HIPS, a hybrid self-adapting expert system for nuclear magnetic resonance spectrum interpretation using genetic algorithms

R. Wehrens, C. Lucasius, L. Buydens and G. Kateman

Laboratory of Analytical Chemistry, Catholic University of Nijmegen, Toernooiveld 1, 6525 ED Nijmegen (Netherlands)

(Received 3rd September 1992)

Abstract

An automatic system for the interpretation of two-dimensional NMR spectra of proteins, HIPS, is presented. Several artificial intelligence techniques are combined to form a flexible, hybrid system that has (limited) learning capabilities. Following the structure of the problem, the system is divided in modules with distinct functionalities. The first two modules are rule-based, and can be validated and refined semi-automatically using a set of already interpreted spectra. In this way, an optimized ruleset can be obtained to interpret unknown spectra. Results indicate a significant effect of training on performance. In the third module, a genetic algorithm is used to tackle a search problem of huge dimensions in which patterns found in the NMR spectra should be mapped to amino acids in the sequence.

Keywords: Nuclear magnetic resonance spectrometry; Expert systems; HIPS; Genetic algorithms; Hybrid expert systems

In recent years, much work has been done to make the determination of the three-dimensional structure of proteins possible. A technique that is most useful is two-dimensional nuclear magnetic resonance (NMR) spectrometry. A complete interpretation of the NMR spectra of a protein yields a set of distance constraints that can be used to calculate the conformation of the protein. The bottleneck, however, lies in the interpretation stage. This is a complicated process of trial and error that may take months or even years of expert’s time. Automation of the process is clearly desirable.

Few programs have been written that cover the complete interpretation of two-dimensional protein spectra (e.g., [1–3]); other programs (e.g., [4–6]) cover parts of the process. Most of these programs depend on human input during the problem solving stage to keep the interpretation process manageable. Since experts in most cases have access to more detailed information than the computer programs, for instance regarding peak shapes and spin patterns, they are able to prune the number of possibilities significantly. This makes the task easier for the computer program. However, it is difficult to compare different systems in literature. In many cases, different types of input spectra are used, or spectra that have been simulated in different ways.

The use of expert systems for spectrum interpretation has several advantages. One of the most important advantages is that the chemical knowledge is encoded explicitly in the system, in a form that is natural to an expert. In conventional pro-
grams the knowledge is typically hidden in the code, which can make it difficult to communicate with the program during the interpretation, and to adapt the program if conditions change. Expert systems are more flexible and therefore more useful in an iterative interpretation process, since the expert using the system can easily understand the individual pieces of knowledge and use them for his or her own purposes. There are, however, problems that cannot be handled well except by brute force. Large-scale search, for instance, is a problem solving paradigm that is not easily implemented in purely rule-based systems. Therefore, hybrid systems, combining classical expert system techniques with other paradigms like neural networks or genetic algorithms can often be very useful.

In this paper, we will describe the interpretation of two-dimensional NMR spectra of proteins by such a hybrid expert system. Starting from the peak positions in several types of NMR spectra (see below), HIPS (Heuristic Interpretation of Protein Spectra) combines adaptive expert system techniques and genetic algorithms to automatically produce an ordered list of spin patterns matching the sequence of amino acids in the protein. This means that all protons near the backbone of the protein have been assigned to resonance positions in the spectra. With this information distance constraints can be set up, from which the overall three-dimensional structure of the protein can be calculated. In the following section, we will discuss the benefits of hybrid systems and some of the techniques used will be described. After that, we will give a more detailed account of the NMR interpretation process. Finally, interpretation results of three test proteins will be given.

HYBRID EXPERT SYSTEMS

Hybrid expert systems are systems that do not rely solely on knowledge-based inference, or more popularly said, if–then rules, to find solutions to problems. Also other techniques are applied if appropriate. Combinations of expert systems with the pattern recognition capabilities of neural networks [7,8] are the most common, but also other techniques like genetic algorithms [9] can be very useful. The latter technique exploits the structure in a solution space to perform a very efficient search (vide infra). The combination of classical expert system techniques with other paradigms allows for a very natural representation, in which each problem solver is used for its own type of problems. This leads to better performance and more flexible systems. However, relatively few hybrid systems have been built so far because of the difficulties encountered in coupling two or more sophisticated problem solving paradigms.

In this paper, a system is described that combines a self-adapting expert system with a genetic algorithm. The former is used for synthesis and pattern recognition subproblems, whereas the latter is used to sample a huge subspace of possible solutions. In the next sections, the two techniques will be discussed in greater detail.

