The following full text is a publisher’s version.

For additional information about this publication click this link.
http://hdl.handle.net/2066/28227

Please be advised that this information was generated on 2019-10-06 and may be subject to change.
PARALLEL PROCESSING OF CHEMICAL INFORMATION IN A LOCAL AREA NETWORK—III. USING GENETIC ALGORITHMS FOR CONFORMATIONAL ANALYSIS OF BIOMACROMOLECULES

M. L. M. BECKERS,* E. P. P. A. DERKS, W. J. MELSSEN and L. M. C. BUYDENS
Laboratory for Analytical Chemistry, Faculty of Science, Catholic University of Nijmegen, Toernooiveld 1, 6525 ED Nijmegen, The Netherlands

(Received 19 June 1995; in revised form 31 August 1995)

Abstract—Multi-dimensional nuclear magnetic resonance experiments are an excellent means of revealing the three-dimensional structure of biomacromolecules in solution. However, the search space in the conformational analysis of biomacromolecules, using multi-dimensional NMR data, is huge and complex. This calls for global optimization techniques with good sampling properties. This paper describes a genetic algorithm that optimizes the fit between (simulated) experimental two-dimensional Nuclear Overhauser Effect spectra and the corresponding calculated spectra for trial structures. This is a very computational intensive procedure. Speed-up of performance is achieved by parallelizing the algorithm, i.e. creating small subpopulations of trial structures, each of which can be processed on different processors. Good sampling behavior is obtained by initializing each subpopulation with its own random seed and the introduction of a migration operator. The latter replaces the best performing individual from one subpopulation with the worst performing individual from another subpopulation after a predetermined number of generations. A parallel genetic algorithm for the conformational analysis of nucleic acids is developed using the software package HYDRA. It is demonstrated that, for the data sets used in the study, a considerable reduction in computation time is obtained for the parallel genetic algorithm as compared to a sequential implementation, while the same optimal solutions are found.

1. INTRODUCTION

The three-dimensional spatial structure or conformation of biomacromolecules is strongly related to their biological activity. Nowadays, much attention is paid to the conformational analysis of interesting molecules like, e.g. nucleic acids and proteins. In general, the analysis of molecular conformations is subdivided into two steps. First, a number of trial conformations of a molecule is generated. Then the conformations are evaluated according to their agreement with available experimental data or according to their energetic reasonableness. Spectroscopic techniques, in particular Nuclear Magnetic Resonance (NMR) spectroscopy, are the most important methods to study molecular conformations in solution (Wüthrich, 1986; van de Ven & Hilbers, 1988). After assigning the resonances in NMR experiments, and possibly estimating an initial structure, a refinement of trial structures has to be performed until the resulting conformations are in agreement with the experimentally obtained NMR data. The conformation space, which is constituted by all possible candidate conformations, usually is very large, even for relatively small biomacromolecules. For example, a nucleic acid consisting of two nucleotides can be described by 16 torsion angles. When each torsion angle needs to be known with a precision of, say, 0.1% of its confined range, this yields a search space of $10^{46}$ possible molecular conformations (Lucasius et al., 1991). Obviously, the number of possible solutions contained in the search space is too large to be explored by a systematic (exhaustive) search method in order to find the global optimum (i.e., the best conformation) within reasonable time limits. Additionally, commonly applied conformational analysis techniques such as, e.g. the distance geometry methods, are basically gradient descent based search methods and, hence, tend to end up in local optima in the conformational space. To circumvent this problem, stochastic or dynamic simulations are invoked in the conformational analysis as well. However, the final outcome of these simulation techniques strongly depends on the set of initial trial structures (Crippen, 1983; van Gunsteren et al., 1984; Metzler et al., 1989). Often the distance geometry method is applied to provide reasonable initial structures for molecular dynamic simulations.

