UNIVERSITÀ DI PISA

DIPARTIMENTO DI INFORMATICA

TECHNICAL REPORT: TR-04-16

Predicting Thermodynamic Properties from Molecular Structures by Recursive Neural Networks. Comparison with Classical Group Contributions Methods.

Luca Bernazzani

Celia Duce

Vincenzo Mollica

Maria R. Tiné

Alessio Micheli

Alessandro Sperduti

Antonina Starita

October 12, 2004

ADDRESS: via F. Buonarroti 2, 56127 Pisa, Italy. TEL: +39 050 2212700, FAX: +39 050 2212726

Predicting Thermodynamic Properties from Molecular Structures by Recursive Neural Networks. Comparison with Classical Group Contributions Methods.

Luca Bernazzani[a], Celia Duce[a], Alessio Micheli[b*], Vincenzo Mollica[a], Alessandro Sperduti[c], Antonina Starita[b], and Maria R. Tiné[a]

[a]*Dipartimento di Chimica e Chimica Industriale, Università di Pisa, Via Risorgimento 35 – I-56126 Pisa, Italy.*

[b]*Dipartimento di Informatica, Università di Pisa, Via F. Buonarroti 2 – I-56127 Pisa, Italy.*

[c]*Dipartimento di Matematica Pura ed Applicata, Università di Padova, Via Belzoni 7 – I-35131 Padova, Italy.*

Abstract

The recursive neural networks deal with prediction tasks for compounds represented in a structured domain. These approaches allow combining, in a learning system, the flexibility and general advantages of a neural network model with the representational power of a structured domain. As a result a completely new approach to QSPR/QSAR analysis is obtained through the adaptive processing of molecular graphs whose performance is even better than that of traditional approaches. In this paper a Recursive Cascade Correlation model (RecCC) has been applied to the analysis of the Gibbs free energies of solvation in water of 179 monofunctional open chain organic compounds. An original representation of the molecules in terms of labeled directed positional acyclic graphs has been developed by individuating a limited set of constituent atomic groups and representation rules. The descriptive and predictive abilities of the RecCC model have been tested and compared with those of a traditional group contributions method. The inherent ability of the RecCC model to abstract chemical knowledge through the adaptive learning process have been investigated by Principal Components Analysis of the internal representations developed by the neural network, finding that the model recognizes the chemical compounds on the basis of a not trivial combination of their chemical structure and target property.

* Corresponding author. E-mail: micheli@di.unipi.it

INTRODUCTION

The application of QSAR (Quantitative Structure-Activity Relationship) and QSPR (Quantitative Structure-Property Relationship) to the prediction of biomedical and pharmacological properties of molecules of new design is becoming increasingly important also due to the development of artificial intelligence techniques.[1,2]

The prediction of the biological activity of a drug requires the knowledge of the interaction mechanism between the molecule and the specific site of action of the receptor but is determined also by the physicochemical parameters regulating the absorption into the cells.[3] The transport of a molecule from the aqueous medium of the extracellular region to the cytoplasm is ruled by passive diffusion across the lipid bilayer of the membrane and is closely connected to its lipophilicity. For a given solute this property is usually expressed by its partition coefficient between water and a suitable apolar organic solvent.[4] In this respect, experimental evaluation and rationalisation of thermodynamic solvation properties in two not miscible solvents are very important. Indeed, these quantities can be used to calculate the corresponding partition properties and to define a lipophilicity scale.[5]

Since a long time a thermodynamic investigation of the solvation of organic molecules in different solvents has been carried out by many authors.[6-8] The collected information were rationalised by using group or bond contribution methods[6,9,10] and/or empirical correlation based on solute molecular descriptors such as connectivity indexes,[11] hydrogen bond indexes,[12] solvatochromic parameters,[13] etc. All these methods can be defined as QSP-Relationships and, as they use *a priori* analytical correlations, can be classified as equation based regression approaches. The choice of a suitable group of descriptors for the modelling of examined systems represents a fundamental issue of these correlation procedures. Basically, the above methods use as input parameters (i) indexes whose values can be directly inferred by the molecular structure; (ii) properties experimentally determined (boiling point, chromatographic retention time, molar volume, etc.); (iii) descriptors as polarity, polarizability, molar refractivity, hydrogen bonding capability etc. which can be calculated through simple computational methods.

In recent years, various machine learning techniques, such as artificial neural networks (NN) and genetic algorithms were applied for the formulation of QSPR/QSAR relationships as alternative to equation based regression approaches. In the NN-QSPR approach no *a priori* correlation is instituted between the molecular descriptors and the physicochemical property. Moreover, the NNs have the inherent ability to incorporate nonlinear relationships between chemical structural parameters and the target property, which represents the main advantage of their use over the classical methods. Concerning our field of interest, English and Carroll[1] reported recently NN-QSPR models for estimating Henry's Law constants of organic compounds based on various feedforward multi-layer perceptron (MLP) neural network architecture trained by back-propagation learning algorithm, whereas Yaffe *et al.*[2] proposed other models based on the cognitive classifier fuzzy ARTMAP for the estimation of boiling temperature, critical properties, aqueous solubility, octanol-water partition coefficient, and Henry's Law constant. Highly specific software packages are employed to generate the parameters for describing the molecules and to optimize their number according to the set of examined compounds. Both the approaches represent the molecules by a fixed-size numerical array, thus using a predetermined number of molecular descriptors. This is a compulsory choice since usually standard methods including also MLP neural networks are not suited to deal with structured domains.

The recursive neural networks (RecNNs) overcome this limit allowing to deal directly with a structured representation of the compounds. In fact, RecNNs belong to the area of Machine Learning models developed to directly handle structured data.

The neural network paradigm was first adapted to deal with (temporal) sequences: specifically, *recurrent neural networks* generalize MLP feedforward networks to sequence. These models provide the neural networks with dynamic properties by the use of contextual internal states (*memory*), and a recursive encoding process of the input signal. Successively, the computational capabilities of neural networks have been extended to the processing of structured domains (*recursive neural networks*)[14,15] generalizing the recursive-encoding concept to hierarchical data.

The model presented in this paper is based on RecNN models, to deal with chemical domains. The possibility of processing structured information using neural networks is appealing for several reasons, especially in the context of prediction tasks in Chemistry, where data are usually gathered experimentally and the compounds can naturally be represented as labeled graphs. In fact, neural networks are universal approximators and they are able to learn from a set of examples, dealing with noisy and incomplete, or even ambiguous, data.

The exploitation of the RecNN method allow us to obtain a completely new approach to QSPR/QSAR analysis through the adaptive processing of molecular graphs, as previously shown by the successfully application of this procedure compared to traditional approaches.[16-18]

In particular, a structured representation of chemical compounds is able to convey much more information than standard vectorial representation, and it is enough general and flexible to deal with either different kind of compounds or problems of different nature using the same approach.

The recursive model can encode the structured representation of the molecules according to the given QSPR/QSAR task and therefore it models a direct and adaptive relationship between molecular structures and target properties. In fact, RecNNs take directly as input the molecular graph of the compounds and they discover by learning the specific structural descriptors for the task at hand. As a result, no a priori definition and/or selection of properties by an expert is needed in the new method.

In the frame of a research project concerning the application of recursive neural networks to the analysis of a wide number of thermodynamic properties in a QSPR/QSAR approach, we report here the use of RecNNs to describe the standard free energy of solvation, $\Delta_{solv}G°$, in water of a set of linear monofunctional organic compounds. Solvation properties were selected as target because of both the availability of a large number of literature data (which allows an effective training of the neural network) and the modelistic implications of these properties.

We shall present and discuss the rules used to represent the examined molecules as objects in a structured domain in the form of labelled directed ordered trees and the results of the RecNN procedure.

This RecNN approach has been successfully applied to the prediction of the bioactivity of a class of benzodiazepines.[17] However, in this paper, where the aim is to address a chemical task in a very general way, the representation rules have been thoroughly revised in order to deal with the widest set of molecular structures.

The results obtained with the RecNN procedure will be compared with those obtained by applying a standard group contributions method (GCM) to the same set of molecules. This comparison is meaningful since both approaches deal with a representation of the molecules directly inferred by the molecular structure alone. Moreover the GCM method has proved one of the most reliable in the prediction of

thermodynamic properties of molecules in solution at infinite dilution and can be considered in some sense as the reference method in the solvation field.

THEORETICAL BASIS

1. A general innovative approach to QSPR/QSAR studies

As it is well known, the basic idea of a QSPR/QSAR study is to find an appropriate function F that predicts any molecular property, or the biological activity, using information related to molecular structure only:

$$Property = F\left(Structure\right) \tag{1}$$

The input domain of F is a set of molecular structures, where the term "structure" refers to global information characterizing the molecule (molecular shape, chemical functionalities, etc.), and the output domain is typically the set of real numbers, which are used to quantify the property of interest. Hence, the function F can be seen as a functional transduction defined on a structured domain.

As already outlined,[17,18] the function F can be described as the sequential solution to two main problems: (i) the representation problem, i.e., how to encode molecules through the extraction and selection of structural features and (ii) the mapping problem, usually faced by linear or non-linear regression tools.

According to this view, F can be decomposed as follows

$$F = g\left(f\left(\ \ \right)\right) \tag{2}$$

where f is the *encoding* function or *feature representation* function from the domain of the chemical structures to the descriptors space, while g is the mapping function from the descriptors space to the physicochemical property and/or bioactivity space. This scheme is summarized in Figure 1 where chemical features are represented by a suitable set of numerical descriptors obtained through the encoding procedure and then used to predict the target property.

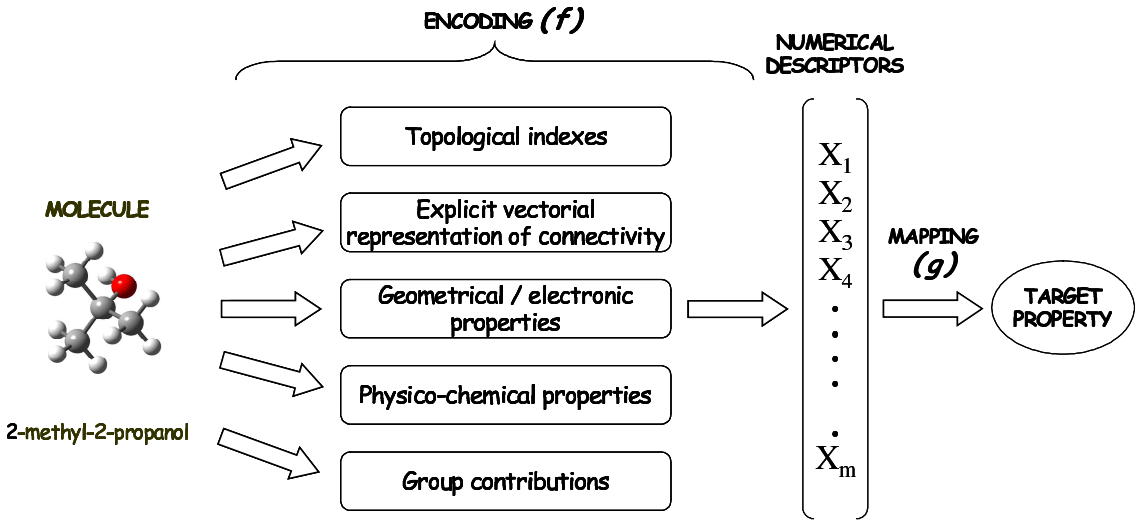

Figure 1. Outline of the traditional QSPR/QSAR approach. Structural features of the molecule are represented by a proper set of numerical descriptors obtained by using different encoding approaches (*f* function). The target property is then obtained by applying a regression procedure (*g* function).