Self-adapting expert system techniques

In the past, many expert systems have been built that, once optimized and delivered, were completely rigid. In many cases, it was extremely difficult even for the knowledge engineers who built the system to adapt the system to fit new requirements, and the inflexibility of such systems decreased their level of acceptance. Apart from the inflexibility issue, optimization of rule sets and validation of prototype expert systems are difficult tasks in themselves, for which no general strategy has been accepted yet. One of the approaches addressing these issues is called the refinement strategy [10,11]. It assumes that the knowledge in the knowledge base is essentially correct and needs only minor changes to obtain optimal performance. Whereas this assumption is a demanding one, in many cases it will be at least partially fulfilled. A set of solved cases is used to identify weak spots in the problem solving process, and the rules that are responsible for the flaws in the reasoning are identified on the basis of some simple statistics. Then, appropriate actions are considered and ranked to their expected merits. In the original system, SEEK [10], the user was presented the ordered list of alternatives and could select the one that appeared best. In
this way, the proposed changes would be validated by the user (in most cases the expert) selecting them. The approach has been automated in the successor of SEEK, SEEK2 [11], where the best alternative was chosen automatically. More sophisticated techniques, such as an extensive meta-language, were used to rank the different refinements to their merits. Although the SEEK and SEEK2 programs are very specific, the ideas behind the refinement approach are quite simple and therefore easily applicable in other domains, such as the present one.

The advantages of the refinement strategy are clear: not only is the system validated against a dataset of known cases, but also the parts of the system that perform inferiorly are identified and corrected if possible. Even if the requirement that the knowledge in the knowledge base is essentially correct is not completely satisfied, the performance may be improved. Furthermore, the user will be able to see where the refinement strategy fails in such a case and therefore it is easier to find larger errors. An additional advantage is that the refinement module in the expert system can be built as a separate subsystem, validating the results of the real system. This also leaves open the possibility to disconnect the refinement module when the system has been optimized. Including such a semi-automatic refinement module in an expert system will also increase the flexibility of the system. If new conditions are met (for instance a better analysis method), the refinement module is connected to the system that in this way can be trained with examples from the new situation. New optimal settings can be found without extensive reprogramming. The refinements apply only to rule-based (sub)systems. If more knowledge representations are used, other techniques should be used to obtain the same flexibility.

In the present application, refinements are limited to a number of cases in which a set of ordered alternatives is set up on the basis of supports that have been collected. As well the number of supports needed for acceptance as the relative importance of the individual supports could be refined to obtain the best set of alternatives. By restricting the refinements to these parameters, it proved possible to build a fast and strongly focussed refinement module that can be coupled to an existing expert system that should be refined, and decoupled from the system without leaving any further traces. In most cases the generality in the original SEEK systems is not needed, and the refinements can be limited to a small set of operators.

The refinements provide an easy way to adapt the system to external parameters. If, for instance, it is known that the protein of interest resembles the structure of another, already interpreted protein, then the latter can be used to tailor the expert system settings to the particular problem. One can even think of the possibility to train the expert system on partial results of the protein of interest. Another situation in which the flexibility of the system is useful, is the case in which not all spectra have been recorded. For instance, not all spectra are crucial for the interpretation and some may therefore be omitted. However, this influences the validity of the rules. New settings may be found then by training on other proteins without using these spectra.

**Genetic algorithms**

Genetic algorithms [9] form a class of problem solvers that are especially powerful in sampling a large solution space with many local optima. They usually start with a random population of candidate solutions, where each solution is represented as a string. Each string is evaluated according to a fitness criterion. The true solution is expected to have the highest fitness. In the next step, strings are allowed to reproduce themselves with a probability relative to their fitness. A crossover operator is used to combine parts of highly fit parent solution strings in order to obtain even better child strings. This way, the new population is largely derived from successful solutions. A small amount of random mutations prohibits the system to converge prematurely. The ultimate goal is to approach the ideal solution by combining those parts of successful trial solutions that contribute positively to the fitness.

The difficult part in using genetic algorithms lies in the fact that a convenient and efficient representation must be found, along with opera-
tors for crossover and mutation that match the representation. In numerical optimization problems, this is usually not so much of a problem, but in sequencing and subset selection problems, as in the present case, other operators than usual are necessary to obtain good results [12]. Furthermore, a fitness criterion must be defined that also matches this representation.

NUCLEAR MAGNETIC RESONANCE OF PROTEINS

In this section, the principles of NMR as applied in the structure determination of proteins will be reviewed briefly. More information can be found in the references and NMR literature.

