6. $EDMC — the control data for an EDMC run;
7. $FFIELD — force field specification and options;
8. $GEOM — dihedral angle of the template (or initial) conformation;

9.	$GRID	the control data for calculating energy maps;
10.	$MINIM	the control data for energy minimization;
11.	$REGIONS	the control data for region–biased sampling;
12.	$RMSFIT	the control data for RMS deviation calculation;
13.	$SCAN	the control data for the scanning of side–chain conformations;
14.	$SELEC_PDB	the control data to select the reference conformation for RMSD calculations;
15.	$SEQ	the amino–acid sequence;
16.	$SPEC	the specification of the variable dihedral angles;
17.	$ENERCALC	the control data for energy calculations;
18.	$VTF	the control data for variable target function calculations;
19.	$WINDOWS	the control data for sequence–region biased sampling in EDMC;
21.	$OMCIS	specifies which peptide groups should be kept in the cis and which in the trans configuration.

Three of these data groups: $CNTRL, $SEQ and $GEOM are essential and the program will abort, if they are absent.

9.3. Example of calculations (conformational search using the EDMC method)

The EDMC method can be use to produce a search for the lowest–energy conformation of a molecule. The conformational path followed by the method corresponds to a series of energy minima of the potential energy surface. During a very long run, the method is likely to find the conformation corresponding to the global energy minimum (at least, for relatively small sequences with no more than 30 amino acid residues). The description of a simple input file to produce a short test run with only five accepted conformations for a ten–residue chain of L–alanine follows. The generated input file should have a name that uses the suffix "inp" (i.e., `ten_ala_edmc.inp`).

```
$CNTRL
runtyp = edmc              ! run type is a edmc run
PRINT_CART                 ! print Cartesian coordinates
OUTFORMAT =PDB             ! format of the Cartesian file
                             is PDB
FILE = A10edmc             ! prefix of the PDB file is
                             A10edmc
res_code= one_letter       ! use one-letter code to
                             specify input sequence
$END
```

```
$EDMC
MAXIT=20
SEED= -5555
TEMP= 300
THERMAL_SHOCK T_UP = 5000
RAND_TO_ELEC=0.3
$END

$SEQ
A
AAAAAAAAAA
C
$END

$GEOM
 180.000 180.000
-160.000-140.000 180.000  60.000
-160.000-140.000 180.000  60.000
-160.000-140.000 180.000  60.000
-160.000-140.000 180.000  60.000
-160.000-140.000 180.000  60.000
-160.000-140.000 180.000  60.000
-160.000-140.000 180.000  60.000
-160.000-140.000 180.000  60.000
-160.000-140.000 180.000  60.000
-160.000-140.000 180.000  60.000
 180.000
$END
```

Explanation:

1. The $CNTRL data group is used to define the type of run (EDMC) and indicate the program to write the coordinates corresponding to each accepted conformation using a PDB format. The names of the PDB files carry the prefix "A10edmc".
2. The $SEQ data group includes the amino acid sequence described by using a single–letter code. The sequence must include the terminal group, in this case, the AMINO–COCH3 and CARBOXYL–NHCH3 at the N– and C–terminus, respectively, are used. The two end groups must be specified also using a one–letter code. The one–letter and three letter codes for the residues and the end groups are provided in the ECEPPAK manual. Since this form of sequence specification (single–letter code) is not the default we have used and extra keyword in the $CNTRL data group: "res_code = one_letter".
3. The $GEOM data group is used to input the set of dihedral angles defining the

conformation of the polypeptide chain.

4. The conformational search protocol is defined through a set of specific keywords. These keywords must be included in the data group $EDMC. Most of the EDMC keywords (see manual) are assigned default values. Few of them have been used here to indicate how to specify certain aspects of the conformational search run.

   a) Length of the run: One possible manner of specifying the length of the Monte Carlo run is to define the maximum number of conformations accepted by the Monte Carlo criterion. This is accomplished by using the keyword MAXIT. In this example, five (5) accepted conformations is specified with MAXIT = 5.

   b) Random numbers: since the EDMC procedure uses random numbers, there is a need to initialize the random number generator by providing an integer (positive or negative). This is accomplished by using the keyword SEED.