The molecule can be represented by using different encoding approaches such as the selection of physicochemical, geometrical and electronic properties, the calculation of topological or connectivity indices, as well as the occurrence of each group in the molecular structure. Obviously, the number and type of descriptors used to represent the chemical compound depend on the specific QSPR/QSAR problem at hand and are decided by an expert in the field. In more detail, the encoding process requires the solution of two subtasks aimed the first one to explicitly represent the significant structural information carried by molecules and the second one to codify this structural information into a numerical representation. Thus, the function $f(\ )$ can be understood as the following composition

$$f = f_E\left(f_R(\ )\right) \tag{3}$$

where $f_R(\ )$ extracts a specific structural aspect from the molecule, and $f_E(\ )$ computes a numerical value from the structure returned by $f_R(\ )$. However, in traditional QSPR/QSAR approach, both $f_R(\ )$and $f_E(\ )$ are defined *a priori*, i.e., they do not depend on the regression procedure. Therefore they are often designed through a very expensive trial and error procedure on the basis of prior knowledge of the problem in order to adapt them to the specific QSPR/QSAR task. So the problem of using the molecular graph (whose power in conveying chemical structural information is well recognized) in a form directly amenable to QSPR/QSAR analysis is still open though representing one of most exciting goal in this field.

In this paper we exploit recursive neural networks[14] to infer the $f_E(\ )$ function by learning on the structured data. The adaptive nature of $f_E(\ )$ allows the automatic generation of numerical descriptors that are specific for the regression task to be solved. This is possible because the RecNN is able to take directly as input the graph generated by $f_R(\ )$ and to implement adaptively both $f_E(\ )$ and g. In order to exemplify the above concepts, in Figure 2, we show the outline of the proposed approach assuming that a given molecule is represented by $f_R(\ )$ as a labelled tree. This tree-structured representation is then processed by a RecNN. The output of the RecNN constitutes the regression output, while the internal representations of the recursive neural network (i.e., the output of the hidden units) constitute the neural implementation of the numerical descriptors returned by $f_E(\ )$. It must be stressed, at this point, that the RecNN does not need to take as input a fixed-size numerical vector for each input graph, as it happens with standard neural networks typically used in QSPR/QSAR studies, because it is able to treat variable-size representations of the input graph. Moreover, since the encoding function ($f_E(\ )$) is learned by the neural network together with the mapping function (g), the resulting numerical code represents the "best" numerical coding of the input graph for the given QSPR/QSAR task. We may observe that the main difference between the traditional QSPR/QSAR scheme shown in Figure 1 and the proposed new scheme reported in Figure 2 is due to the automatic definition of the $f_E(\ )$ function obtained by training the RecNN over the regression task. This implies that no *a priori* selection and/or extraction of features or properties by an expert is needed in the new scheme for $f_E(\ )$. A detailed description of how the encoding function $f_E(\ )$ is computed for each input graph is reported in section 3 and in Ref. 18.

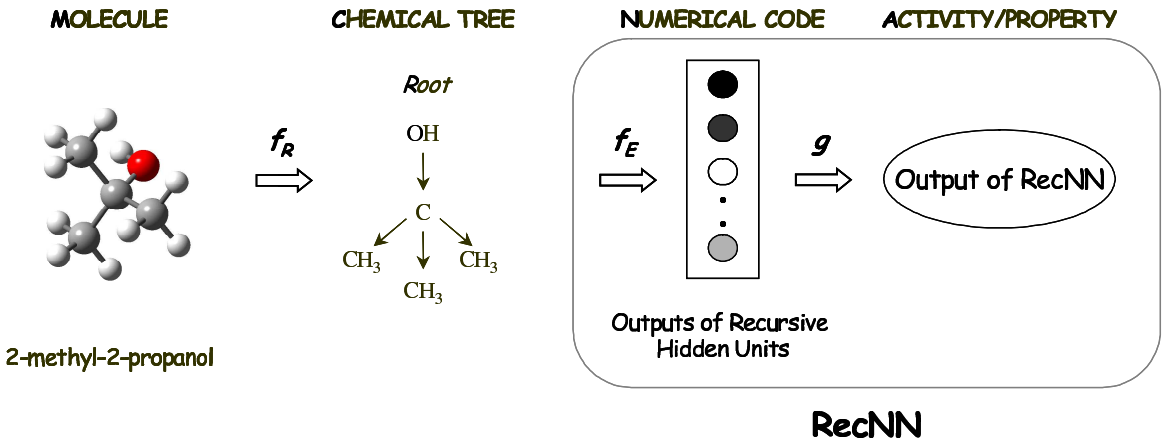

Figure 2. Outline of the RecNN approach to QSPR/QSAR. RecNN takes directly as input the molecular graph generated by $f_R(\ )$ encoding it through the adaptive function $f_E(\ )$ (recursive hidden units). The function *g*, implemented by the output part of RecNN, uses this internal representation to produce the final prediction value (target property).

On the basis of the above framework, we review here the classical group contributions method and the recursive neural network with comparison aims. Specifically, we can prove that RecNNs can be a suitable tool for the automatization of two fundamental parts of the QSPR/QSAR analysis, *i.e.* the encoding and mapping phases. Moreover, reviewing the different approaches in a general perspective should help to analyze the advantages of the new approach.

2. *Group contributions methods.*

Group contributions schemes have been extensively employed as a practical method for predicting a plethora of thermodynamic and related properties such as free energies and enthalpies of formation,[10,19] enthalpies of vaporisation,[20] critical properties and boiling points,[21] vapour pressure of pure liquids,[22] and so on. The basic idea underlying these approaches is that a solute molecule acts as a number of fragments contributing to the investigated property as independent of each other. These fragments can be constituted by individual atoms or simple atomic groups and usually coincide with portions of alkyl chains or with the functional groups characterising the different classes of organic compounds.

In particular, several schemes of group additivity have been developed in order to rationalise and predict thermodynamic functions of solvation and partial molar properties at infinite dilution of a wide variety of compounds in solvents of different features.[6,23] Though these procedures are essentially empirical, at least as regards free energies of solvation, some molecular basis can be found in the Ben-Naim's theory.

According to Ben-Naim[24], the solvation process can be defined as the transfer of a solute molecule from a fixed position in the ideal gas phase to a fixed position in a given solvent. Therefore, the free energy of solvation can be calculated as the difference between the pseudochemical potentials of the molecule S, μ_S^*, in the two phases, the latter quantity representing the chemical potential of the molecule with no translational degrees of freedom. Since the interaction potential is null in the ideal gas state, the free energy of solvation reduces to:

$$\begin{aligned}\Delta_{\text{solv}}G^{*} &= \mu_{\text{S}}^{*}(liq) - \mu_{\text{S}}^{*}(gas) \\ &= -kT\ln\left[\left\langle \exp(-\beta B_0)\right\rangle\right]_{(liq)}\end{aligned} \tag{3}$$

where B_0 is the binding energy of the newly added particle at fixed position with all the other particles of the system in a given specific configuration and $\langle\ \rangle$ indicate the average over all the instantaneous configurations.

As remarked by Ben-Naim, the quantity $\Delta_{\text{solv}}\text{G}^{*}$ corresponds in magnitude to the change of the standard free energy of solvation when the 1M concentration of the solute in both gas and liquid phase is assumed as the standard state.

On the basis of this model we can attempt to break down the free energy of solvation in a sum of contributions each related to the solvation of an individual fragment of the solute molecule. Let us consider a solute molecule of the form A-B where A and B are the two groups constituting the molecule. We can assume that the binding energy of the solute molecule, $B_{\text{A-B}}$, can be split as follows:

$$B_{\text{A-B}} = B_{\text{A}} + B_{\text{B}} \tag{4}$$

the terms B_{A} and B_{B} being the binding energies of the groups A and B with the solvent molecules in a particular configuration, respectively. Under this hypothesis the free energy of solvation has the form

$$\Delta_{\text{solv}}G^{*}_{\text{A-B}} = -kT\ln\left\langle \exp(-\beta B_{\text{A}})\exp(-\beta B_{\text{B}})\right\rangle \tag{5}$$

The average product of two functions may not in general be factorized into a product of two average quantities unless the two functions are independent of each other. In the present case, this should mean that the molecular fragments A and B interact with the liquid phase in an independent manner, thus allowing to write

$$\begin{aligned}\Delta_{\text{solv}}G^{*}_{\text{A-B}} &= -kT\ln\left[\left\langle \exp(-\beta B_{\text{A}})\right\rangle\left\langle \exp(-\beta B_{\text{B}})\right\rangle\right] \\ &= \Delta_{\text{solv}}G^{*}_{\text{A}} + \Delta_{\text{solv}}G^{*}_{\text{B}}\end{aligned} \tag{6}$$

This relation should furnish the molecular basis of the group additivity concept when applied to the solvation free energies by merely assuming that the interaction potential can be split according to Eq. (4). This situation is certainly verified when the groups A and B are located at infinite distance from each other and, in terms of interactions, this also means null correlation between the groups inside the molecule and with respect to the solvent molecules.

In any case a form of additivity can always be deduced by introducing the concept of conditional solvation, thus obtaining from Eq.(5):

$$\Delta_{\text{solv}}G^{*}_{\text{A-B}} = \Delta_{\text{solv}}G^{*}_{\text{A}} + \Delta_{\text{solv}}G^{*}_{\text{B/A}} \tag{7}$$

The quantity $\Delta_{\text{solv}}G^{*}_{B/A}$ represents the conditional Gibbs energy of solvation of the group B at a point near the group A already interacting with the solvent. The latter quantity differs in general from $\Delta_{\text{solv}}G^{*}_{B}$ because the group B is brought to a location very near to A. Only in very particular cases it is possible to assume that the Gibbs energy is strictly additive as in Eq. (6), a generalized form as in Eq. (7) can always be achieved.

The above procedure can obviously be extended to large molecules with many groups.

The general form of equation commonly employed in these additivity scheme is:

$$Y = \sum_{\text{j}} n_{\text{j}}B_{\text{j}} \tag{8}$$

where Y is the thermodynamic function of interest and B_j is the contribution to the property of the jth group present n_j times in the solute structure.

The usual procedure employed for determining the values of the group contributions is based on a linear multiparameter regression in which the experimental values of the solvation property are fitted to Eq. (8) by introducing the frequencies of each group in the molecular structures as independent variables. The group contributions Bj are then obtained as the regression parameters.

Eq. (8) can suitably be written as

$$Y = \mathbf{N} \cdot \mathbf{B} \tag{9}$$

where **N** is the row vector of the group frequencies and **B** the column vector of the group contributions.

In the framework of the QSPR/QSAR approach it is straightforward to recognize that the function $f_R(\ )$ consists in extracting from the molecular formula the fragments in which the molecule should be divided and the frequency of their occurrence, while $f_E(\ )$ is the construction of the row vector of the frequencies, by putting each frequency in the correct position univocally identifying each group. The mapping function $g(\ )$ is simply the product of the row vector of the frequencies by the column vector of the group contributions. It may be observed, however, that both **N** and **B** are fixed dimension arrays, their length be defined once the whole set of groups constituting the molecules under consideration is chosen.

3. The Recursive Neural Network Model

A neural network is a powerful data modelling tool that is able to approximate complex input/output relationships. For standard neural networks the input pattern is a fixed-size vector. RecNN extends standard NN to deal with structured domains, i.e. input patterns can be a class of graphs, trees, sequences, etc. Hence, RecNN allows to learn directly a mapping between molecular graphs and their target properties.

In this section we present the approach based on recursive neural networks for the processing of structured domains. We first provide a proper instantiation of the input and output domains of the functions $f_E(\ )$ and $g(\ )$ implemented by the RecNN.