In proton NMR, each peak in the spectrum can be assigned to a proton or an interaction between two protons. The position of each peak is determined by the chemical environment of the proton(s) related to it. Because the number of protons in proteins is very large, the spectra are very complicated. One-dimensional spectra can not be interpreted for this reason. Therefore, two-dimensional spectra are used and sometimes even spectra with more dimensions. However, recording three- and higher-dimensional spectra of sufficient resolution may take days of expensive machine time, and therefore it is worthwhile to try to interpret less complicated spectra that contain less information. In a two-dimensional spectrum the ordinary one-dimensional spectrum is more or less present on the diagonal, in principle showing a signal for each proton in the chemical structure. The off-diagonal crosspeaks give, dependent on the kind of spectrum, additional information. An example of the kind of spectra that is used is the NOESY (Nuclear Overhauser Enhancement Spectroscopy) spectrum [13]; in this kind of spectrum a crosspeak appears between the resonance positions of two protons if they are less than 5 Å apart. From a complete interpretation of a NOESY spectrum, where each proton is mapped to a peak position on the diagonal and each off-diagonal crosspeak indicates a through-space interaction of two protons, a set of distance constraints can be calculated. These can be used to construct a three-dimensional model of the protein, e.g., using the distance constraints algorithm [14]. The difficult part is to map the peak positions in the spectrum to protons in the chemical structure. For this, several types of spectra are used that give complementary information. The key to the interpretation is the fact that protons that have more or less the same chemical environment, give rise to peaks that are close together in the spectrum. For instance, most protons in the backbone of the protein that are attached to the nitrogen lie in an area between 6 and 10 ppm. In general, the expert may use information from peak positions, but also information about peak shapes and peak intensities to interpret the spectra.

Interpretation of protein spectra

The common approach is the “sequential assignment” strategy [13]. It consists of three stages. In the first stage, sets of peaks are identified that belong to one single amino acid in the protein. Such a set of peaks will hereafter be denoted as “pattern”. It consists of a number of diagonal peaks, representing the protons in the amino acid, and a set of off-diagonal peaks, representing the interactions between these protons. As an example, two amino acids and their patterns are depicted in Fig. 1. Normally, only protons up to the γ position are included in the pattern, since their peak positions provide enough information to determine the structure of the protein backbone. The result of the pattern search step consists of a number of patterns that eventually should be mapped to separate amino acids in the sequence.

Each pattern has distinct characteristics, depending on the type of amino acid it belongs to. Among these characteristics are for instance the resonance position of the $C_\beta$ protons, the presence or absence of $C_\beta$ or $C_\gamma$ protons, and many others. In the second step of the spectrum interpretation, each pattern found in the first part is classified as an amino acid, sometimes a specific type like glycine, sometimes a group of amino acids, like aromatic amino acids. After this step, for each pattern a preliminary classification or a set of possible classifications is obtained.
In the third and last step, data from NOESY spectra are used, together with the results of the previous assignment part, to establish a list of all patterns that are possibly neighbours in the sequence. From this list, a sequence of patterns (or possibly more than one valid sequence) is obtained that maps onto the sequence of amino acids. At this point, all peaks present in the patterns are explicitly assigned to a proton in the protein. The assignment of the other protons is now relatively straightforward because of the large number of crosspeaks that have been assigned already. The construction of a valid sequence of patterns, however, is a large problem. To place seventy patterns (a medium-sized protein) in a specific sequence, there are $70! \approx 10^{100}$ possibilities. In most cases, more patterns will be found than the number of amino acids, so the problem becomes even bigger. This results in a scenario where only the obvious patterns are interpreted in the first place (for instance, there is only one plausible pattern for the one tryptophan residue in the sequence), and where these assignments are used to further refine other assignments. This process continues iteratively until a satisfactory solution has been obtained.

IMPLEMENTATION AND RESULTS

In this section, the results of the expert system on the interpretation of three proteins will be discussed. For each of the three steps in the spectrum interpretation, pattern search, pattern assignment, and sequential pattern assignment, a separate module has been implemented (see Fig. 2). At each point in the interpretation, results can be inspected, saved, altered, and results of earlier sessions may be read. It should be noted that the system only uses peak positions with an uncertainty of 0.01 ppm, whereas an expert can also use information on peak shapes and intensities. Incorporating results of manual assignments will improve the performance of the system significantly. However, all results reported here are obtained without manual help, unless indicated otherwise.

