Automatic Structure Determination of Organic Molecules: Principle and Implementation of the LSD Program

NUZILLARD, Jean-Marc*

Pharmacognosy Laboratory, UMR CNRS 6013, University of Reims, Moulin de la Housse, BP 1039, 51687 REIMS Cedex 2, France

The LSD (Logic for Structure Determination) program generates organic molecular structures from 1D and 2D NMR data without resorting to chemical shift databases. Its use in the resolution of natural product structure determination problems has been already reported in the literature. This paper describes how data and structures are internally represented and processed by LSD to build solution structures.

Keywords structure elucidation, nuclear magnetic resonance, correlation spectroscopy

Introduction

Automatic structure elucidation emerged about forty years ago as an interdisciplinary field based on both computer science and spectroscopy advances. Most of the software systems that were developed in this framework are isomer generators that use spectroscopy to propose structure fragments and to validate the proposed complete structures.[1] Submitting an organic molecule to various optical spectroscopic methods (in which NMR is included) provides series of characteristic transition energies (expressed in cm^{-1}, nm or ppm, depending on the technique). A particular energy value is generally compatible with the presence of more than a single fragment. The known relationships between energy value and substructure are stored in spectroscopic data or knowledge bases.

Most of the published applications of computer-assisted structure elucidation programs deal with the resolution of natural product chemistry problems. This originates from the wide diversity of natural product structures and from the usual lack of molecular structure knowledge after the isolation and purification steps. Conversely, synthetic organic reactions only rarely cause deep structure changes between the known starting material and the final product, thus providing less challenging problems.

About twenty years ago, the analysis of natural products has gained in efficiency and reliability with the availability of the 2D NMR techniques. They have introduced the powerful concept of correlation spectroscopy.[2] The structural information brought by NMR chemical shifts was enriched with proximity relationships between molecular fragments. It was then possible, not only to identify chemical functions, but also to determine whether they are close or far within the studied molecules. Thus, the number of isomers that are compatible with a given gross formula is strongly reduced.

Softwares for automatic structure analysis were modified to incorporate 2D NMR data,[3-9] either for structure validation, or prospectively for structure generation. The LSD (Logic for Structure Determination) program[4] was the result of the first attempt to generate organic structures on the main basis of heteronuclear correlation spectroscopy. The present paper describes how data and structures are internally represented and processed by LSD to build solution structures. Recently, the source code of LSD was made publicly available (www.univ-reims.fr/LSD) so that the interested reader can access all implementation details.

Preparing data for LSD

Data from 1D NMR

LSD does not incorporate any chemical shift data base. Therefore, the user has to propose his own constraint from 1D NMR data. Bonds may directly be deduced from elementary shift analysis: a carbon atom resonating at 200 ppm can safely be bound to an oxygen atom. Moreover, the hybridization state of both atoms is sp^2 and the two other neighbors of the carbon atom are also carbon atoms. The atomic element, hybridization state (sp^3 or sp^2 only) and number of bonded hydrogen atoms form the so-called atom status that must be defined for all skeletal (meaning non-hydrogen) atoms. This constraint is not so strong than it seems to be for a practical application of LSD. Chemical shift value and proton multiplicity pattern constraint must be translated to atom property. An atom for which a property is defined has a known number of neighbors within a subset of the whole skeletal atom set. In the previous example, the carbon atom resonating at 200 ppm has exactly two neighbors in the carbon atom set.

In order to define properties, the skeletal atoms are

* E-mail: jm.nuzillard@univ-reims.fr

Received April 4, 2003; revised and accepted July 14, 2003.

referenced by an integer index. It is generally handy to rank carbon atom indexes in the order of the corresponding ^{13}C NMR chemical shifts.

Data from correlations

The LSD software is designed to produce organic molecular structures compatible with HMQC[10] (or HSQC[11]), COSY[12] and HMBC[13] data. Each correlation is transformed into an atom index pair. An hydrogen atom is preferentially labeled with the same index as the skeletal atom which it is directly bonded to. The relationship between atom references is thus established by means of the HMQC spectrum. Two or three hydrogen atoms that are bonded to the same skeletal atom get the same index.

A COSY correlation that originates from a ^3J ^1H-^1H coupling is translated to the pair of the corresponding hydrogen atom references. Two-bond correlations are easily discarded as the involved atoms are bonded to the same skeletal atom. Long-range correlations through more than three bonds are generally of weak intensity and must also be discarded. Their interpretation would lead to bind atoms that are not bonded and thus to generate an incoherence and the failure of the problem resolution.