Let the structured input domain for $f_E(\ )$, denoted by G, be a set of labeled directed positional acyclic graphs (DPAGs). In a DPAG, for each vertex (or *node*) a total order on the edges leaving from it is defined and a position is assigned to each edge.

Moreover let us assume that G has for each node a bounded out-degree and that each DPAG possesses a super-source, i.e. a vertex s such that every vertex in the graph can be reached by a directed path starting from s. *Labels* are tuples of variables and are attached to vertexes. Let $\Re^n$ denote the label space.

Here, we consider a subclass of DPAGs formed by prohibiting cycles in the undirected skeleton of the graph, the set of the k-ary trees. In the case of trees the super-source is defined by its *root node*. *k-ary trees* (*trees* in the following) are rooted positional trees with finite out-degree k. This class of structures includes the class of rooted ordered trees, and, clearly sequential and vectorial data.

Given a node v in the tree $T \in G$, we give the following definitions

- The children of v are the node successors of v, each with a position $j=1,...k$;
- k is the maximum out-degree over G, i.e. the maximum number of children for each node;

— $L(v)$ in $\Re^n$ is the input label associated with v, and $L_i(v)$ is the i-th element of the label;
— The *subtree* $T^{(j)}$ is a tree rooted at the j-th children of v.

Vertexes with zero out-degree are *leaves* of the tree. The set of external vertexes is the *frontier*. *Traversal* of a tree allows to systematically visit (and process) all the nodes of the tree in some order: in particular, processing in a recursive manner all subtrees and finally the root we define a *postorder* traversal. The scheme of a *k-ary tree* is reported in Figure 3.

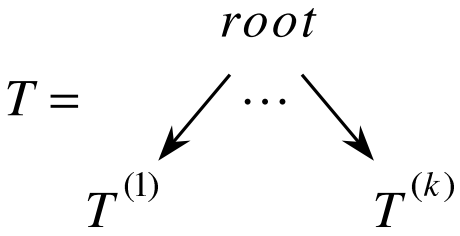

Figure 3. *k-ary tree T* rooted in the node *root,* with subtrees $T^{(1)}$, ... , $T^{(k)}$.

The descriptor (or code) space is chosen as $\Re^m$ while the output space, for our purpose, is defined as $\Re$. Finally, the class of functions which can be realized by a RecNN can be characterized as the class of functional graph transductions described in the form $g(f_E(\ ))$, where $f_E(\ ): G \to \Re^m$ is the encoding function, and g: $\Re^m \to \Re$ the output function.

The functions and domains involved in the definition of the RecNN are shown in Eq. (10).

$$Structure \xrightarrow{f_R} G \xrightarrow{f_E} \Re^m \xrightarrow{g} \Re \tag{10}$$

In our approach we define a function $f_E(\ )$ that allows to progressively encode an input structure, e.g. a tree, using at each step a neural computational model τ_{NN}.

The function τ_{NN} is used to process each node of a given structure.

Given a node in a tree T, τ_{NN} uses the information available at the current node:

1. the numerical label attached to the node (in $\Re^n$),
2. the numerical code for each subgraph of the node (in $\Re^m$),

and produces a code in $\Re^m$.

As a result, if k is the maximum out-degree of T in G, τ_{NN} is defined as

$$\tau_{NN} : \Re^n \times \underbrace{\Re^m \times ... \times \Re^m}_{k} \to \Re^m \tag{11}$$

The information coded in the k spaces $\Re^m$ are the new parts introduced in the RecNN model. We show in the following as these extra inputs to the model allow to memorize structural information and make the model responsive to the topological structure of the input compound.

First of all, let us consider, for example, the simplest non-linear realization for τ_{NN} that uses a single recursive neural unit ($m=1$). Given a node v of T, the output X in $\Re$ of the recursive neuron (i.e., the code of v), is computed as follows:

$$X = \tau_{NN}(L(v), X^{(1)}, ..., X^{(k)}) = \\ = \phi\left(\sum_{i=1}^{n} w_i L_i(v) + \sum_{j=1}^{k} \hat{w}^j X^{(j)} + \theta\right) = \phi(\mathbf{W}L + \sum_{j=1}^{k} \hat{w}^j X^{(j)} + \theta) \quad (12)$$

where ϕ is a non-linear sigmoidal function, $L(v)$ in $\Re^n$ the label of v, θ is the bias term, $\mathbf{W}$ in $\Re^n$ is the weight vector associated with the label space, $X^{(j)}$ in $\Re$ is the code for the j-th subgraphs (subtree) of the v, and $\hat{w}^j$ in $\Re$ is the weight associated with the j-th subgraph space.

When m>1 we obtain a neural network (with m units):

$$\mathbf{x} = \tau_{NN}(L(v), \mathbf{x}^{(1)}, ..., \mathbf{x}^{(k)}) = \Phi(\mathbf{W}L + \sum_{j=1}^{k} \hat{\mathbf{W}}^j \mathbf{x}^{(j)} + \theta) \quad (13)$$

Where $\mathbf{x}$ is a vector in $\Re^m$, Φ is a set of m sigmoidal functions, θ in $\Re^m$ is the bias vector, $\mathbf{W}$ in $\Re^{m \times n}$ is the weight matrix associated with the label space, $\mathbf{x}^{(j)}$ in $\Re^m$ are the vectorial codes for each subgraph (subtree) of the current node and $\hat{\mathbf{W}}^j$ in $\Re^{m \times n}$ is the weight matrix associated with the j-th subgraph space.

The composition of τ_{NN} used to encode a structured set of nodes, e.g. a tree T as shown in Figure 3, is defined by the following *recursive* definition of $f_E(\ )$ (Eq. 14):

$$f_E(T) = \begin{cases} \mathbf{0} & \text{if } T \text{ is empty} \\ \tau_{NN}\left(L(root), f_E(T^{(1)}), ..., f_E(T^{(k)})\right) \end{cases} \quad (14)$$

where $\mathbf{0}$ is the null vector in $\Re^m$, *root* is the root node (or super-source of the tree T), $L(root)$ is the label attached to the root, and $T^{(1)}, ..., T^{(k)}$ are the subtrees pointed by *root*.

Note that the same definition may be applied to DPAGs once the super-source s corresponds to the root of the tree.

Eq. (14) comprehensively defines the functionality of the recursive neural network. The recursive definition of $f_E(\ )$ determines a systematic visit of the input tree. It guides the application of τ_{NN} to each node of the structures, from the frontier to the root of the input tree, allowing the neural model to incrementally compute a numerical code for the whole structure. The process corresponds to a postorder traversal of the input tree T, that entails the recursive processing of all the subtrees and finally of the root of T.

Let us consider an example that allows us to grasp the recursive encoding of structures. In Figure 4 we describe the encoding process visualizing through boxes the progressive processing of the input tree performed by the RecNN. This corresponds to "unfold" Eq. (14) through the current input structure (the molecular graph for the compounds 2-methyl-2-propanol and methyl propyl ether in the example).

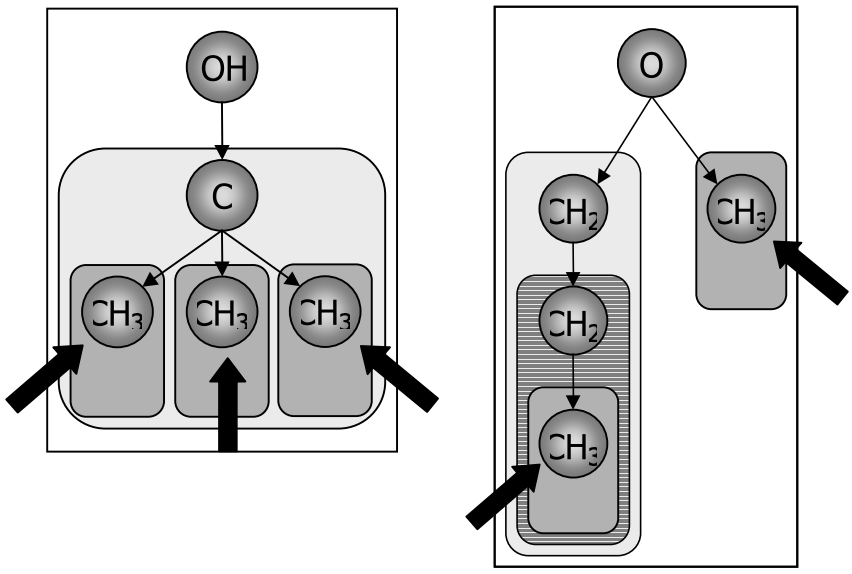

Figure 4. Unfolding the encoding process through structures. Each box includes the subtree progressively encoded by the recursive neural network.

Figure 4 allow us to show the dynamic of the encoding process:

1. The encoding is a bottom-up step-by-step process following the inverse of a topological order (visit of the input graph). The beginning of the process is shown by arrows in Figure 4.
2. For each node we apply a neural model τ_{NN} with auxiliary inputs, i.e. the code of the substructures.

The encoding process terminates at the root of the tree and the numerical code computed for the root is used as numerical code for the whole molecular structure.

As a result we can observe that:

— the encoding process is modeled to the morphology of each input compound;

— the encoding process is adaptive via the free parameters of τ_{NN}.

With respect to the neural realization of τ_{NN} (Eqs. 12 and 13) we can observe that the connection $\hat{\mathbf{W}}^j$ and the internal state variables $\mathbf{x}$ are the machinery able to store information from the structure of inputs and to use it together with the current input (the label of the node). The internal state of the RecNN encodes a representation sensible to the context. Moreover, due to the learning capabilities, the memory of a recursive neural network is a dynamic memory that is dependent from the task. The state-units discover adaptive abstract representations of structural data containing the information relevant to the prediction.

In order to produce the final prediction value, the RecNN model is completed by the output function g which can be realized by choosing any known mathematical model. In the class of neurocomputing models the function g may be obtained using a multilayer network to perform regression or classification tasks.

Here we use a single linear output neuron to realize a regression model:

$$y = g(\mathbf{x}) = \mathbf{A}^t\mathbf{x} + \theta \tag{15}$$

where $\mathbf{A} \in \Re^m$ and θ is the output threshold.

4. Training algorithm and RecCC.

Since, as for standard feedforward neural networks, the error function is differentiable and the weights of the model define a continuously parameterized space of functions, the learning algorithm of the RecNN can still be based on a gradient descent technique.

However, the learning algorithm must be adapted to face the contextual nature of the approximation task, in this case a map from set of trees to output values. It must account for the set of encode transitions developed by the model for each step of the inputs, i.e. the computing a gradient must take into consideration not just the current input node, but all input nodes seen in the encoding process by the RecNN.

Standard supervised algorithms for feed-forward neural networks have been extended to deal with structures, i.e. for RecNN.[14] In the learning algorithm of recursive neural networks, encoding of structures and the iterative weights updates are interleaved. For RecNNs of the type described so far we can outline a general learning algorithm as in Scheme 1.

Scheme 1. Outline of the learning algorithm for a training set constituted by couples (*T*, *d*), where *T* is a tree and *d* the corresponding target property value.

Repeat

For each example of the training set (*T*,*d*)

Encode the input tree *T*: compute f_E through *T* applying τ_{NN} for each node of *T* following a post-order traversal of the tree

Compute the output value for *T* by *g*, i.e. $g\left(f_E(T)\right)$

Compute the error evaluating the differences between the target value *d* and $g\left(f_E(T)\right)$

Compute the gradient of error for each weight of the model accounting for the influences of the weight for each step of the encoding process: the error is back-propagated unfolding the encoding process through the structure of *T*

Update the weights **W** of *g* and τ_{NN} according to the value of the gradient computed over the structures of the training set

Until convergence.