The spectra of the proteins that have been used were obtained from the Department of Biophysical Chemistry at the University of Nijmegen.
TABLE 1

Characteristics of the spectra of the three test proteins [The second column indicates the number of amino acids for which startpeaks are given in the literature (see text). The third column contains the number of peaks in the bottom half of the seven spectra for each protein]

<table>
<thead>
<tr>
<th>Protein</th>
<th>Amino acids</th>
<th>Startpeak patterns</th>
<th>Peaks</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-L30</td>
<td>58</td>
<td>52</td>
<td>388</td>
</tr>
<tr>
<td>BPTI</td>
<td>58</td>
<td>52</td>
<td>953</td>
</tr>
<tr>
<td>Tendamistat</td>
<td>74</td>
<td>71</td>
<td>2868</td>
</tr>
</tbody>
</table>

They have been used as test spectra for the program PROSPECT [3] earlier. Seven types of spectra are used: NOESY, COSY and RCT spectra, recorded both in H$_2$O and D$_2$O, and DQSY spectra [13]. The spectra of two of the three proteins, BPTI (bovine pancreatic trypsin inhibitor) and Tendamistat, were constructed by taking the peak positions published in the literature [15,16] and simulating the NOESY spectra from the (known) three-dimensional structure. This way, the simulated spectra can be said to be very much like real spectra. Peaks may be missing, overlapping or unexpectedly be present in the same way as is found in experimental data. The third protein, E-L30, was measured at the laboratory of Biophysical Chemistry at the University of Nijmegen [17,18] and the DQSY spectra have been added manually. Some characteristics of the proteins and their spectra have been gathered in Table 1.

Patternsearch

In the first module of the expert system, a list of so-called start peaks is set up. These off-diagonal peaks lie in the fingerprint region in the spectrum where peaks are expected that are caused by interactions between the C$_\alpha$ and the amide protons. Each startpeak is considered to be part of a separate pattern. At least as many startpeaks as amino acids are expected. From each startpeak, probable C$_\beta$ positions are found using a number of criteria. If, for example, crosspeaks are found between an amide, C$_\alpha$, and C$_\beta$ resonance positions in as well NOESY as COSY spectra, then all three probably belong to the same amino acid. Several other crosspeaks can increase the faith in such a conclusion. It is obvious that not all crosspeaks are equally reliable or important, and therefore they are divided in three classes: minor, medium and major importance. Combinations of supports are handled explicitly in the rules. The same strategy is used in the search for C$_\gamma$ peaks starting from the C$_\beta$ positions found earlier.

The importance of each of the supporting crosspeaks is used to identify patterns in the spectra of the three test proteins. These are compared with the real patterns that should be found. A refinement module similar to the one originally used in SEEK [10] is used to fine-tune the importances of the individual crosspeaks and the amount of supporting evidence needed for a peak to be accepted as, for instance, a C$_\beta$ peak. As an example, consider the situation where a peak is incorrectly accepted for the C$_\beta$ position because of two supporting crosspeaks, N-$\beta$ and $\alpha$-2$\beta$. Then the following refinements may help to correct the error:

(i) diminishing the importance of the N-$\beta$ crosspeak;
(ii) diminishing the importance of the $\alpha$-2$\beta$ crosspeak; and
(iii) raising the amount of evidence needed for acceptance above the total support for the incorrect peak.

Eventually, the refinements that will correct the largest number of errors are selected and presented to the user, who then can choose what refinement should be applied. This refinement process has been elaborated upon previously [19]. Since then, small changes have increased the performance to the level indicated in Table 2. The three proteins all have slightly different optimal settings, and a compromise has to be found to obtain the globally optimal settings. It ap-

---

* Exceptions are the amino acids proline, which does not have an amide proton, and glycine, which has two. In the former case, dummy patterns containing no peak positions are used, that can be filled in after manual interpretation; in the latter case, both startpeaks are taken into account, as well as the combination of startpeaks. Each glycine thus results in three separate patterns.
TABLE 2
Patternsearch results for the three proteins
(The third, fourth and fifth columns indicate the number of
the patterns found correctly using the initial, optimal, and
global settings, respectively)

<table>
<thead>
<tr>
<th>Protein</th>
<th>Patterns</th>
<th>OK (init.)</th>
<th>OK (opt.)</th>
<th>OK (global)</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-L30</td>
<td>52</td>
<td>41</td>
<td>43</td>
<td>43</td>
</tr>
<tr>
<td>BPTI</td>
<td>52</td>
<td>43</td>
<td>50</td>
<td>40</td>
</tr>
<tr>
<td>Tendamistat</td>
<td>71</td>
<td>40</td>
<td>65</td>
<td>65</td>
</tr>
<tr>
<td>Total</td>
<td>175</td>
<td>124</td>
<td>158</td>
<td>148</td>
</tr>
</tbody>
</table>

Table 2: Patternsearch results for the three proteins. The third, fourth and fifth columns indicate the number of patterns found correctly using the initial, optimal, and global settings, respectively.