To sample the conformation space effectively in the conformational analysis of nucleic acids using NMR data, a genetic algorithm (GA) has been used. GAs are global optimization algorithms based on Darwin's natural selection principles. By this algorithm, a population of trial solutions (i.e. molecular
conformations), which are often referred to as chromosomes or strings, is optimized by invoking repeatedly some dedicated evolutionary operators (selective reproduction, crossover and mutation) during a number of iterations (in the GA terminology referred to as generations) (Goldberg, 1989; Holland, 1992). The framework, of the conformational analysis application we developed is based on earlier studies (Levy et al., 1991; Lucasius et al., 1991; Blommers et al., 1992). The trial conformations are evaluated using an appropriate objective function, which expresses the quality of the individual trial structures. In this study, objective functions are developed which incorporate specific criteria, in order to satisfy experimentally obtained proton–proton distance constraints. Alternatively, molecular conformations can be optimized to fit an experimental two-dimensional NMR spectrum. The fitting procedure, however, includes a time-consuming singular value decomposition. As a second alternative, an objective function based on the molecular force field of a structure can be used (Brodmeier & Pretsch, 1993; Judson et al., 1993; McGarrah & Judson, 1993; Hermann, 1994).

As has been stated previously, the conformational search space can be very large and complex. Inevitably, even when powerful computers are available, a thorough exploration of the conformational space will be very time consuming. Hence, it would be worthwhile to perform this extensive exploration by means of parallel computing. Given the fact that each string in a GA population can be evaluated independently of the others, provides the clue to parallelize the sequential GA application, yielding a drastic reduction of the required computing time. One way to achieve this is to divide the population of trial solutions into a number of smaller subpopulations. This subdivision now enables the evaluation of each subpopulation by instances of the GA on different computers (Levy et al., 1991; Gordon et al., 1992; Takanashi & Sano, 1992; Dorigo & Maniezzo, 1992). To deal efficiently with the complexity of the search space, explorations can be started from different points in the search space, which is achieved by initializing the subpopulations by different random seed values. Inclusion of a migration operator makes it possible for promising trial solutions to migrate from one subpopulation to other subpopulations. It is demonstrated that migration improves the sampling behavior of the whole system of parallel GAs.

A strategy for the conformational analysis of nucleic acids using parallel GAs will be outlined. The aim of this paper is to provide an evaluation of a method, to effectively parallelize a GA using the so-called migration or island model. The HYDRA programming environment, described in Part I of this series, was used to implement and execute the parallel application in a local area computer network. Inclusion of the migration model implies that the parallel processing had to be performed in a synchronous operation mode (Melssen et al., 1996). In order to test the parallel GA application, several data sets were simulated and evaluated.

2. BACKGROUND

2.1. Nucleic acids

Nucleic acids are sequences of specific building blocks called nucleotides. Deoxyribose nucleic acids (DNA) are probably the most outstanding nucleic acid species as they carry the genetic information. Nucleotides contain a phosphate backbone, a furanose ring or sugar moiety and a base. The four bases found in DNA are the pyrimidines cytosine (C) and thymine (T) and the purines guanine (G) and adenine (A) (Saenger, 1984). A schematic view of a DNA nucleotide is given in Fig. 1. In this figure, the torsion angles that describe the nucleotide are depicted as well (IUPAC/IUB, 1983). In general, DNA in cells has a double-helix structure. The conformation of DNA double-helices is known to a reasonable extent. However, uncommon species such as hairpin-loops and damaged strands provide a large field of unrevealed conformations. Hence, from a scientific point, it is a challenge to try to elucidate their structure which can be seen from the numerous publications on the problem domain. To test our application we concentrated on a hairpin of which the tertiary structure is already elucidated.

2.2. Nuclear Overhauser effect

Although there are several NMR techniques that can provide useful data for conformational analysis of nucleic acids, nuclear Overhauser effect spectroscopy (NOESY) is among the most important ones. In NOESY experiments a pulse sequence is given that correlates protons which are in close proximity. Protons that are less than 3 Å apart can

Fig. 1. A single DNA nucleotide with the torsion angles α, β, γ, δ and ε determining the backbone, the endocyclic torsion angles γ, defining the furanose ring (v = δ) and χ determining the orientation of the base to the furanose ring. The endocyclic torsion angles are related by the pseudorotation parameters P and v (Altona & Sundaralingam, 1992; de Leeuw et al., 1980). Protons that are attached to carbon atoms are not shown.
exchange magnetization via dipolar relaxation. The magnetization takes place during the mixing time \( t_m \) of the pulse sequence in the experiment. The magnetization transfer between interacting protons can be observed as a cross-peak in a two-dimensional NOESY spectrum. For short mixing times, the cross-peak intensity builds up linearly with the mixing time. This is the result of a direct magnetization transfer between adjacent spins (i.e. protons in case of NOESY). For longer mixing times, the magnetization of the individual spins can be described by a set of differential equations which are given by (Macura & Ernst, 1980):