Indeed, there are different ways to realize the recursive neural network.[14] In the present work we choose to use a constructive approach that allows the training algorithm to progressively add the hidden recursive neurons during the training phase. The model is an (recursive) extension of Cascade Correlation based algorithms,[25,26] the Recursive Cascade Correlation (RecCC). The built neural network has a hidden layer composed of recursive (hidden) units. The recursive hidden units compute the values of f_E (in $\Re^m$) for each input tree. The number of hidden units, i.e. the dimension *m* of the descriptor space, is automatically computed by the training algorithm, thus allowing an

adaptive computation of the number and type of (numerical) descriptors needed for a specific QSPR/QSAR task.

The learning algorithm of the RecCC is still based on gradient computation. However the sequence of steps of the general learning algorithm presented above is changed to consider one hidden unit at time of the growing network: learning is performed as in standard Cascade Correlation by interleaving the minimization of the total error function (LMS) by a simple backpropagation training of the output layer (g function), and the maximization of the (non-normalized) correlation, i.e. the covariance, of the new inserted hidden unit with the residual output error (where the hidden units realize f_E).

When, as in the present application, a rather low number of training data is available, special care have to be paid to avoid overfitting. Several expedients can be used at this purpose. First of all, a reduced RecCC architecture is built since no connection between hidden units are allowed. Then the gain of the sigmoids of the hidden units is set to 0.4. Specifically, an incremental strategy (*i-strategy*)[16] on the number of training epochs can be adopted for each new inserted hidden unit of the RecCC model. Allowing few epochs to the first units we avoid the increase of the weight values. The advantages of this strategy are already shown in Ref. 17. The work of Bartlett[27] gives theoretical support for techniques, like the *i-strategy*, that allows to produce networks with small weights. As a result we can continue the learning, adding new hidden unit in RecCC, without overtraining the model. Anyway in the present work we also performed experiments aimed at studying the behaviour of the model with different fitting conditions. The results demonstrate that actually an early stopping of the training convergence does not improve the general performance.

A complete description of the RecCC algorithm and a formulation of the learning method and equations can be found in Ref. 16.

Summarizing, using the recursive neural networks models we realize the f_E function through an adaptive mapping. The process can consider both the graph topology (connectivity) and the atom types (or the chemical functionalities). Since the encoding function (f_E) is learnt by the neural network together with the mapping function (g), we allow the automatic generation of numerical descriptors which are specific for the regression (QSPR/QSAR) task to be solved. Differently from the previous approaches, this implies that no *a priori* selection and/or extraction of features or properties by an expert is needed in the new scheme for f_E.

5. *Representational Issues.*

As mentioned above, in our approach the representation of the molecule is directly inferred from the molecular structure alone. For this purpose we chose to describe the molecule by means of a 2-D graph easily obtained from the structure formula. The overall procedure in general leads to a loss of information with respect to a 3-D representation. However, part of the lost information can be recovered by introducing an order among the atomic clusters individuated as the "groups" constituting the molecule, thus matching, in some sense, the well known Newmann projections employed to assign the absolute configuration of a chiral molecule.

We have represented the molecular structures in terms of labelled rooted ordered trees. Since the investigated dataset included alkanes, alkenes, alkynes, alcohols, ethers, thiols, thioethers, aldehydes, ketones, carboxylic acids, esters, amines, amides, haloalkanes, nitriles and nitroalkanes, we decided a set of rules allowing to obtain an

unique structured representation of each molecule. The adopted rules can be summarised as follows:

— Each molecule is divided in the following atomic groups: CH_3, CH_2, C, H, C=C, C≡C, OH, O, C=O, NH_2, NH, N, SH, S, CN, NO_2, F, Cl, Br, I.
— Each group corresponds to a node of the tree and each bond between them corresponds to an edge.
— We decided an 'ad hoc' priority scale for chemical groups and the root of the tree is placed on the group with the highest priority.
— The orientation of the edges follows the increasing levels of the trees.
— The total order on the subtrees of each node is defined according to the same priority scale for chemical groups used to select the root.

The whole set of rules is reported in Appendix A.

It must be stressed that the above defined atomic groups coincide, but not completely, with the functional groups identifying the different classes of organic compounds. In particular the groups COOH, COO–, $CONH_2$, CO–N<, COX (X = Cl, Br, I) were not defined and, were represented as subtrees constituted by two atomic groups.

It can be observed that we chose to divide the CH group into C and H. In such a way we maintain the same approach in describing the C–H bond independent of the hybridisation of the carbon atom.

It is noteworthy that, by using the total order of the subtrees, we were able to build up different representations for the *cis* and the *trans* isomers of alkenes (see point 2.2 of Appendix A) and for the R and the S enantiomers of an optically active molecule (point 2.4 of Appendix A).

A numerical label is associated with each node. The labels discriminate among different groups of atoms and do not contain any physicochemical information.

We use a 1-of-n coding scheme for categorical variables. The labels are represented by 20-bit vectors, with one or a few specific bits turned on (+1) and all the others turned off (0). Sharing bits between different labels allows to represent similarity between chemical groups. On the other hand, two orthonormal vectors (*i.e.* bits turned on in different positions within the vector) represent groups of different chemical nature. In particular we stated that:

— All the H-groups have the same numerical label.
— CH_3, CH_2, C have similar numerical labels.
— NH_2, NH have similar numerical labels, but orthonormal to N.
— OH, O and SH, S have orthonormal labels.
— F, Cl, Br, I have similar numerical labels.

The similarity or the orthonormality between groups, decided according to their chemical features, is the maximum of "chemical information" transferred to the RecNN as input data.

In Figure 5 is reported, as an example, the representation of the 2-methyl-2 propanol as a chemical tree and the conversion of the tree in the input data file for the RecCC. The input data file contains the dimension of the tree (number of nodes), the value of the target property and the connection table of the structure. In the same figure, for the sake of clarity, the numerical labels corresponding to the groups present in 2-methyl-2 propanol are reported. This is the representation of the input data used to recursively read the structure by a recursive neural network: for each node, the RecCC exploits this table to decide which information encoded for the substructures of the node to use, as shown in Section 3.

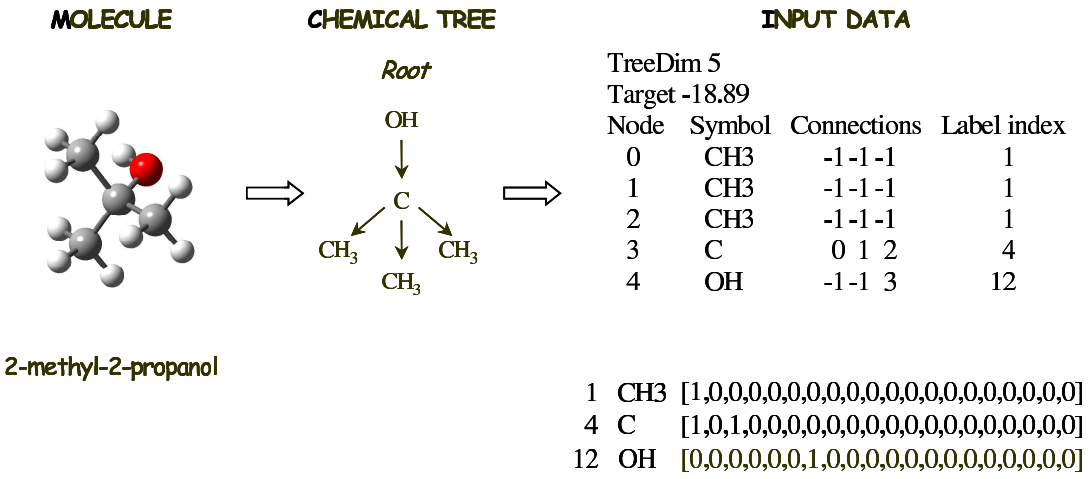

Figure 5. Representation of the 2-methyl-2-propanol as a chemical tree and input data file. The input data file contains, starting from the top, the dimension of the tree (number of nodes), the value of the target property and the connection table of the structure. In the first column of the connection table the order number identifying a specific node is reported, which corresponds to the group indicated in column 2; columns from 3 to 5 indicate for each node the presence of a "child" identified by its order number, i.e. a pointer to the substructure rooted in the child (–1 means the absence of a child); column 6 reports the number identifying the numerical label associate to the group. At the bottom of the figure the numerical label for CH_3, C and OH groups are reported.

RESULTS AND DISCUSSION

1. Model Evaluation

In order to investigate the application of RecCCs as a general and powerful predictive tool to model the solvation process, we studied the standard free energy of solvation, $\Delta_{solv}G°$, in water of a set of 179 linear molecules[6] including: alkanes, alkenes, alkynes, alcohols, ethers, thiols, thioethers, aldehydes, ketones, carboxylic acids, esters, amines, amides, haloalkanes, nitriles, nitroalkanes. The $\Delta_{solv}G°$ values were taken from Ref.6, which report an accurate critical revision of the experimental literature data of thermodynamic solvation functions and partial molar properties of a large number of organic compounds in water. The results of the RecCC procedure will be compared in this section with those obtained by applying the GCM method to the same set of compounds.

The whole set of molecules was divided in disjoint training and test set for the learning and the validation process respectively. The molecules were selected so that the test set was representative of the different molecular sizes and topologies and of all the functional groups, provided they were represented in the training set. Three different splitting strategies of the data were used.

In the first experiment the test set (33 compounds) was chosen as it contains the compounds giving the best performance with the group contributions method when applied to the whole set of 179 compounds. The target values in the training set were expressed in kJ mol^{-1} and ranged from –40 to 12 kJ mol^{-1}. A trial using dimensionless target values, normalised in the range 0÷1, was also run without any significant improvement in the performance.

In the second experiment the training set has been obtained from the first one by removing three compounds and specifically methane, in view of a comparison of the

results of the RecCC procedure with those of the standard GCM method (which cannot handle this molecule), and acetamide and fluoromethane that are the only representatives of the respective class of compounds in the molecules set. Moreover thiobismethane was moved from the test to the training set being the –S– group scarcely represented. The target values of the training set ranged in this case from –28 to 12 kJ mol^{-1}.

In the third case (experiment 3) fluoromethane has been removed from the data set. Moreover, three disjoint molecules set were defined. First a test set (33 compounds) was randomly chosen for the validation process while a 138 compounds set was used for the learning procedure, their target values ranging from –28 to 12 $kJmol^{-1}$. In this way we are able to obtain results not influenced by *a priori* choices and to test in the meantime the sensitivity of the network to different learning conditions. Finally the remaining seven compounds (methane, 1-buten-3-yne, 2-propen-1-ol, 2,4-hexadienal, acetamide, chloroethene, 3-chloro-1-propene) whose molecular features were scarcely represented in the whole data set were considered a "guess" test set. This special test set, whose predicted properties have to be evaluated punctually and not statistically, was designed in order to test the prediction ability of the RecCCs in some challenging conditions. More specifically, all these molecules, except methane, contain two or more atomic groups whose combination is not represented in any molecule of the training set. Methane represents a sort of outbound extrapolation from the alkanes series. In fact, while standard group contributions methods can't handle this molecule due to the fact that a group H bonded to a sp^3 C atom is not defined, the representation rules chosen in this application overcome this problem.

Eight trials were carried out for the simulation involving each one of the different training sets. The initial connection weights used in each simulation were randomly set. For experiments 1-3, learning was stopped when the RecCC inserted 100 hidden units, resulting in a mean training error less than 0.1 kJ mol^{-1}, this tolerance being below the experimental error on $\Delta_{solv}G°$. An experiment was also carried out in which learning was stopped when the maximum error for a single compounds was below 0.45 kJ mol^{-1} which is close to the standard deviation of the GCM method (experiment 4). The complete list of investigated compounds with the corresponding values of the target property ($\Delta_{solv}G°$) and of the mean error for each performed experiment are given in Appendix B.