Peared that the results of E-L30 were not very sensitive to the refinements that were applied; some peaks were missing from the spectra, and some patterns overlapped in such a way that the system could not conclude something else that it already did. This was different in the proteins BPTI and Tendamistat. There, and especially in the search for C_γ positions, the results were extremely sensitive to the refinements. The optimal settings for both proteins were rather different: BPTI required less strict acceptance criteria than Tendamistat. This is clearly visible in the large number of errors in BPTI in the globally optimal settings whereas the optimal BPTI limits yield only two errors. However, errors in the C_γ positions are less important than errors in the C_β positions, and it was felt that the global performance level was satisfactory.

Assignments of patterns to types of amino acids
In the second module of HIPS, the patterns that have been found in the first part are assigned to types of amino acids. For this, characteristics of the patterns are used. These characteristics include the presence or absence of certain peaks, and the resonance positions of the peaks. If, for instance, a C_α and a C_β share crosspeaks in the aromatic region, then it is probable that the pattern belongs to one of the aromatic amino acids (phenylalanine, tryptophan, tyrosine and histidine). These crosspeaks then are considered in greater detail to be able to distinguish between the separate aromatic amino acids. Furthermore, specific regions in the spectra give information about the peaks that are in them, such as the aromatic region mentioned previously [20]. Thus, by looking at the positions of the peaks in a pattern one is able to predict the amino acid type, or at least a set of probable types. This is implemented in rules, where the presence or absence of each characteristic contributes for an overall certainty factor for a type of amino acid. A low certainty factor of a pattern for a specific amino acid type indicates that there are no or few indicators that the pattern belongs to that type; a very high certainty factor indicates that the pattern is most probably of that type. Altogether, approximately one hundred contributions to certainty factors are possible. This large number makes it impractical to use classes of supports like minor, medium and major, as in the patternsearch module. Instead, numbers are used. No explicit boundaries are given, but initially all contributions were in the interval between −30 and +80. They are combined by summing. After all contributions have been gathered, the best patterns are selected for each type of amino acid. If for instance five amino acids of type threonine are present in the sequence, then the five patterns that have the highest certainty factor for threonine are selected, and all others that have the same certainty factor for threonine. This way, the chances are very small so that a pattern belonging to for instance a threonine is classified otherwise.

However, patterns may be classified in more than one class of amino acids. It is very important that the correct amino acid is among the possibilities, because otherwise it is very difficult in the last module, the sequential assignment, to provide a correct sequence of patterns. In this case, so-called false negative (FN) errors, where the correct answer is not in the list of possible solutions, are far more serious than false positive (FP) errors, where more patterns, including the correct ones, are possible for a specific type of amino acids. The first aim of the refinement module is therefore to minimize the number of false negative errors of the ruleset. However, a large number of false positive errors will lead in the sequential assignment to many solutions that seem equally probable. A second objective is
therefore to maximize the discriminating abilities of the expert system by minimizing the number of false positive errors. Between these two objectives, an equilibrium should be found.

In Table 3 the results of refining the contributions to the certainty factors are given in a similar way as in the first module, the patternsearch. The results are based on the patterns that were found using the globally optimal settings of the patternsearch module. In the case of BPTI, for example, twelve patterns were incorrect in the patternsearch part. As will be clear, in many cases this does not prevent the assignment module to correctly classify most of them. In the second column, the number of patterns is given for which the correct amino acid type was concluded using the initial settings (which were obtained from an expert). Refinements, in which the number of incorrect assignments was minimized, yielded settings that gave the results in the fourth column in Table 3. In all cases a very good performance was achieved. The third and fifth columns contain the numbers of false positive errors of the initial and optimal settings, respectively. As well regarding the number of false negative errors as regarding the false positive errors, remarkable improvements can be achieved. From the optimal settings of the three proteins, a set of globally optimal settings was derived. The results obtained with these settings are also included in the table. In approximately fifty percent of the cases, the patterns that were assigned incorrectly to an amino acid were already incorrect in the patternsearch module. However, as can be seen in the BPTI results, an incorrect pattern can be classified correctly, especially if errors are at the $C_\gamma$ position.