   c) Temperature: A parameter associated with the temperature (in Kelvin's degrees) for the simulation is defined using the keyword TEMP.

   d) Whenever the search is trapped in a region of the conformational space, the method attempts to overcome the barriers by generating conformations with major conformational changes and relaxing the criterion of acceptance by increasing the temperature parameter. There are a few alternative procedure to change the temperature. One of them, indicated by the keyword THERMAL_SHOCK, is to produce a sudden jump in the temperature. The high temperature is defined by the keyword T_UP.

   e) Generation of conformations: the EDMC method utilizes different protocols for generating new conformations. These conformations can be generated by random predictions or by using electrostatic predictions. The following keywords are used to control the process of generation: RAND_TO_ELEC defines the ratio of randomly– to electrostatically–generated conformations. In this example a ratio of 3:10 is used (RAND_TO_ELEC = 0.3).

   f) Note: In the present test, the search starts from the initial conformation whose geometry is provided in the $GEOM data group. However, it is possible (and quite common) to override this option by requesting a starting conformation with dihedral angles generated at random. This can easily be specified by using the RAND_START keyword.

9.3.1. Running the EDMC calculation

To run the previous example, the following instruction must be typed in the command line:

```
recepp.s EDMC ten_ala_edmc TEN_ALA_EDMC x x 1
```

The program writes three different type of files:

a) `main_out.TEN_ALA_EDMC` with a description of the results of the conformational search procedure;

b) `outo.TEN_ALA_EDMC` a file containing all the conformations accepted by the Monte Carlo procedure (for each of them, the first line lists the different energy terms, the next line(s) contains the sequence (in ECEPP format) followed by the list of dihedral angles that describe the conformation, and

c) `A10edmc #\#\#.pdb files` (#\#\# represents the number of accepted conformation) containing the Cartesian coordinates of the conformations accepted by the Monte Carlo procedure.

9.4. Availability, hardware platforms and technical documentation

The program is available free for academic users. It can be obtained from the Cornell Theory Center software repository at ***http://www.tc.cornell.edu/reports/NIH/resource/CompBiologyTools/eceppak***. It has also been installed in TASK and is located in the directory ***/chemia/eceppak***; the technical documentation can be found in ***/apl/chemia/eceppak/doc/Manual***. The program runs on the IBM–SP2 supercomputers and IBM workstations, SGI supercomputers and workstations, SUN workstations and E–family servers, and Pentium computers under PC–Linux. The parallel version of the program can be installed on any platforms with the standard Message Passing Interface (MPI) software installed.

10. The ANALYZE program

10.1. Functions of the program

ANALYZE processes the dihedral–angle outo.* files obtained from calculations (usually global conformational analysis with the EDMC method) using ECEPPAK. These functions include the following:

1. Calculations of conformational characteristics, such as hydrogen bonds, turn position and types, RMS deviation from a reference conformation, interchromophore distances, interproton distances, etc.
2. Calculations of Boltzmann–averaged properties of the conformational ensemble.
3. Calculations the dihedral angles from supplied Cartesian coordinates.
4. Cluster analysis of the conformational ensemble by the minimal spanning tree or minimum–variance method.
5. Fitting the statistical weights of the conformations so as to achieve the best agreement between the calculated average and experimental NOE spectra and coupling constants, using the algorithm outlined in section 8 [31].

10.2. Input structure

The input to the program includes control data organized in data groups, as in the case of the ECEPPAK program, the outo.* file containing the dihedral angles of the conformations being analyzed, and, optionally, the NOE intensity data for conformational ensemble–fitting to NMR data [31]. The data groups are as follows:

1. $TITLE the title of the run (a single line);
2. $CNTRL main control variables;
3. $SEQ sequence data;

4. $BRIDGE covalent bridge data;
5. $PROPERTY control data for conformation–dependent property evaluation;
6. $RMSCALC control data for RMS calculation;
7. $BOUNDS reading the NMR–derived distance constraints (interactive at the moment);
8. $CHROMO control data for calculating inter–chromophore distance;
9. $CLUSTER control data for cluster analysis;
10. $SUPAT specifies the atoms being superposed.
11. $NOES specifies the options in the calculations of NOE spectra and coupling constants;
12. $MORASS parameters for theoretical evaluation of the NOE spectra;
13. $COUPLING specification of the calculation of coupling constants;
14. $MARQUARDT options in least–square or maximum–entropy fitting of the theoretical to the experimental NOE spectra.