The main statistics computed over all the experiments is reported in Table 1. Specifically the mean absolute error, the maximum absolute error, the correlation coefficient, and the standard deviation are reported as obtained by an ensemble averaging method, *i. e.* computing the mean output over the performed trials. The number of hidden units of each experiment is also reported.

As we can observe the training values, independently of the learning conditions, are reproduced within the experimental error (0.1 $kJmol^{-1}$) while in the test set the errors usually don't exceed 0.5 kJ mol^{-1}. It is noteworthy that this value matches the mean absolute error obtained by the GCM approach when applied to the whole set of 179 molecules. Moreover, the differences among the statistical parameters in the four experiments are quite low and only in the fourth one the standard errors increase appreciably. Indeed, this is a rather significant result since, as above mentioned, in case of experiment 4 the training procedure was stopped when a higher error threshold was attained and a corresponding lower number of hidden units was involved in the calculation. Since an early stopping of training does not improve the general performance of the model, we can take this result as a direct experimental proof that overtraining conditions have been avoided in all the experiments, the applied *i-strategy*

was effective, and the good fitting of data achieved in our experiments is not at expense of the predictive accuracy.

Table 1. Mean and maximum absolute error, correlation coefficient (R), standard deviation of error (S) and number of hidden units (U) of the different training data sets[a].

Exp. n°	U[b]	Training set				Test set			
		Mean abs. error[c]	Max abs. error[d]	R[e]	S[f]	Mean abs. error[c]	Max abs. error[d]	R[e]	S[f]
1	100	0.05	0.22	0.99998	0.07	0.40	1.19	0.9987	0.50
2	100	0.06	0.28	0.99996	0.09	0.53	2.63	0.9972	0.77
3	100	0.06	0.33	0.99998	0.08	0.46	2.12	0.9985	0.68
4	72[g]	0.09	0.43	0.99995	0.12	0.49	2.39	0.9982	0.73

[a]All the statistical parameters, except R, are expressed in kJ mol^{-1}. [b]Number of hidden units. [c]Mean absolute error defined as the mean of the absolute residuals. [d]Maximum absolute error. [e]Linear correlation coefficient between the experimental and calculated values. [f]Standard deviation. [g]Average of the number of hidden units over the eight trials.

As a general remark we can conclude that the different choice of the test set molecules does not significantly affect the results in view of the specific prediction task.

A punctual analysis of the residuals of both training and test set of each experiment allowed us to assess that no improvement in the average standard errors over the performed trials is achieved by not including the worst one (*i.e.* the one with the largest standard deviation). Moreover, it can be noticed that larger errors are frequently found for molecules with five carbon atoms or less. In particular C5 compounds are usually the most critical in all the classes. Though the adaptive process is probably less effective when a short, unbranched tree is processed by RecNN, at the moment no univocal simple explanation can be given for this trend.

For comparison aims the same training and test sets employed in experiment 3 were also analyzed by the group contributions method In particular a maximum absolute error of 2.74 and 2.10 kJ mol^{-1} and a standard deviation of 0.58 and 0.61 kJ mol^{-1} were found for the training and for the test set, respectively. As it can be observed the standard deviation of the test set is very similar to the value obtained by the RecCC model, whereas the standard deviation of the training set is much lower in the case of RecCC with respect to the GCM. This reflects the inherent flexibility of the new proposed model which abstracts, through the adaptive process, much more information and chemical knowledge from the data set with respect to the standard group contributions method. On the other hand the RecCC would need a rather wider data set in order to achieve a further improvement in its predicting ability. Figures 6 and 7 report the plots of the residuals of the training set and of the test set obtained by both RecCC and GCM in case of experiment 3.

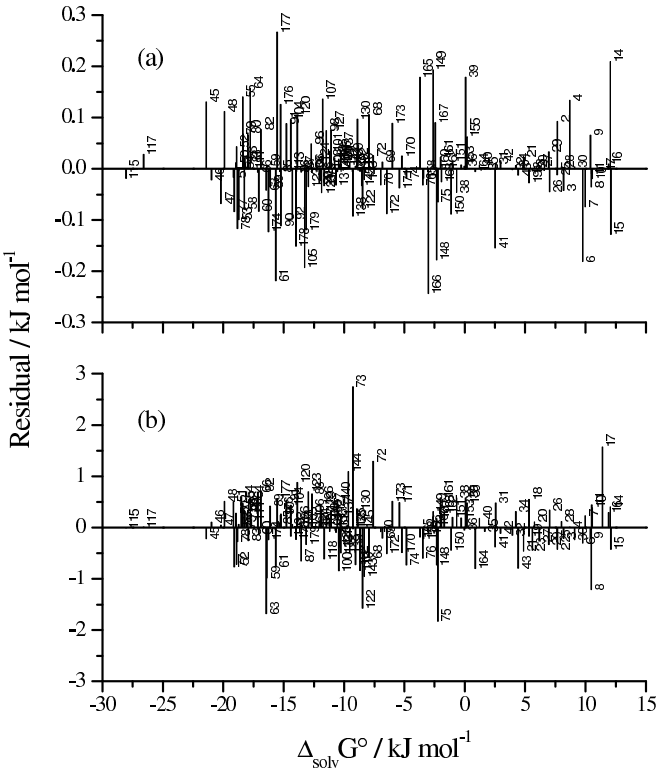

Figure 6. Plot of the residuals for the training set. (a), Recursive Cascade Correlation model; (b), Group Contributions Method. The compounds are individuated by their order number as indicated in Appendix B.

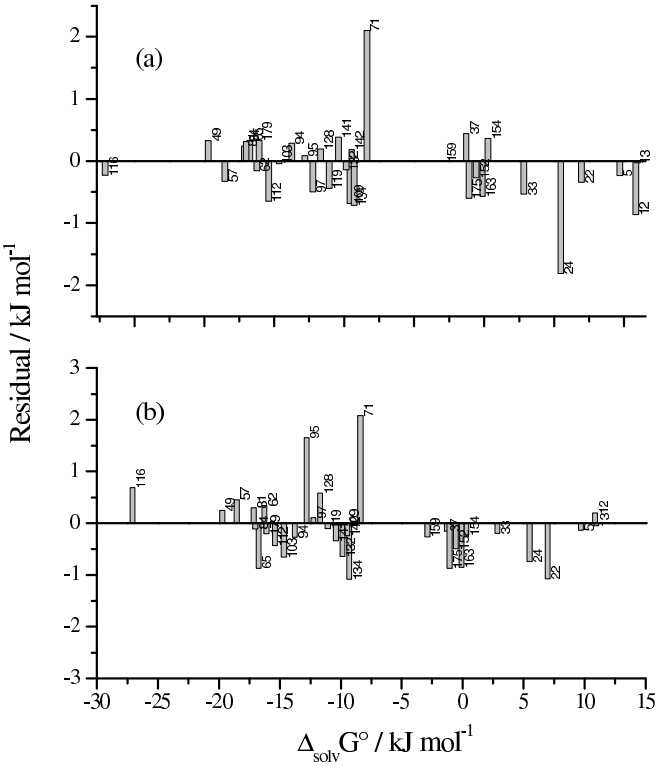

Figure 7. Plot of the residuals for the test set. (a), Recursive Cascade Correlation model; (b), Group Contributions Method. The compounds are individuated by their order number as indicated in Appendix B.

It is straightforward to observe that RecCC model gives residuals much lower in magnitude than those of GC method for the molecules of the training set, whereas the two methods show similar performance within the test set. In particular, some molecules exhibit significantly large errors with both methods [see *f.i.* methyl propyl ether (71); 2-methyl-2-butene (24)]. In such cases experimental errors cannot be excluded.

As already observed, molecules whose functional groups and/or structures are scarcely sampled in the training set tend to show larger errors.

In Table 2 the experimental Gibbs energies of solvation, of the molecules selected for the guess test set are reported together with the mean errors, δ, calculated as the difference between the predicted and the experimental values, for both RecCC and GCM approaches.

Table 2. Experimental $\Delta_{solv}G°$ values and mean errors for both Recursive Cascade Correlation, δ RecCC, and Group Contribution Method, δ GCM, in experiment 3.[a]

Molecule	$\Delta_{solv}G°$	δ RecCC[b]	δ GCM[b]
methane	8.37	1.15	
1-buten-3-yne	0.17	-5.02	-3.10
2-propen-1-ol	-21.06	-2.01	-1.91
2,4-hexadienal	-19.39	5.69	-0.97
acetamide	-40.63	15.69	
chloroethene	-2.48	-0.60	-2.68
3-chloro-1-propene	-2.4	-1.46	-2.01

[a]All data in kJ mol^{-1}. [b] Calculated as difference between the predicted and experimental value.

In the neural network model the average error is reported as computed over the eight trials of experiment 3, run in different learning conditions. It may be observed that the values predicted by the RecCC for methane, 2-propen-1-ol and for the two chloroalkenes are quite good and even better than those obtained with the GCM method whenever the comparison is possible. On the contrary we have a very large error for acetamide, but we have to stress that the $\Delta_{solv}G°$ data in the training set range from –28 to 12 kJ mol^{-1} so that a value of –40 kJ mol^{-1} is out of the experience of the network which correctly assign to acetamide a $\Delta_{solv}G°$ value close to the negative limit of the target data range. A comparison with the group contributions method is not possible in case of acetamide due to the lack of other occurrences of the amide group in the data set. A rather high error is also found in case of 1-buten-3-yne and 2,4-hexadienal. In both cases the RecCC is not trained at all over this kind of molecules. In the first case, the neural network can only infer the target property by averaging between the behaviour of alkenes and alkynes, thus predicting a more negative value. In fact the substitution of an ethyl group with either a –CH=CH_2 or a –C≡CH group produces a large decreasing in the $\Delta_{solv}G°$. The GCM only predicts a slightly better value. In the second case the RecCC tries to average between the behaviours of α, β unsaturated aldehydes and conjugated dienes. The predicted value is lower in magnitude with respect to the experimental one. Indeed we may also observe that only a few molecules included in the training set have highly negative $\Delta_{solv}G°$ values so that the neural network is scarcely trained within this range of target property. This probably concurs to the unsatisfactory prediction of the $\Delta_{solv}G°$ of 2,4-hexadienal.

2. Internal representations and domain knowledge

One of the most appealing issues of this RecCC application is the understanding whether the proposed model is able to capture significant domain knowledge from the training data. To do this we can investigate the internal representations, i.e. the output of hidden units (**X**) computed by the neural network trained with the selected set of molecules. These outputs represent the encoding values generated for each compound or molecular fragment. Therefore the analysis of internal representations could in principle be done at any level of the molecular tree. However, due to the quite high number of chemical classes of compounds and the complexity and variety of sampled molecular structures, we do prefer to analyse the internal representations only at the root level, i.e. by considering the whole molecules alone.

Since the number of hidden units is high and the dimension of the representational space is correspondingly large, we performed a Principal Component Analysis (PCA) of the internal representations and studied 2-D plots of the first two principal components. These plots show in a rather direct way the relative distance and position of the internal representation, approximately describing the knowledge learned by the neural network from the training data. As the results of the RecCC application are nearly independent of the learning conditions, we applied PCA analysis of the internal representations only to one of the experiments (experiment 3). Each one of the eight trials of experiment 3 were then separately analysed by PCA and the plots of the first two principal components from two of these trials are reported in Figure 8. Due to the large number of represented molecules we have split the set of compounds into three separate plots according to their basic chemical nature and namely hydrocarbons (alkanes, alkenes, alkynes), polar compounds (alcohols, ethers, amines), carbonyl compounds (aldehydes, ketones, carboxylic acids, esters). For the sake of simplicity the classes of compounds scarcely represented in the data set have been neglected.