**Sequential assignment of patterns**

After the patterns have been assigned to types of amino acids, only one step is needed to have a complete interpretation of the NMR spectra for the backbone, $C_\beta$ and $C_\gamma$ protons. This step comprises a specific mapping of separate patterns to amino acids in the known sequence. For this, the information derived in the second module is used to ensure that each pattern matches the amino acid at its position in the sequence. In addition, peaks connecting neighbouring patterns that can be found in the NOESY spectra are used to determine whether two patterns may be neighbours in the chain. This way, for each pair of amino acids that occurs in the sequence, a look-up table of possible pattern pairs can be set up. Two criteria for a valid pair of patterns are applied:

(i) both patterns should match the types of the amino acids in the pair; and

(ii) Both patterns should be connected by crosspeaks in the NOESY spectrum. These connections can be crosspeaks between the amide-position of the second pattern and the amide, $C_\alpha$ or $C_\beta$ positions of the first pattern.

Several pruning procedures, controlled by flags, can be used to limit the number of possibilities:

(i) All patterns that are used to construct a glycine pattern with two startpeaks and themselves only contain one startpeak are excluded from the sequential analysis. These patterns are

---

**TABLE 3**

Assignment results for the three proteins
(All results have been obtained with the global settings from the patternsearch module. OK indicates the number of correctly interpreted cases, FP indicates the number of false positive cases)

<table>
<thead>
<tr>
<th>Protein</th>
<th>OK (init.)</th>
<th>FP</th>
<th>OK (opt.)</th>
<th>FP</th>
<th>OK (global)</th>
<th>FP</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-L30</td>
<td>32/52</td>
<td>147</td>
<td>45/52</td>
<td>191</td>
<td>41/52</td>
<td>191</td>
</tr>
<tr>
<td>BPTI</td>
<td>46/52</td>
<td>185</td>
<td>49/52</td>
<td>182</td>
<td>47/52</td>
<td>217</td>
</tr>
<tr>
<td>Tendamistat</td>
<td>58/71</td>
<td>208</td>
<td>61/71</td>
<td>133</td>
<td>62/71</td>
<td>259</td>
</tr>
<tr>
<td>Total</td>
<td>136/175</td>
<td>540</td>
<td>155/175</td>
<td>506</td>
<td>159/175</td>
<td>667</td>
</tr>
</tbody>
</table>
frequently assigned to a glycine pattern, but except in the case of extreme overlap, this is incorrect. This step is actually already performed in the second module, since otherwise too few glycine patterns may be selected.

(ii) All combinations of patterns that can not be matched with each other are excluded. If, for instance, for a combination of amino acids AA-1 AA-2 only the combination PATTERN-1 PATTERN-2 is possible, all pattern combinations for any other amino acid combination that begins with AA-2 are forced to start with PATTERN-2. Any other pattern combinations are deleted. If AA-2 occurs more than once, this is of course taken into account.

(iii) All patterns that have not been assigned to any amino acid type will be excluded from the sequential assignment step. This, in general, will not cause any correct patterns to be removed. In both the cases of BPTI and Tendamistat, two patterns were discarded that should have been included in the sequence. In all four cases, however, the patterns were misinterpreted in the pattern search module, and only their amide and Cα positions were correct. Their places in the sequence are taken by other patterns.

This way, the number of possibilities is reduced as much as possible. In Table 4 the number of different amino acid pairs and their associated number of pattern pairs is given for the three proteins, after pruning. For each protein the optimal settings (in both the pattern search and the assignment modules) as well as the global settings are used. Although the E-L30 protein is of the same size as BPTI, the number of possible pattern pairs is significantly larger in the latter case if global settings are used. This is a logical consequence of the composition of the proteins: some amino acids are more easily recognized than others, and therefore give rise to fewer possibilities in the pattern combinations. Tendamistat, again, poses the largest demands on the system.

The solution space for this problem is huge. For example, the number of solutions is 97!/(97 - 74)! ≈ 10^{130} for Tendamistat using the global settings. Some approaches break down the problem in smaller subproblems [3] by only tackling part of the amino acid sequence at once, but the resulting overall sequence does not have to be the best one in that case. Also, such approaches rely heavily on human intervention. For instance, in the case of the semi-automatic assignment of BPTI with the program PROSPECT [3], 80% of the patterns was unambiguously assigned to one type of amino acid. In our automatic assignment module, not more than 20% of all patterns, including glycines and prolines was assigned to only one type of amino acids, to avoid incorrect assignments. Other programs require the number of input patterns to be equal to the number of positions in the sequence [4]. Furthermore, in our case we deal with realistic NOESY simulations that may yield a large number of sequential connections between patterns. In other programs, the number of sequential connections is much smaller, either because smaller proteins are used [2] or because less NOESY peaks are included in the input files [1].