\[
\frac{d}{dt} A = -RA.
\]

where \( A \) represents the \( N \) by \( N \) matrix with NOE intensities, i.e. the dipolar magnetization transfer between proton pairs. \( N \) is the number of protons in the trial structure and \( R \) is an \( N \) by \( N \) relaxation matrix. The off-diagonal elements are the relaxation rates between proton \( i \) and \( j \) which are called \( \sigma_{ij} \). The diagonal elements are the proton auto-relaxation rates \( \sigma_{ii} \) or \( \rho_i \) (Solomon, 1955):

\[
\rho_i = K \sum_{j, j \neq i} \frac{6}{1 + 4 \omega^2 \tau_i^2} \left[ 1 - \frac{1 + \omega^2 \tau_i^2}{1 + 4 \omega^2 \tau_i^2} \right],
\]

\[
\sigma_{ii} = K \left( \tau_i^{-3} - 1 \right).
\]

where \( K = 0.1 y^4 h^2 \), in which \( y \) is a gyromagnetic constant and \( h \) denotes Planck’s constant. \( \tau_i \) is the rotational correlation time for a single molecule which can be estimated using the Stokes–Einstein relation (Cantor & Schimmel, 1980), and \( \omega \) represents the Larmor frequency of the protons. The distance \( r_{ij} \) between the protons \( i \) and \( j \) in a trial structure can be calculated from their Cartesian coordinates.

Equations (2) and (3) are valid for small sphere-like molecules which behave like rigid rotors, i.e. rigid molecules undergoing isotropic tumbling with a correlation time \( \tau_c \). In these types of calculations \( \tau_c \) is estimated once for the molecule and then used in all calculations. The formal solution of equation (1) is given by:

\[
A = e^{-Rt_m}.
\]

The exact way to calculate NOE intensities is equivalent to an eigenvalue problem and equation (1) is then solved as:

\[
A = X e^{-\Lambda t_m} X^{-1},
\]

where \( \Lambda \) represents the eigenvalue matrix (eigenvalues \( \lambda \) on the diagonal) and \( X \) represents the matrix with eigenvectors of \( R \) after diagonalization of \( R \) (thus, \( X^{-1} R X = \Lambda \)) (Keppers & James, 1984).

As can be deduced, this involves a singular value decomposition which requires a lot of computing time (e.g. a dinucleotide has 20 protons that can give rise to an NOE intensity; consequently, a 20 by 20 relaxation matrix, \( R \), has to be diagonalized).

2.3. Genetic algorithms

A GA is a large-scale optimization algorithm based on population genetics and Darwin's "survival of the fittest" principle. A general flowchart of a GA is depicted in Fig. 2. The problem parameters are encoded on a (binary) bitstring. A set of bitstrings forms a population. The first stage in GA optimization is the initialization of the population, i.e. each bit in a bitstring is given randomly the value "0" or "1". The next stage entails the evaluation of the bitstrings; the bitstrings are assigned a quality value, in general referred to as fitness, according to a problem-dependent objective function. Then the population is subjected to a series of evolutionary operators. In the selection stage, bitstrings are selected according to some selection criterion which is usually proportional to their fitness, i.e. strings with a high fitness have the highest probability to produce offspring in a new population. This is called the exploitative or selective reproduction stage. With a probability \( p_m \), pairs of bitstrings from the new population are then chosen to undergo crossover. The crossover operator exchanges sequences of bits from a pair of bitstrings. Crossover ensures that relevant information is preserved. Finally, with a probability \( p_m \), single bitstrings are subjected to a mutation operator which inverts the value of single bits (in the case of binary encoding) on a bitstring to introduce new information. Crossover and mutation form the explorative stage. Now, the new population replaces the old population which completes one full cycle or generation. The GA is terminated when an arbitrary termination criterion is fulfilled for example, a maximum number of generations.