At a first glance we can see that the molecules are clustered in a well defined area according to their own class. In particular it can be noticed that alkanes, alkenes and alkynes occupy the second, the first, and the fourth quadrant respectively, whereas all the other classes of compounds approximately lay in the third one. Since molecules pertaining to different classes and whose molecular tree differs from the root lay in the same region we can conclude that the observed groupings are not due to the simple fact that the internal representation maintains memory of the molecular graph. Indeed, these groupings represent the knowledge of some chemical features that cannot be directly inferred from the molecular graph rather only by the association of the molecular structure to the target property. In other words they represent what the recursive neural network did learn about the chemical features of the molecules.

In this frame there are a number of observation we can do by analyzing the distribution of the points in each class. For example as regards alkenes we can observe that, though generally laying in the first quadrant, some of them are scattered in the alkanes region. More specifically this happens for long chains or alkyl substituted alkenes, that is when the alkyl portion prevails in the molecule. On the other hand the internal representations demonstrate very effective in discriminating among isomeric compounds. For instance 2–methylpentane (10) and 3–methylpentane (11), as well as 2–methyl–1–pentene (27) and 3–methyl–1–pentene (28) lay significantly apart from each other in the plots. In particular, the case of the isomeric alkanes is meaningful since they cannot be discriminated by a standard group contributions method. On the contrary dienes, though laying in the alkenes region, show a quite unexpected behaviour: 1,3–butadiene (31) and 2,3–dimethyl–1,3–butadiene (35) move significantly going from trial III plot to trial V plot, but still remaining very close to each other.

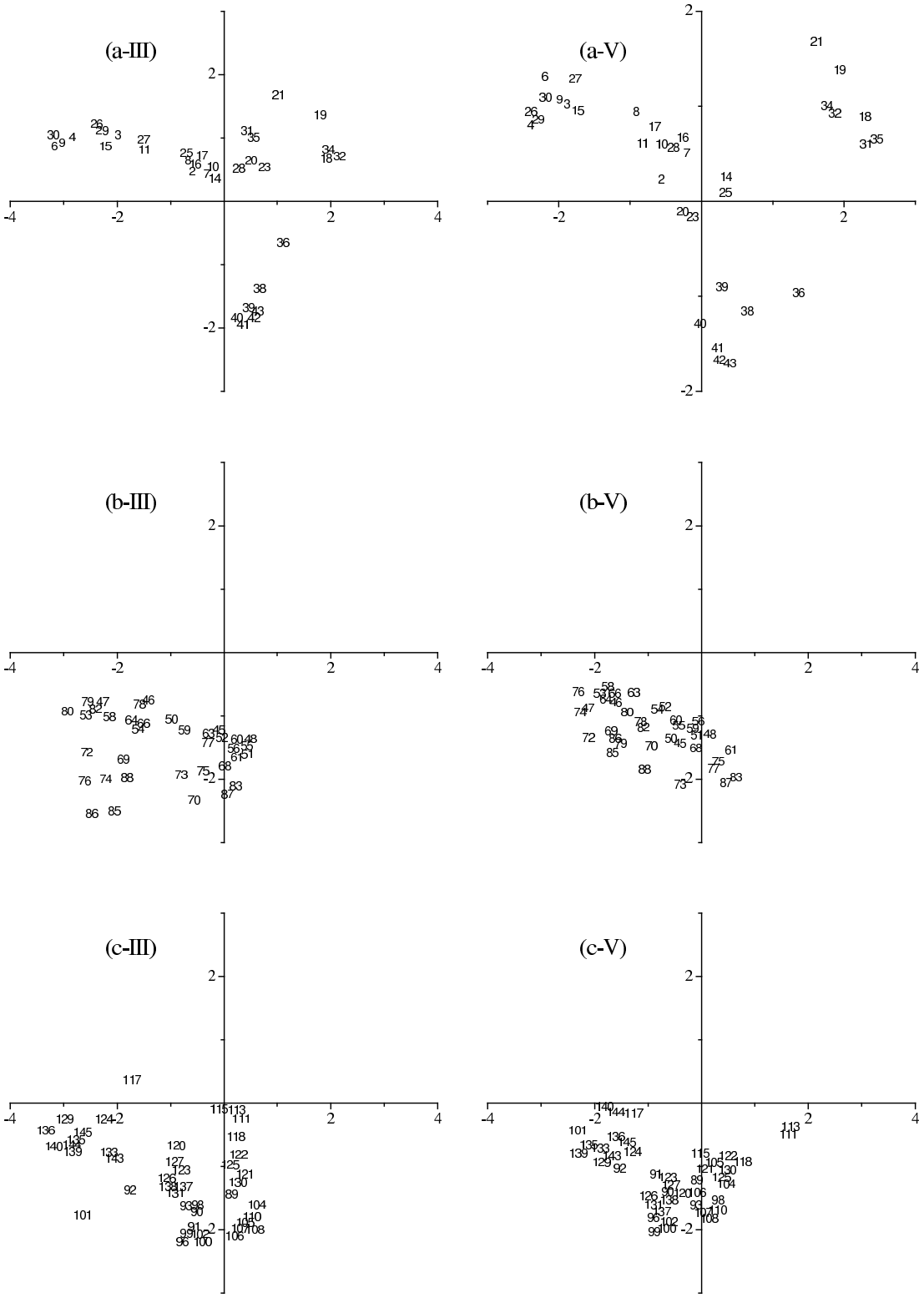

Figure 8. Plots of the two principal components of training compounds used in experiment 3 derived from trials III and V. (a), alkanes (2-17), alkenes (18-35), alkynes (36-43); (b), alcohols (45-66), ethers (68-76), amines (77-88); (c), ketones (89-102), aldehydes (104-113), carboxylic acids (115-117), esters (118-145). The compounds are individuated by their order number as indicated in Appendix B.

The most appealing features emerging from the PCA analysis probably concerns the behaviour of polar and carbonyl compounds. As above mentioned all of them are grouped in a big cloud laying in the third quadrant (and marginally in the fourth one). However, when the individual points are taken into consideration some very interesting results appear. For instance, it can be noticed that methanol (45) and methylamine (77), ethanol (46) and ethylamine (78), 1–propanol (47) and 1–propylamine (79) lay very close to each other, specially in the plot of trial III. This should mean that, in the representational space, an alkanol is much more similar to a primary amine of the same number of carbon atoms than the corresponding superior homologous. This is a largely unexpected result since either the trees of the two molecules have a completely different root and the target property is also different. Moreover alcohols and primary amines are clustered close to the upper boundary of the cloud whilst ethers and tertiary amines lay significantly below. This seems to suggest that the accessibility of the polar group by the solvent and its ability to act as a hydrogen bond donor and/or acceptor is responsible of the distribution of polar molecules in the representational space. This behaviour seems to be confirmed also by the plot of carbonyl compounds, where carboxylic acids and aldehydes lay close to the upper boundary while ketones and esters are located below.

It has to be stressed that this kind of chemical knowledge abstracted by the RecCC does not appear trivially decoupled into single effects. On the contrary the model combines these features developing a sort of "smooth rule", reflected by the spread of the points in the clusters, globally accounting for the complexity of the stereo-electronic properties of molecules.

CONCLUSIONS

The proposed Recursive Cascade Correlation model demonstrates very flexible and effective in dealing with the specific task it has been applied to. In particular, the model shows a better descriptive ability when compared with the group contributions method as the reference method in the field, while both approaches reveal nearly the same predictive power. However, the comparison between the statistical parameters obtained by both methods for the training and the test set and a punctual analysis of the results suggest that while the GCM is at the best of its performance, the predicting ability of RecCC model should be significantly improved by increasing the learning basis. In fact, the worst results are obtained in case of molecules whose functional groups and/or structures are scarcely represented in the training set. Indeed, we plan to extend this study to sets of compounds spanning over a wider survey of chemical structures and functionalities and including bi- and poly-functional, cyclic, aromatic, and eteroaromatic compounds, where it is well known that the group contributions method shows theoretical limitations and, consequently, worse results. We hope in this way to reach the ultimate goal of extending the applicability of the prediction methods to compounds and problems not yet covered by the standard QSPR approaches.

We can conclude that, due to the amount of chemical knowledge the RecCC model is able to abstract in the learning process, it appears as a reliable alternative to traditional QSPR methods in instituting structure – property correlations.

Acknowledgements. This work has been financially supported by Ministero dell'Istruzione, dell'Università e della Ricerca, Roma (Cofin 2001 and 2003 MIUR).

REFERENCES

(1) English, N.J.; Carroll, D.G. *J. Chem. Inf. Comput. Sci.* **2001,** *41*, 1150.

(2) Yaffe, D.; Cohen, Y.; Espinosa, G.; Arenas, A.; Giralt, F. *J. Chem. Inf. Comput. Sci.* **2003,** *43*, 85 and references therein.

(3) (a) Hansch, C. *Acc. Chem. Res.* **1969**, *2*, 232. (b) Hansch, C.; Leo, A. *Substituent Constants for Correlation Analysis in Chemistry and Biology*; Wiley: New York, 1979. (c) Topliss, F. G. *Quantitative Structure Activity Relationships of Drugs*; Academic Press: New York, 1983.

(4) (a) Leo, A.; Hansch, C.; Elkins, D. *Chem. Rev.* **1971**, *71*, 525. (b) Smith, R. N.; Hansch, C.; Ames, M. M. *J. Pharm. Sci.* **1975**, *64*, 599. (c) Roberts, M. S.; Pugh, W. J.; Hadgraft, J.; Watkinson, A. C. *Int. J. Pharm.* **1995**, *126*, 219. (d) Gratton, J. A.; Abraham, M. H.; Bradbury, M. W.; Chadha, H. S. *J. Pharm. Pharmacol.* **1997**, *49*, 1211.

(5) (a) Dallas, A. J.; Carr, P. W. *J. Chem. Soc., Perkin Trans. 2,* **1992**, 2155. (b) Bernazzani, L.; Cabani, S.; Conti, G.; Mollica, V. *J. Chem. Soc., Faraday Trans.* **1995**, *91*, 649.

(6) Cabani, S.; Gianni, P.; Mollica, V.; Lepori, L.; *J. Solution Chem.* **1981**, *10*, 563.

(7) (a) Cabani, S.; Conti, G.; Mollica, V.; Bernazzani, L. *J. Chem. Soc., Faraday Trans. I*, **1991**, *87*, 2433. (b) Bernazzani, L.; Carosi, M. R.; Ceccanti, N.; Conti, G.; Gianni, P.; Mollica, V.; Tiné, M. R.; Lepori, L.; Matteoli, E. *Phys. Chem. Chem. Phys.* **2000**, *2*, 4829.

(8) (a) Abraham, M. H. *J. Am. Chem. Soc.* **1982**, *104*, 2085. (b) Fuchs, R.; Stephenson, W. K. *Can. J. Chem.* **1985**, *63*, 349. (c) Lepori, L.; Matteoli, E.; Marongiu, B. *Fluid Phase Equilib.* **1988**, *42*, 229. (d) Abraham, M. H.; Whiting, G. S.; Fuchs, R.; Chambers, E. J. *J. Chem. Soc., Perkin Trans. 2*, **1990**, 291.

(9) (a) Hine, J.; Mookerjee, P.K.; *J Org. Chem.* **1975**, *40*, 292. (b) Platts, J. A.; Abraham, M. H.; Butina, D.; Hersey, A. *J. Chem. Inf. Comp. Sci.* **2000**, *40*, 71.

(10) Guthrie, J. *P. Can. J. Chem.* **1992**, *70*, 1042.