As already said, a genetic algorithm is a search technique for large solution spaces with many local optima, and therefore can be used in an

<table>
<thead>
<tr>
<th>Protein</th>
<th>Amino acids</th>
<th>Amino acid pairs</th>
<th>Patterns</th>
<th>Pattern pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-L30 (opt.)</td>
<td>58</td>
<td>52</td>
<td>61</td>
<td>1032</td>
</tr>
<tr>
<td>E-L30 (glob.)</td>
<td>58</td>
<td>52</td>
<td>61</td>
<td>922</td>
</tr>
<tr>
<td>BPTI (opt.)</td>
<td>58</td>
<td>54</td>
<td>63</td>
<td>1032</td>
</tr>
<tr>
<td>BPTI (glob.)</td>
<td>58</td>
<td>54</td>
<td>64</td>
<td>1567</td>
</tr>
<tr>
<td>BPTI (manual)</td>
<td>58</td>
<td>54</td>
<td>63</td>
<td>352</td>
</tr>
<tr>
<td>Tendamistat (opt.)</td>
<td>74</td>
<td>64</td>
<td>90</td>
<td>2244</td>
</tr>
<tr>
<td>Tendamistat (glob.)</td>
<td>74</td>
<td>64</td>
<td>97</td>
<td>4042</td>
</tr>
</tbody>
</table>
automatic spectrum interpretation program. The initial population of solutions is constructed by making random permutation strings of patterns with the length of the sequence of amino acid combinations. The evaluation function is very simple: the pattern combinations that satisfy the two criteria concerning sequential crosspeaks and pattern assignments are counted. Thus, the highest fitness that can be obtained is $N - 1$, where $N$ is the number of amino acids in the sequence. In practice, this will not be achieved, since errors in the earlier interpretation stages may cause a "correct" pattern combination to be absent in the look-up table. However, if the previous assignment modules are working correctly, then the fitness of the true solution should be close to the maximally obtainable fitness. Other deviations are possible because of missing sequential connectivities, and patterns that have not been identified because their startpeaks are missing from the fingerprint area. In the latter case, the place of such a pattern will be taken by another.

In Fig. 3 the fitnesses of a typical genetic algorithm run are plotted on the y-axis. The algorithm converges very fast to a point where the fitness stabilizes somewhat and then by chance finds a permutation that allows for further improvement. As can be seen in the second column of Table 5, the fitnesses of the true solutions of the three proteins lie below the maximally obtainable fitness $N - 1$. The difference is most distinctly present in the case of E-L30, which also proved to be the most difficult case in the previous interpretation stages (see Tables 2 and 3). Remarkable is also the difference in the fitnesses of the true solutions in the case of BPTI, where the optimal settings of the assignment and patternsearch parts clearly provide a better result for the fitness of the true sequence.

In a real-world situation, however, one does not have the true solution, and the highest fitness obtained in a number of runs may be taken as the fitness of the "true" solution. However, given the stochastic nature of the genetic algorithm, it is better to combine results of a number of runs to form a global solution. In Table 5, results of five runs for each protein, both with global and optimal settings in the previous interpretation parts, are gathered. In each case, a population of 500 candidate solutions was used. For E-L30 and BPTI, 3000 generations, and in the case of Tendamistat, 5000 generations were taken. In the cases of BPTI and E-L30, fitnesses as high as the fitness of the correct solution were found in almost all cases within 200 generations. For Tendami, not more than 1000 generations were needed in general. The fitness obtained with the

<table>
<thead>
<tr>
<th>Protein</th>
<th>True F</th>
<th>Mean F</th>
<th>Assignments</th>
<th>Correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-L30 (opt.)</td>
<td>29</td>
<td>33.8</td>
<td>40</td>
<td>15</td>
</tr>
<tr>
<td>E-L30 (glob.)</td>
<td>27</td>
<td>34.2</td>
<td>37</td>
<td>17</td>
</tr>
<tr>
<td>BPTI (opt.)</td>
<td>42</td>
<td>40.4</td>
<td>46</td>
<td>30</td>
</tr>
<tr>
<td>BPTI (glob.)</td>
<td>37</td>
<td>43.8</td>
<td>40</td>
<td>18</td>
</tr>
<tr>
<td>BPTI (manual)</td>
<td>42</td>
<td>42.8</td>
<td>53</td>
<td>41</td>
</tr>
<tr>
<td>Tendamistat (opt.)</td>
<td>54</td>
<td>61.0</td>
<td>58</td>
<td>17</td>
</tr>
<tr>
<td>Tendamistat (glob.)</td>
<td>55</td>
<td>62.4</td>
<td>47</td>
<td>12</td>
</tr>
</tbody>
</table>