2.4. Parallel genetic algorithms

Having chosen a proper representation model and objective function for conformational analysis with a GA, the next step is to decide how to parallelize the GA. Implementing a parallel GA basically comes down to partitioning a population of strings into smaller subpopulations. After the partitioning step, there are essentially three ways in which a parallel execution of a GA can be achieved. In the first one, at least one operator is executed independently of the others. The simplest way is to perform string evaluation of the subpopulations in parallel and perform crossover and mutation sequentially. In more elaborate implementations crossover and mutation can also be performed in parallel. It has to be stressed that selection must always be done sequentially as any two strings in the ensemble of subpopulations can be
paired. This model for a parallel GA, in which at least one operator is executed independently of the others, is called the standard model or partitioned GA.

A second model by which a parallel GA can be created also divides the population into subpopulations, but invokes a sequential GA on each subpopulation. However, after a predefined period, promising strings in each subpopulation are selected and exchanged between the subpopulations. That is, each subpopulation sends copies of its best strings to its neighbors. These copies replace the worst strings in these subpopulations in order to get, on average, a higher quality of the genetic material. This exchange of strings is called migration. Where do migrants go to? There are many ways to transfer migrants, ranging from passing single strings to a neighboring subpopulation to transferring (ensembles of) ranked strings. All evolutionary operators are in this case executed independently of each other except for the migration operator. This operator is controlled by two parameters: the migration interval, which defines the number of generations between each migration, and the migration rate, defining the proportion of strings selected from each subpopulation for migration. This model is called the migration or island model or distributed GA. It provides a coarse grained division of the GA.

A third model that can be used to parallelize a GA is called the diffusion or neighboring model. In this case each string of a single population is placed in a cell of a planar grid and the division of the GA is hence fine grained. Selection and crossover are applied only between neighboring individuals on the grid according to a predefined neighborhood classification. The neighborhood of an individual is a set of potential partners. Strings are only allowed to migrate to geographically nearby subpopulations. The neighborhood can therefore be regarded as a subpopulation. The implementation of this model is mainly limited to expensive multi-processor based computer architectures.

2.5. Objective functions

No exact proton-proton distances can be derived from an NOE spectrum. Due to experimental error (peak-overlap and the spin diffusion effect), only ranges for the proton–proton pairs involved in an NOE intensity can be defined. An objective function that calculates to which extent these distance ranges are satisfied by a trial conformation can be used. Obviously, an ensemble of structures can satisfy proton–proton distance ranges, depending on the width of the ranges and the number of ranges. Hence, structures that satisfy these ranges can be used as starting structures for further refinement. By means of equations (2)–(5) theoretical NOE spectra for trial structures can be calculated. An objective function can be defined that calculates a Root Mean Square Difference (RMSD) between experimental NOE intensities and the corresponding calculated intensities (Lucasius et al., 1991):

\[
\text{RMSD} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left( \frac{a_{\text{calc}} - a_{\text{exp}}}{a_{\text{exp}}} \right)^2},
\]

in which \( n \) is the number of experimentally available NOE intensities, \( a_{\text{exp}} \) the experimental NOE intensity,
and $\Delta_{\text{calc}}$ the calculated NOE intensity. Finally a molecular force field for trial conformations can be invoked which can be used as an alternative objective function. Optionally, a cascade objective function can be used. For example, the NOE intensity objective function or force field objective function is exclusively triggered when a calculated structure satisfies the experimental distance constraints. In this paper, results with the NOE intensity objective function will be given. A paper based on results derived with the distance constraints objective function and force field objective function is in preparation.

3. IMPLEMENTATION

Torsion angle representations* and objective functions described above were implemented in prototype GAs for parameter estimation from the toolbox GATES (Lucasius & Kateman, 1993, 1994a,b). This was done according to the principles described by Lucasius et al. (1991) for the conformational analysis program DENISE† (Lucasius et al., 1991; Blommers et al., 1992). For parallel applications GATES was extended with a subpopulation creation function, a migration operator, and some basic I/O procedures for parallel processing (the extended toolbox is called PARGATES).

The parallel application was embedded in the HYDRA programming environment which is described in Part I of this series (Melssen et al., 1996). In the conformational analysis of nucleic acids with a GA, the evaluation of strings is the most time-consuming step. Hence, parallelization can in this case be achieved without specialized parallel hardware such as expensive multi-processor systems (diffusion model) that make memory access and communications fast. Using HYDRA, a local area network of workstations suffices to execute a parallel implementation of the GA. At the Laboratory of Analytical Chemistry in Nijmegen the network consists of a number of SUN Sparc™ workstations. HYDRA can be used to implement the standard model GA as well as the island model based GA. There are two ways in which PARGATES can deal with (creating) subpopulations (Figs 3 and 4).