(11) (a) Medir, M.; Giralt, F. *AIChE J.* **1982**, *28*, 341. (b) Zhong, C.; Yang, C.; Li, Q. *Ind. Eng. Chem. Res.* **2002**, *41*, 2826. (c) Santiuste, J. M. *J. Chrom. Sci.* **2003**, *41*, 215.

(12) (a) Abraham, M. H.; Platts, J. A. *J. Org. Chem.* **2001**, *66*, 3484. (b) Mancini, P. M.; Perez, A. del C.; Vottero, L. R. *J Solution Chem.* **2001**, *30*, 695.

(13) See *f. i.*: (a) Abraham, M. H. *Chem. Soc. Rev.*, **1993**, 73. (b) Abraham, M. H.; Zissimos, A. M.; Acree, W. E. Jr. *Phys. Chem. Chem. Phys.*, **2001**, *3*, 3732.

(14) Sperduti, A.; Starita, A. *IEEE Transactions on Neural Networks* **1997**, *8*, 714.

(15) Frasconi, P.; Gori, M.; Sperduti, A. *IEEE Transactions on Neural Networks* **1998**, *9*, 768.

(16) Bianucci, A. M.; Micheli, A.; Sperduti, A.; Starita, A. *Appl. Int. J.* **2000**, *12*, 117.

(17) Micheli, A.; Sperduti, A.; Starita, A.; Bianucci, A. M. *J. Chem. Inf. Comput. Sci.* **2001**, *41*, 202.

(18) Micheli, A.; Sperduti, A.; Starita, A.; Bianucci, A. M. In *Soft Computing Approaches in Chemistry*; Sztandera, L. M.; Cartwright, H. M. Eds.; Springer-Verlag: Heidelberg, 2003**,** pp 265-296.

(19) Benson, S. W. *Thermochemical Kinetics*; Wiley: New York, 1976.

(20) Ducros, M.; Gruson, J.F.; Sannier, H.; Velasco, L. *Thermochim. Acta* **1981**, *44*, 131.

(21) (a) Reid, R. C.; Prausnitz, J. M.; Poling, B. E. *Properties of gases and liquids*, 4th ed.; McGraw-Hill: New York, 1987 and references therein. (b) Iwai, Y.; Yamanaga, S.; Arai, Y. *Fluid Phase Equil.* **1999**, *163*, 1.

(22) (a) Li, P.; Ma, P.-S.; Yi, S.-Z.; Zhao, Z.-G.; Cong, L.-Z. *Fluid Phase Equil.* **1995**, *106*, 284. (b) N'Guimbi, J.; Berro, C.; Mokbel, I.; Rauzy, E.; Jose, J. *Fluid Phase Equil.* **1999**, *162*, 143.

(23) Edward, J. T.; Farrell, P. G.; Shahidi, F. *Can. J. Chem.* **1979**, *57*, 2585.

(24) Ben-Naim, A. *Solvation Thermodynamics*; Plenum Press: New York and London, 1987.

(25) Fahlman, S. E.; Lebiere, C. In *Advances in Neural Information Processing Systems 2*; Touretzky, D. S., Ed.; Morgan Kaufmann Publishers: San Mateo, CA, 1990; pp 524-532.

(26) Fahlman, S. E. In *Advances in Neural Information Processing Systems 3*; Lippmann, R. P.; Moody, J. E.; Touretzky, D. S., Eds.; Morgan Kaufmann Publishers: San Mateo, CA, 1991; pp 190-196.

(27) Bartelett, P. L. *IEEE Transaction on Inform. Theory* **1998**, *44*, 525.

APPENDICES

Appendix A. *Rules for the representation of the molecular structure*

We have represented the molecular structures in terms of labeled directed ordered trees. In order to obtain an unique structured representation of each molecule we decided a set of general rules that can be summarised as follows:

a. Each molecule is partitioned in the following groups: CH_3, CH_2, C, H, C=C, C≡C, OH, O, C=O, NH_2, NH, N, SH, S, CN, NO_2, F, Cl, Br, I.
b. Each group corresponds to a node of the tree and each bond between them corresponds to an edge.
c. We defined an 'ad hoc' priority scale for chemical groups and the root of the tree is fixed on the group with the highest priority.
d. The orientation of the edges follows the increasing levels of the trees.
e. The total order on the subtrees of each node is defined according to the same priority among chemical groups used to state the root.

1. Priority scale:

1.1. Molecules containing only carbon and hydrogen atoms.

1.1.1. Alkanes: The groups of the longest chain are numbered beginning with the end that is closest to an alkyl substituent and the root is fixed on the first group. If two (or more) alkyl substituents are present at equal distance from the two ends of the longest chain, the elements along the substituent chains are ranked until a point of difference is reached at which a distinction in priority is possible. The substituent of lowest priority is hydrogen (Es. $-CH_2-CH_3 > -CH_3$; $CH_2-C(CH_3)_3 > -CH_2-CH_3$). The groups of the longest chain are then numbered beginning with the end closest to the higher priority branched chain.

1.1.2. Alkenes and dienes: the main chain is the longest one that includes the group C=C. Numbering of the chain starts at the end farther from the double bond and the root is set on the first group. In dienes with both terminal double bonds the root is set on the apical hydrogen atom of the more substituted double bond.

1.1.3. Alkines and enines: the root is set on the triple bond.

1.2. Molecules containing heteroatoms (N, O, S, F, Cl, Br, I).

1.2.1. A group containing a heteroatom has higher priority than any other functional group and the root is set on it.

1.2.2. In molecules with different groups, the C=O has the highest priority. This choice allows to give a similar representation to ketones, aldheydes, carboxylic acids, esthers, alkanoyl halides, anhydrides, and amides. As far as the other groups are concerned the priority is assigned on the basis of the heteroatm. The priority decreases going to the right in the group of the periodic table (N>O>F) and down in the period (O>S; F>Cl>Br>I). Fixed the heteroatom, the priority among the groups follows the order: (a) nitrogen N> NH >NH2> NO >NO_2; (b) oxygen OH> O; c) sulphur C=S> S> SH >S=O.

1.2.3. If two or more identical functional groups are present the root is set on the inner one in order to minimize the depth of the structure.

1.2.4. In polyhaloalkanes the root is fixed on the highest priority halogen atom bound to the carbon bearing the highest number of the same halogen (Es. $F_{root}-CF_2-CF_2Cl$). In polyhaloalkenes the root is fixed on the highest

priority halogen atom bound to the sp_2 carbon bearing the highest number of the same halogen (Es. F_{root}-CF=CF-CF_3).

2. Edges order:

2.1. The edges starting from a node are ordered following the groups priority rules. If two (or more) substituents in a node have the same priority the elements along the substituent chains are ranked until a point of difference is reached at which a distinction in priority is possible.

2.2. In a double bond the edges are numbered starting from the *cis* position with respect to the root and follow the order *cis*>*trans*>*gem*. When the double bond stereoisomery is *cis*, absent or not specified, positions 1 and 2 are occupied according to the priority rules of the groups. If the double bond stereoisomery is *trans* the higher priority group occupies position 2.

2.3. In a triple bond the edges are ordered following the groups priority rules

2.4. We can distinguish R and S enantiomers by ordering the edges of the asymmetric carbon according to the priority rules. The second enantiomer is obtained by changing the order of two edges.

Appendix B. *Values of the target property ($\Delta_{solv}G°$) and of the mean error, δ, of each performed experiment for all the examined compounds.*[a]

Order N°	Compound	$\Delta_{solv}G°$[b]	δ 1	δ 2	δ 3	δ 4
1	Methane	8.37	0.070	-	-	-
2	Ethane	7.66	-0.164	-0.187	0.111	0.209
3	Propane	8.18	-0.015	-0.082	-0.116	-0.138
4	Butane	8.70	0.026	0.123	0.083	0.121
5	2-Methylpropane	9.70	-0.409[c]	-0.532[c]	-0.158[c]	-0.252[c]
6	Pentane	9.76	-0.051	-0.090	-0.072	-0.176
7	2-Methylbutane	9.97	-0.022	-0.073	-0.043	-0.145
8	2,2-Dimethylpropane	10.46	-0.069	-0.084	-0.044	-0.068
9	Hexane	10.40	0.099[c]	0.032[c]	0.034	0.026
10	2-Methylpentane	10.56	-0.026	-0.004	-0.020	-0.009
11	3-Methylpentane	10.50	-0.002	0.010	0.006	-0.021
12	2,2-Dimethylbutane	10.85	0.153[c]	0.451[c]	0.276[c]	0.143[c]
13	Heptane	10.96	0.039	0.104	0.171[c]	0.146[c]
14	2,4-Dimethylpentane	12.04	0.102	0.169	0.209	0.300
15	Octane	12.10	-0.095	-0.176	-0.164	-0.196
16	2,2,4-Trimethylpentane	11.93	0.010	0.009	0.000	0.015
17	2,2,5-Trimethylhexane	11.39	-0.014	0.000	0.003	0.004
18	Ethene	5.32	-0.020	0.007	-0.015	-0.014
19	1-Propene	5.31	0.094	0.034	0.014	0.016
20	1-Butene	5.77	0.129	0.191	0.006	-0.004
21	2-Methyl-1-propene	4.87	0.010	0.008	-0.002	-0.023
22	1-Pentene	6.96	-0.011	-0.011	-0.662[c]	-0.950[c]
23	*trans*-2-Pentene	5.59	0.474[c]	2.015[c]	-0.004	0.029
24	2-Methyl-2-butene	5.48	-0.014	0.002	-0.537[c]	-0.844[c]
25	3-Methyl-1-butene	7.65	-0.014	0.002	-0.006	0.014
26	1-Hexene	7.02	0.009	0.019	-0.055	-0.067
27	2-Methyl-1-pentene	6.15	0.004	0.014	-0.001	0.045
28	4-Methyl-1-pentene	7.99	-0.010	-0.010	-0.001	-0.035
29	*trans*-2-Heptene	6.96	1.190[c]	2.628[c]	0.060	0.070
30	1-Octene	9.08	0.027	0.039	0.007	0.010
31	1,3-Butadiene	2.57	-0.012	-0.007	0.012	-0.002
32	1,4-Pentadiene	3.94	-0.598[c]	-0.758[c]	0.004	0.032
33	2-Methyl-1,3-butadiene	2.85	0.012	0.007	-0.147[c]	-0.171[c]
34	1,5-Hexadiene	4.22	0.011	0.005	0.000	-0.005
35	2,3-Dimethyl-1,3-butadiene	1.66	0.006	-0.006	-0.008	-0.005
36	Ethyne	-0.05	0.005	-0.005	0.002	-0.009
37	1-Propyne	-1.28	0.162	0.148	0.502[c]	0.181[c]
38	1-Butyne	-0.68	-0.088	-0.021	-0.008	-0.021
39	1-Pentyne	0.06	0.003	0.000	0.104	0.202
40	1-Hexyne	1.20	-0.038[c]	0.158[c]	-0.012	-0.019
41	1-Heptyne	2.51	-0.045	-0.092	-0.034	-0.154
42	1-Octyne	2.97	0.045	0.028	-0.020	-0.064
43	1-Nonyne	4.40	-0.062	-0.054	-0.032	0.079
	1-Buten-3-yne					