A HMBC correlation that originates from a ^2J or a ^3J ^{13}C-^1H (or ^{15}N-^1H) coupling is translated to a pair containing the reference of the involved atoms. In practice, HMBC correlations of skeletal atoms with nearly identical chemical shifts rather frequently occur. LSD offers the possibility of defining a fuzzy correlation between a skeletal atom group and an hydrogen atom. In this case, at least one member of the group is responsible for the correlation existence. Again, longer range correlations lead to inconsistencies and to solution search failure. A detection scheme for such situations has been recently proposed.[9]

Data from other sources

The user can provide a substructure that must be present in all solution structures. It may originate from prior knowledge about the studied compound, like biogenesis for natural products. The substructure is not necessarily connected, meaning that the presence of two or more non-overlapping fragments can be checked. The substructure is a set of "sub-atoms" bonded by "sub-bonds". The sub-atoms are defined by their index, chemical element, multiplicity and hybridization state. Sub-atom status may include alternatives. The assignment of a sub-atom (the skeletal atom index it corresponds to) may also be provided.

The data file structure

All gathered information about the studied molecules is translated to LSD commands. Other commands control program execution. All of them are made of a four-letter code followed by arguments. The commands related to spectroscopy include atom status and property definition, the setting of initially known bonds, and HMQC, COSY and HMBC correlation description. Execution control commands concern:

* The output structure format. Each solution structure may be output either as a list of pairs of bonded atoms or in a format suitable for automatic structure drawing by means of the "outlsd" program.

* The printing of the stored input data. This possibility is offered essentially for data set debugging purpose.

* The effective resolution of the problem. This allows one only to check the validity of a data set with entering in the resolution process.

* The verbosity level. This option allows to print a track of the events occurring during resolution.

* The taking into account of the substructure.

* The possibility of running LSD in step-by-step mode. In this mode, the user may interact with the choices made by the program during resolution.

* The printing of resolution history. Each solution can be output along with the list of the choices made during the resolution and that led to it.

* The search for partial solutions (*vide infra*).

* The elimination of duplicated solution. Under some circumstances identical solutions may be produced and only one is kept for output. This filtering process may be disabled.

* The stopping of the resolution process at a given search depth. See section (*vide infra*).

Data representation in LSD

The LSD program is written in C language. All required data structures are statically allocated. Array dimensions are defined at compilation time and can be changed by the user because the source code is available. A skeletal atom is described by a data structure that includes status, bond, correlation, property, substructure, and equivalence (see below) descriptors. Two arrays are used to convert hydrogen atom references from and to skeletal atom references, as described by the HMQC related commands. Sub-atoms also have their own status, bond, and assignment descriptors. Before and during resolution, all non-bonded non-correlating skeletal atoms are grouped into equivalence classes derived from the status identity equivalence relationship. Only a single class member is available at a time for bond formation. This prevents to lose time to produce identical structures in which only "dummy" atom references are permuted. The atom lists that define atom property are stored as arrays of Boolean values. Other lists, for HMBC correlations of atom groups or for alternative sub-atom status definition, are stored as arrays of integers.

During the command file reading phase, each four-letter code and the following parameters are transformed into integer values and stored in an array of pre-processed commands. Any input data syntax error causes the immediate printing of a message and program termination. The

initial state of the structure resolution process is obtained by scanning through the pre-processed command array. Each command type is associated to a priority index, so that commands with the lowest priority are processed at the end. This means that commands are not processed in the order they are written in the data file. For example, HMBC commands are always processed after HMQC commands, whatever the writing order. At this stage, semantic error are detected and reported (like an attempt to bind an atom whose status is unknown).

A last-in first-out stack data structure (a "heap") is used for both data management and resolution history bookkeeping. As a remain of the initial coding of LSD in Prolog language, structure search involves self-calling (or recursive) procedures to achieve backtracking. This latter concept allows to handle ambiguous data by systematically exploring all possible hypothesis, in a so-called "depth-first" analysis. The execution of recursive procedures requires the duplication of their private data structures. In order to limit this time and memory consuming process, all data structures related to atoms, correlations, properties, equivalence classes, and sub-atoms are implemented as global variables. Thus, recursion is simulated through simple heap manipulation procedures. Each time some value must be changed and recovered after return from a recursive call, its original value and memory address are inserted on top of the heap before the call. On return, the address-value pair is extracted from the heap and used to restore the original value at its address.