* Conversion from torsion angle representation to Cartesian coordinates and vice versa was taken from Hendrickson (1961).
† DNA Evolutionary NOE Interpretation for Structure Elucidation.

- A large population can be generated which then is divided into several smaller subpopulations. Hence, one only needs a single random seed to create these subpopulations. With this approach, a large population is created for each processor in the initialization stage. Then this population is divided into $P$ subpopulations (which, of course, requires that for an optimal parallel execution at least $P$ different workstations are available) and are stored in separate disk files. In the HYDRA read stage, processor 1 reads subpopulation $p_1$, processor 2 reads subpopulation $p_2$, etc. Each processor now has its own set of strings to manipulate. This occurs in the HYDRA calc stage during which a regular sequential GA is applied.

Fig. 3. Island model parallel GA. In the upper diagram three subpopulations are created from a set of randomly initiated individuals. Genetic operators (recombination, crossover and mutation) are executed on the subpopulations. At regular time intervals migration takes place. This is illustrated in the lower diagram for a ring topology and a migration rate of 1.
An alternative approach is to initiate several subpopulations for each processor, hence taking a different random seed for each processor in the initialization stage of HYDRA. The next stages are the same as described above.

The type of subpopulation creation can be selected by the user. Migration of individuals, in the HYDRA implementation, is conducted during the read and write stages of the parallel executing application instances. When the migration operator is called, selected migrants of a subpopulation are written to file. Next a selected individual in a subpopulation is replaced by a migrant which is read from file. This means that apart from the usual GA configuration parameters (crossover and mutation probability, etc.), the user has also to define the extra migration parameters. Hence, the migration rate and migration interval can be chosen. In this study the worst performing string of one subpopulation is replaced by the best performing string of another subpopulation. Other schemes, such as random replacement of strings, are also possible. In addition, the migration topology or where the strings migrate to must also be specified. In this paper, a ring topology is used, implying that selected migrants of subpopulation \( P_1 \) move to subpopulation \( P_2 \), selected migrants of subpopulation \( P_2 \) move to subpopulation \( P_3 \), and selected migrants of the last subpopulation in the ring move to subpopulation \( P_1 \) (see Fig. 3).

4. RESULTS AND DISCUSSION

With the use of equations (2)-(5) and torsion angle values based on earlier studies by Blommers et al. (1991) we simulated NOE spectra of three structures, namely TT, TTTA and A-TTTA-T molecules (Blommers et al., 1991).* Corresponding to Blommers et al. (1991) we used an NMR frequency of 600 MHz, a mixing time of 200 ms and a rotational correlation time for the molecules of 2 ns. Sequential and parallel GAS were run with the NOE intensity objective function. In principle, all torsion angles can make 360° turns. However, NMR experiments show that they are restricted to certain ranges (staggered conformations gauche plus, 0°–120°; and gauche minus, 240°–360°; and the trans conformation, 120°–240°). The torsion angle ranges on the GA strings are therefore restricted to the experimentally allowed ranges.

Both GAS were run until convergence was reached. The GA configuration parameters for the runs are given in Table 1.† Each subpopulation (10 strings for the TT structure and 25 strings for the other structures) for the parallel GA was initialized at random. Then the subpopulations were distributed and executed on eight processors.

Table 2 shows that both the sequential and the parallel GA converged for the three simulated datasets.

Structures found by both algorithms were almost identical to the simulated structures. Table 3 gives an

*Experimental distance constraints for these molecules were available from literature. Generating structures that satisfy the available distance constraints was done with a sequential genetic algorithm. This took less than 100 generations.
†Lucasius & Kateman (1994a, b) give a detailed explanation of the configuration parameters in GATES.