45	Methanol	-21.40	0.206	0.211	0.137	0.092
46	Ethanol	-20.98	-0.102	-0.158	-0.136	-0.184
47	1-Propanol	-20.19	-0.438[c]	-0.051[c]	0.050	0.009
48	2-Propanol	-19.90	0.079	0.137	0.144	0.252
49	1-Butanol	-19.73	0.036	0.157	0.156[c]	0.165[c]
50	2-Methyl-1-propanol	-18.93	1.141[c]	1.281[c]	-0.010	0.016
51	2-Butanol	-19.15	-0.058	0.029	0.002	-0.007
52	2-Methyl-2-propanol	-18.89	0.084	0.058	0.105	0.132
53	1-Pentanol	-18.72	-0.099	-0.111	-0.123	-0.164
54	3-Methyl-1-butanol	-18.50	0.013	-0.003	0.003	0.004
55	2-Pentanol	-18.38	0.109	0.081	0.142	0.230
56	3-Pentanol	-18.22	0.208[c]	-0.103[c]	-0.064	-0.015
57	2-Methyl-2-pentanol	-18.54	0.215[c]	0.760[c]	0.555[c]	0.610[c]
58	1-Hexanol	-18.26	-0.011	-0.173	-0.111	-0.094
59	2,3-Dimethyl-2-butanol	-16.38	-0.004	-0.002	0.000	-0.007
60	3-Hexanol	-17.05	-0.083	-0.032	-0.054	-0.146
61	4-Methyl-2-pentanol	-15.64	-0.104	-0.183	-0.212	-0.315
62	2-Methyl-3-pentanol	-16.26	0.005	-0.005	0.375[c]	0.783[c]
63	2-Methyl-2-pentanol	-16.44	-0.003	0.006	-0.065	-0.092
64	1-Heptanol	-17.76	0.125	0.153	0.180	0.294
65	4-Heptanol	-16.76	-0.006	0.002	-0.620[c]	-0.640[c]
66	1-Octanol	-17.13	-0.019	0.012	0.013	-0.001
67	2-Propen-1-ol	-21.06	-0.175	-0.187	-	-
68	Dimethyl ether	-7.93	-0.003	-0.008	0.039	0.220
69	Diethyl ether	-6.84	-0.032	-0.149	-0.025	-0.159
70	Methyl propyl ether	-6.96	0.791[c]	-0.077[c]	0.005	0.032
71	Isopropyl methyl ether	-8.39	0.072	0.235	2.117[c]	2.395[c]
72	Ethyl propyl ether	-7.59	0.008	0.053	0.061	0.145
73	*tert*-Butyl methyl ether	-9.25	0.002	-0.006	0.005	0.024
74	Dipropylether	-4.83	0.458[c]	-0.558[c]	-0.051	-0.145
75	Diisopropyl ether	-2.23	-0.053	-0.112	-0.065	-0.179
76	Dibutylether	-3.48	-0.008	-0.018	0.027	0.028
77	Methylamine	-19.09	-0.085	-0.016	-0.002	-0.018
78	Ethylamine	-18.84	0.024	-0.137	-0.143	-0.174
79	Propylamine	-18.37	0.337[c]	0.124[c]	0.019	0.084
80	Butylamine	-17.97	0.061	0.136	0.045	0.046
81	Pentylamine	-17.14	0.024	-0.071	-0.104[c]	-0.179[c]
82	Hexylamine	-16.87	-0.028	0.089	0.093	0.095
83	Dimethylamine	-17.93	0.023	-0.015	-0.005	-0.062
84	Diethylamine	-17.02	-0.198[c]	-0.008[c]	-0.231[c]	-0.298[c]
85	Dipropylamine	-15.31	-0.026	0.026	0.029	0.054
86	Dibutylamine	-13.92	0.009	-0.005	-0.026	-0.016
87	Trimethylamine	-13.55	0.002	0.007	-0.006	-0.005
88	Triethylamine	-12.65	-0.004	-0.006	0.004	0.003
89	2-Propanone	-16.12	0.050	0.210	0.040	0.011
90	2-Butanone	-15.22	-0.038	-0.108	-0.110	-0.155
91	2-Pentanone	-14.76	0.009	0.042	0.105	0.065
92	3-Pentanone	-14.28	-1.035[c]	-0.037[c]	-0.085	-0.110
	3-Methyl-2-butanone					

94	2-Hexanone	-13.76	0.043	-0.001	-0.150[c]	-0.106[c]
95	4-Methyl-2-pentanone	-12.81	0.518[c]	1.572[c]	1.576[c]	1.527[c]
96	2-Heptanone	-12.72	-0.062	0.011	-0.053	0.064
97	4-Heptanone	-12.24	0.011	-0.064	-0.058[c]	-0.137[c]
98	2,4-Dimethyl-3-pentanone	-11.46	0.049	0.106	0.071	0.161
99	2-Octanone	-12.06	0.345[c]	0.278[c]	-0.021	0.055
100	2-Nonanone	-10.41	-0.017	0.021	0.024	-0.072
101	5-Nonanone	-11.18	0.013	0.027	-0.004	-0.036
102	2-Undecanone	-9.05	0.009	0.005	-0.003	0.003
103	Ethanal	-14.66	-0.222	-0.215	-0.355[c]	-0.558[c]
104	Propanal	-14.40	0.144	0.089	0.080	0.105
105	Butanal	-13.26	-0.637[c]	-0.583[c]	-0.189	-0.187
106	Pentanal	-12.68	-0.054	-0.006	0.003	0.014
107	Hexanal	-11.76	0.074	0.069	0.121	0.132
108	Heptanal	-11.18	0.451[c]	0.168[c]	0.029	-0.012
109	Octanal	-9.58	0.045	0.047	-0.153[c]	-0.160[c]
110	Nonanal	-8.69	-0.006	-0.013	0.008	-0.013
111	*trans*-2-Butenal	-17.68	0.013	0.004	0.006	0.000
112	*trans*-2-Hexenal	-15.40	0.041[c]	-0.633[c]	-0.432[c]	-0.427[c]
113	*trans*-2-Octenal	-14.40	-0.005	0.000	-0.004	-0.003
114	*trans*,*trans*-2,4-Hexadienal	-19.39	0.006	-0.001	-	-
115	Ethanoic acid	-28.05	-0.011	-0.005	-0.057	-0.044
116	Propanoic acid	-27.09	-0.588[c]	-0.490[c]	0.190[c]	0.263[c]
117	Butanoic acid	-26.59	0.016	0.010	0.063	0.062
118	Methyl methanoate	-11.64	-0.087	-0.059	-0.040	-0.100
119	Ethyl methanoate	-11.07	-0.564[c]	-0.478[c]	0.140[c]	-0.143[c]
120	Methyl Ethanoate	-13.87	0.074	0.099	0.057	0.268
121	Propyl methanoate	-10.39	0.020	0.028	-0.021	0.017
122	1-Methylethyl methanoate	-8.45	-0.009	0.022	-0.059	-0.104
123	Ethyl ethanoate	-12.95	-0.023	-0.064	-0.027	-0.060
124	Methyl propanoate	-12.27	0.000	0.036	0.046	-0.063
125	2-Methylpropyl methanoate	-9.30	0.618[c]	1.090[c]	-0.003	-0.007
126	Propyl ethanoate	-11.95	-0.016[c]	-0.260[c]	0.064	-0.020
127	1-Methylethyl ethanoate	-11.07	0.036	0.007	0.058	0.106
128	Ethyl propanoate	-11.70	0.046	0.046	-0.067[c]	0.147[c]
129	Methyl butanoate	-11.85	0.023	-0.040	-0.063	-0.121
130	3-Methylbutyl methanoate	-8.90	0.012	0.041	0.065	0.132
131	Butyl ethanoate	-10.67	0.042	0.001	-0.044	-0.084
132	2-Methylpropyl ethanoate	-9.87	0.201[c]	0.253[c]	-0.370[c]	-0.545[c]
133	Propyl propanoate	-10.27	-0.028	-0.034	-0.010	-0.005
134	1-Methylethyl propanoate	-9.30	-0.032	-0.030	-1.536[c]	-1.265[c]
135	Ethyl butanoate	-10.44	-0.008	0.003	0.024	0.065
136	Methyl pentanoate	-10.77	0.355[c]	0.328[c]	0.029	0.074
137	Pentyl ethanoate	-10.27	-0.041	-0.012	0.008	0.123
138	3-Methylbutyl ethanoate	-9.25	-0.002	-0.045	-0.063	-0.117
139	Propyl butanoate	-9.53	-0.392[c]	-0.079[c]	-0.019	0.057
140	Ethyl pentanoate	-10.56	-0.023	0.015	0.003	0.004
141	Methyl hexanoate Hexyl Ethanoate	-10.41	-0.005	-0.032	-0.331[c]	-0.286[c]

143	Pentyl propanoate	-8.33	0.007	0.007	0.006	-0.084
144	Ethyl heptanoate	-9.64	0.002	0.006	0.000	-0.035
145	Methyl Octanoate	-8.53	-0.008	0.013	-0.020	-0.060
146	Acetamide	-40.63	-0.006	-	-	-
147	Fluoromethane	-0.92	0.003	-	-	-
148	Chloromethane	-2.33	-0.206	-0.032	-0.167	-0.197
149	Chloroethane	-2.63	0.184	0.152	0.180	0.265
150	1-Chloropropane	-1.14	-0.097	-0.094	-0.124	-0.211
151	2-Chloropropane	-1.03	0.004[c]	-0.060[c]	-0.059	-0.073
152	1-Chlorobutane	-0.57	0.016	-0.110	-0.555[c]	-0.471[c]
153	1-Chloropentane	-0.29	0.053	0.150	0.069	0.086
154	2-Chloropentane	0.29	0.013	-0.044	0.158[c]	0.046[c]
155	3-Chloropentane	0.17	0.076	0.033	0.105	0.152
156	Chloroethene	-2.48	-0.082	-0.048	-	-
157	3-Chloro-1-propene	-2.40	0.045	-0.005	-	-
158	Bromomethane	-3.43	-0.018	-0.032	-0.015	-0.040
159	Bromoethane	-2.91	-0.129	-0.130	-0.128[c]	-0.217[c]
160	1-Bromopropane	-2.34	0.097	0.129	0.081	0.061
161	2-Bromopropane	-2.00	0.052	0.048	0.021	0.072
162	1-Bromobutane	-1.71	0.269[c]	0.236[c]	-0.076	-0.080
163	1-Bromo-2-methylpropane	-0.11	0.002	-0.008	-1.022[c]	-0.698[c]
164	1-Bromo-3-methylbutane	0.86	-	-	-0.008	-0.020
165	lodomethane	-3.71	0.223	0.161	0.166	0.111
166	lodoethane	-3.03	-0.121	-0.046	-0.168	-0.136
167	1-Iodopropane	-2.45	0.120[c]	0.284[c]	0.035	0.060
168	2-Iodopropane	-1.94	-0.055	-0.027	-0.033	-0.034
169	1-Iodobutane	-1.08	-0.060	-0.104	-0.951[c]	-0.961[c]
170	Methanethiol	-5.19	-0.068	-0.096	-0.052	-0.127
171	Ethanethiol	-5.42	0.055	0.103	0.048	0.113
172	Dimethyl sulphide	-6.45	-0.160[c]	-0.124	-0.076	-0.171
173	Diethyl sulphide	-5.99	0.012	0.126	0.077	0.193
174	Ethanenitrile	-16.26	0.015	-0.023	-0.103	-0.230
175	Propanenitrile	-16.09	-0.011	0.021	-0.244[c]	-0.177[c]
176	Butanenitrile	-15.25	0.209[c]	0.553[c]	0.110	0.233
177	Nitroethane	-15.52	0.146	0.280	0.327	0.434
178	1-Nitropropane	-13.98	-0.070	-0.173	-0.198	-0.241
179	2-Nitropropane	-13.13	-0.065	-0.111	-0.140	-0.216

[a]All data are in kJ mol^{-1}. Mean errors δ are defined as the difference between calculated and experimental values of the target property averaged over the performed trials. [b]Taken from Ref. 6. [c]Compound included in the test set.