10.3. Example data for clustering calculation

The example consists of minimal–tree cluster analysis of 170 conformations of oxytocin resulting from a series of EDMC runs. The conformations are contained in file `outo.otv16cl`. The control data contained in file `otvp16cl.inp` are listed below:

```
$TITLE
Oxytocin - 170 conformations from EDMC
$END

$CNTRL
RUNTYP=CLUSTER NRCLUS1=1 NRCLUS2=6
RES_CODE=ECEPP NRES=9 VERBOSE PRINT_PDB=1
$END

$SEQ
1 3 20 8 14 12 3 13 10 6 14
$END

$BRIDGE
2 7
$END

$CLUSTER
6
2.0 1.0 -0.7 0.5 0.2 0.1
5.0
$END
```

```
$SUPAT
5
CA C CB N SG
$END
```

Explanation:

1. $TITLE — this data group contains the title of the run.
2. $CNTRL — control data.
   a) RUNTYP = CLUSTER — this is the clustering run. The minimal spanning tree (default) algorithm will be used.
   b) NRCLUS1 = 1 NRCLUS2 = 6 — atoms of residues from 1 to 6 (inclusive) will be superposed.
   c) NRES = 9 — the sequence contains a total of 9 residues and end groups.
   d) VERBOSE — intermediate input will be printed on screen.
   e) PRINT_PDB = 1 — one PDB file will be produced per family at a chosen cut–off (see below). This means that the Cartesian coordinates of the lowest–energy structure of each family will be output.
   f) RES_CODE = ECEPP — ECEPP numeric code is used to identify amino–acid residues (see Tables 1 and 2).
3. $SEQ — numeric code of the amino–acid sequence, including the blocking groups. The sequence is H–c–[Cys–Tyr–Ile–Gln–Asn–Cys]–Pro–Leu–Gly–NH^2.
4. $BRIDGES — covalent–bridge data. In this specific case it indicates that the half–cystines at position 1 and 6 are linked with a disulfide bond.
5. $CLUSTER — clustering control data. Six cut–off values are used, those being 2, 1, 0.7, 0.5, 0.2, and 0.1 Å. The dihedral angles and PDB files will be produces for the results corresponding to the 0.7 Å cut–off (which is preceded by the "–" sign). The number in the last line is the energy cut–off; if the energy difference between the lowest–energy conformation of a family and the lowest–energy conformation in the set is higher than this value, the family is discarded.
6. $SUPAT — the types of atoms used in superposition. Five atom types are considered, whose name are indicated in the list.

10.4. Availability, hardware platforms and technical documentation

The program is available free for academic users. It can be obtained from the Cornell Theory Center software repository at ***http://www.tc.cornell.edu/reports/NIH/resource/CompBiologyTools/analyze***. It has also been installed in TASK and is located in the directory ***/apl/chemia/analyze***; the technical documentation can be found in ***/apl/chemia/eceppak/doc/***ANALYZE.README. The program runs on the IBM–SP2 supercomputers and IBM workstations, SGI supercomputers and

workstations, SUN workstations and E–family servers, and Pentium computers under PC–Linux.

Acknowledgements

This research was supported by grants from the Polish State Committee for Scientific Research, KBN (3 T09A 111 17) and the National Institutes of Health (R03 TW1064–1). Calculations were carried out at the Informatics Center of the Metropolitan Academic Network (IC MAN) at the Technical University of Gdansk and the Interdisciplinary Center for Mathematical and Computer Modeling (ICM), Warsaw, Poland.