Fig. 3. Performance plot of a genetic algorithm on a BPTI dataset obtained with optimal settings. As the number of generations grows, the fitness increases to a level where no improvement can be found.
global settings are in general higher than the fitnesses obtained with the optimal settings for each protein. This is as expected since more possibilities exist to construct a valid sequence in the case of the global settings (see Table 4).

A pattern is assigned to a position if it is at least in two runs present in the same position. This way, 66–88% of the positions were assigned patterns. In all cases, the amino acid on the assigned position matched the assignment of the pattern. However, only 26–65% of the assignments was in agreement with the "correct" assignments. It must be remembered, however, that this is not the result of the genetic algorithm failing to find the global optimum, but rather the genetic algorithm finding solutions that are in better agreement with its input data than the "true" solution. This is also the reason why the fitnesses found by the genetic algorithm are generally higher than the fitness of the "true" solution. Striking is the fact that in protein E-I.30 more patterns are assigned correctly using the global settings, than are assigned correctly with the optimal settings. This may be a consequence of the small number of runs that is done with the genetic algorithm, and in which a pattern is already assigned to a position if it occurs twice.

To investigate the ability of the genetic algorithm to produce valid results starting from a good look-up table, the dataset provided by the global settings in the case of BPTI was pruned manually, so that each combination of amino acids contained maximally ten combinations of patterns. This way, the total amount of pattern combinations was diminished to 352, one-third of the original amount. The results of the runs with this input set have also been gathered in Table 5. In this case, over 77% of all assignments was in agreement with the true sequence.

The above results prove the usefulness of genetic algorithms to solve the problems encountered in the sequential assignment of protein spectra. However, the complete automatic assignment is not yet feasible, and manual input would greatly improve the results of the system. This is clear from the results with the manually pruned BPTI dataset. However, as already said, in real life the expert can at any point intervene and focus the interpretation results. Therefore, the pruned dataset constitutes a reasonable simulation of reality.

HARDWARE AND SOFTWARE

The expert system is written in KEE (Intelli-corp Inc.), version 4.0. The six knowledge bases take approximately 150 K. Some 200 K of compiled LISP code is used. For patterns search and assignments, not more than 5 min real-time is required for Tendamistat on a SUN SPARC-1 workstation. BPTI and E-I.30 are done more than twice as fast. The genetic algorithm is written using the GATES toolbox [21] and runs on the same platform. One run of 3000 generations takes roughly four hours (also real-time).

Conclusions

In this paper we have described a hybrid expert system for the interpretation of NMR spectra of proteins. Although spectrum interpretation is an iterative process, in which a partial assignment can lead to a more complete assignment, it is important to extract as much information as possible from the spectra automatically, so that the expert can concentrate on the really complicated parts. The results indeed show that the trial-and-error process that constitutes the spectrum interpretation can not be discarded by the use of such systems, but can be significantly enhanced by the combination of heuristics and powerful search methods. Using a global set of settings, 47 patterns were assigned correctly in the three proteins without any human intervention, more than a quarter of all patterns. If for each protein an optimized set of settings is used, the number increases to 62 patterns, more than 35%. This illustrates the power of the approach. In real life, however, performance will be probably much higher since additional information from the spectra can be used by a human expert to solve ambiguities. A simulation of that situation on the protein BPTI yielded a performance of 77%. Therefore, we believe that these systems will form a great help in the spectrum interpretation.
The above results can be further improved in a variety of ways, each with its own advantages or disadvantages. Information on peak shapes and intensities can be taken into account, NOESY spectra with different mixing times can be used to distinguish between long-range and short-range NOEs, extra rules can be added to assign protons beyond the Cγ-positions and thus facilitate the assignment to amino acid classes, and more runs with the genetic algorithm can be started simultaneously. Furthermore, a feedback loop that incorporates sequential information in the earlier stages will significantly improve the results of the system. Also, it is important to train the expert system on more proteins than the three that have been used so far, and preferably train the system on experimental spectra of high quality. This way, optimal results can be obtained in the future.

REFERENCES