<table>
<thead>
<tr>
<th>seq</th>
<th>par</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population size</td>
<td>100</td>
</tr>
<tr>
<td>Fitness scaling</td>
<td>linear dynamic</td>
</tr>
<tr>
<td>Selection mode</td>
<td>rank</td>
</tr>
<tr>
<td>Crossover mode</td>
<td>multipoint</td>
</tr>
<tr>
<td>Crossover prob.</td>
<td>0.80</td>
</tr>
<tr>
<td>Mutation mode</td>
<td>multipoint</td>
</tr>
<tr>
<td>Mutation prob. (%)</td>
<td>0.02</td>
</tr>
<tr>
<td>Migration topology</td>
<td>—</td>
</tr>
<tr>
<td>Migrant selection</td>
<td>—</td>
</tr>
<tr>
<td>Migration interval</td>
<td>—</td>
</tr>
<tr>
<td>Migration rate</td>
<td>—</td>
</tr>
</tbody>
</table>
Table 2. Results of the sequential (seq) and parallel (par) GA runs on the three structures

<table>
<thead>
<tr>
<th></th>
<th>TT</th>
<th>TTTA</th>
<th>A-TTTA-T</th>
</tr>
</thead>
<tbody>
<tr>
<td>seq</td>
<td>1500</td>
<td>7500</td>
<td>8500</td>
</tr>
<tr>
<td>par</td>
<td>2000</td>
<td>3000</td>
<td>8000</td>
</tr>
<tr>
<td>RMSD</td>
<td>≈0.01</td>
<td>≈0.05</td>
<td>≈0.07</td>
</tr>
<tr>
<td>Cpu time (s)</td>
<td>53.8</td>
<td>2233.8</td>
<td>8232.3</td>
</tr>
</tbody>
</table>

From Table 2 the total number of evaluations to reach convergence for both algorithms can be derived. It has to be stressed that the available processors for the parallel GA configuration are not

*Overview of the torsion angles of the TT simulated structure and the torsion angles for the optimal structures found by both algorithms.* Results for the TTTA and A-TTTA-T structures were similar to the TT results. Figure 5 shows that there is almost no difference between the optimal TTTA and A-TTTA-T structures found by both the sequential and the parallel GA and the simulated TTTA and A-TTTA-T structures.

*Fig. 5. Optimized TTTA structures after convergence of both the sequential and the parallel GA superimposed on the simulated TTTA structure (upper structure). Optimized A-TTTA-T structures after convergence of both the sequential and the parallel GA superimposed on the simulated A-TTTA-T structure (lower structure).*
homogeneous, i.e. slower processors have to compete with faster ones. Because the parallel GA in our application runs in a synchronous mode, the slower processors determine the overall evaluation time. The cpu time displayed in Table 2 is an average over the processors used. Because of the overhead, resulting from read/write procedures and memory swapping, the expected eightfold reduction in computation time for the parallel GA is not reached for all three structures. Moreover, the number of evaluations to converge are different for both algorithms. However, for the 11 and A-111A-1 structures, the parallel version is 6.1 and 4.4 times faster than the sequential version, respectively. Because the parallel GA converged much faster than the sequential GA in the case of TTTA this even resulted in a 9.2 factor reduction in computation. Hence, the HYDRA driven parallel GA outperforms the sequential GA for all three structures. It offers a promising method for relatively fast and profound sampling of the search space in conformational analysis of biomacromolecules.

5. CONCLUSIONS

The HYDRA programming environment provides a promising method to effectively parallelize a GA for conformational analysis. This is achieved by maximal exploitation of each processor in a local area network. It has been shown, for three simulated nucleic acid NOE spectra, that in this way a faster convergence process is achieved than with a sequential GA operating on one large population of trial solutions. This provides a challenge for handling even larger structures. However, it has to be stressed that the migration operator configuration was not optimized. Future research, therefore, includes the optimization of the migration operator configuration, in order to establish even a faster convergence of the algorithm. Moreover, the parallel implementation of the GA has to prove its feasibility on experimental NMR data. A further optimization can be achieved by developing other (combinations of or a cascade of) objective functions. Implementing other representation methods, and other or improved objective functions, is straightforward because the HYDRA driven parallel GA is built as a prototype algorithm. Dedicated extensions are easily implemented by the user. The method described therefore provides a simple tool for relatively fast and profound sampling of the search space in conformational analysis of biomacromolecules.

Acknowledgements—The authors wish to acknowledge Geert Kott and Kon Wehrens for their support during the development of the parallel conformational analysis application. This study was financially supported by the EC [project RENEGADE (ERBCHRXCT930419)]. E. P. A. Derks is supported by the Dutch Foundation for Chemical Research (SON) on behalf of the Dutch Organization for Scientific Research (NWO).

REFERENCES