References

[1] M. Vásquez, G. Némethy, H.A. Scheraga, Chem. Rev., **94** (1994), 2183
[2] H.A. Scheraga, Int. J. Quant. Chem., **42** (1992), 1529
[3] H.A. Scheraga, Biophys. Chem., **59** (1996), 329
[4] R. Brüschweiler, M. Blackledge, R.R. Ernst, J. Biomol. NMR, **1** (1991), 3
[5] H. Meirovitch, E. Meirovitch, J. Lee, J. Phys. Chem., **99** (1995), 4847
[6] H. Meirovitch, E. Meirovitch, Biopolymers, **38** (1996), 69
[7] F.A. Momany, R.F. McGuire, A.W. Burgess, , H.A. Scheraga, J. Phys. Chem., **79** (1975), 2361
[8] G. Némethy, M.S. Pottle, H.A. Scheraga, J. Phys. Chem., **87** (1983), 1883
[9] M.J. Sippl, G. Némethy, H.A. Scheraga, J. Phys. Chem., **88** (1984), 6231
[10] G. Némethy, K.D. Gibson, K.A. Palmer, C.N. Yoon, G. Paterlini, A. Zagari, S. Rumsey, H.A. Scheraga, J. Phys. Chem., **96** (1992), 6472
[11] Z. Li, H.A. Scheraga, Proc. Natl. Acad. Sci., U. S. A., **84** (1987), 6611
[12] Z. Li, H.A. Scheraga, J. Molec. Str. (Theochem), **179** (1988), 333
[13] D.R. Ripoll, H.A. Scheraga, Biopolymers, **27** (1988), 1283
[14] D.R. Ripoll, H.A. Scheraga, J. Protein Chem., **8** (1989), 263
[15] D.R. Ripoll, L. Piela, M. Vásquez, H.A. Scheraga, Proteins Struct. Funct. Genet., **10** (1991), 188
[16] D.R. Ripoll, M.J. Vásquez, H.A. Scheraga, Biopolymers, **31** (1991), 319
[17] D.R. Ripoll, Int. J. Pepide Protein Res., **40** (1992), 575
[18] D.R. Ripoll, A. Liwo, H.A. Scheraga, Biopolymers, **46** (1998), 117
[19] J. Vila, R.L. Williams, M. Vásquez, H.A. Scheraga, Proteins Struct. Funct. Genet., **10** (1991), 199
[20] J.D. Augspurger, H.A. Scheraga, J. Comput. Chem., **17** (1996), 1549
[21] J.D. Augspurger, H.A. Scheraga, J. Comput. Chem., **18** (1997), 1072
[22] A. Liwo, M.R. Pincus, R.J. Wawak, S. Rackovsky, H.A. Scheraga, Protein Sci., **2** (1993), 1715
[23] A. Liwo, A. Tempczyk, S. Oldziej, M.D. Shenderovich, V.J. Hruby, S. Talluri, J. Ciarkowski, F. Kasprzykowski, L. Lankiewicz, Z. Grzonka, Biopolymers, **38** (1996), 157
[24] M. Vásquez, H.A. Scheraga, J. Biomol. Struct. & Dyn., **5** (1988), 705
[25] M. Vásquez, H.A. Scheraga, J. Biomol. Struct. & Dyn., **5** (1988), 757
[26] H. Späath (1980), *Cluster Analysis Algorithms*, Halsted Press; New York
[27] G. Wagner (1990), *NMR investigation of protein structure*, In Progress in NMR

spectroscopy, **23**, 101

[28] K. Wüthrich (1986), *NMR of Proteins and Nucleic Acids*, Wiley, New York

[29] J.–X.Yang, T.F. Havel, J. Biomol. NMR, **3** (1993), 355

[30] M.J. Blackledge, R. Brüschweiler, C. Greisinger, J.M. Schmidt, P. Xu, R.R. Ernst, Biochemistry, **32** (1993), 10960

[31] M. Groth, J. Malicka, C. Czaplewski, S. Ołdziej, L. Łankiewicz, W. Wiczk, A. Liwo, J. Biomol. NMR (1999), submitted

[32] C.B. Post, R.P. Meadows, D.G. Gorenstein, J. Am. Chem. Soc., **112** (1990), 6796

[33] R.P. Meadows, C.B. Post, B.A. B. A. Luxon, D. G.Gorenstein (1994), *MORASS 2.1*, Purdue University, W. Lafayette

[34] G.J. Daniell (1991), *Of maps and monkeys: an introduction to the maximum entropy method, Maximum entropy in action*, Ed. B. Brian and V. A. Macaulay, 1, Clardenon Press, Oxford

[35] D.M. Gay, ACM Trans. Math. Software, **9** (1983), 503

[36] D.W. Marquardt, J. Soc. Indust. Appl. Math., **11** (1963), 431