The structure search algorithm

Generate and test

The main problem with constrained isomer generation is to avoid combinatorial explosion. One could imagine to first generate all possible isomers without any constraint, and then to retain only the structures that fit with spectroscopic measurements. Of course, this would be not realistic, due to the very high number of possible isomers (the explosion), even for a modest number of skeletal atoms. A better way would be to stop building an isomer when it violates at least one spectroscopic constraint. This means that early testing can greatly reduce computation speed, unless the time cost of the test process is prohibitive. An even better approach would use the same data to directly generate substructures. As already mentioned, a spectroscopic information may be compatible more than a single fragment. Its use for structure element generation requires to consider all hypothesis. The least ambiguous fragments must be used first in order to detect, as early as possible, an incoherence arising from the exploration of a wrong hypothesis. Thus, the efficiency in the generation process strongly depends on the order in which the data is exploited.

These simple considerations are well illustrated by the example detailed in Ref. 3. The authors analyze a molecule for which 22 HMBC correlations are observed. Each one can originate either from a ^2J or a ^3J coupling. This means that the initial generation of the 2^{22} data sets (more than 2 millions) must be considered if one wants to prospectively use the available HMBC data. This huge task is reduced to a manageable one in LSD, because hypothesis are formed all along the resolution process and not solely at the beginning.

Structure resolution in LSD takes place in four main steps. The first one deals with the exploitation of the pre-processed input file content. A COSY correlation through a ^3J coupling between proton H_1 and H_2 gives rise to the X_1—X_2 bond if skeletal atoms X_1 and X_2 are bonded to H_1 and H_2, respectively (see Fig. 1a). This deduction does not give rise to any ambiguity and is therefore carried out as early as possible in the resolution process. The second resolution step is the creation of bonds from HMBC correlations. Three indeterminacy levels are considered. The first one only exists for atom group correlations. Such a correlation between $\{X_1^1, X_1^2, \ldots, X_1^n\}$ and H_2 is successively considered as the hypothetic ones between X_1^1 and H_2, X_1^2 and H_2, $\cdots$, X_1^n and H_2. Then, any HMBC correlation between X_1 and H_2 (either true or hypothetical) is considered as a fictitious ^1J or ^2J X_1-X_2 correlation if X_2 is bonded to H_2 (see Fig. 1b). When the ^2J hypothesis is explored, the third indeterminacy level concerns the identity of atom X that is bonded to both X_1 and X_2. Clearly, the HMBC correlations involving atom groups must be considered after the other ones, due to the wider hypothesis set they lead to. The third resolution step is the systematic pairing of all atoms having less bonds than required by their status (the incomplete atoms), and takes place after all correlations are used. Complete structures (in which all

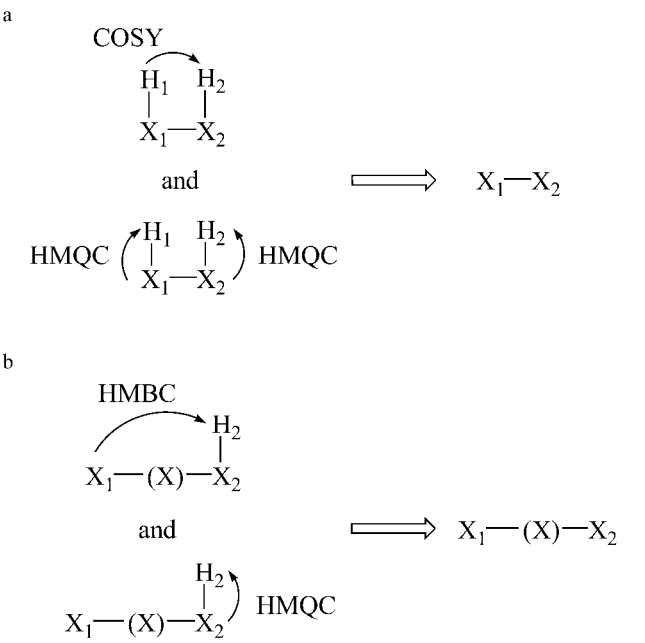

Fig. 1 Deduction of proximity relationships between skeletal atoms (a) from COSY correlations and (b) from HMBC correlations.

atoms are complete) are tested in the forth resolution step for double bond placing, Bredt's rule compliance, substructure presence and originality.

Steps 2, 3 and 4 involve recursive procedure calls, so that when some wrong hypothesis in step 2 leads to structure rejection in step 4, backtracking occurs until this hypothesis is reconsidered. Of course, it is generally impossible to know in advance which hypothesis has caused backtracking. One could imagine not to use the substructure information in test step 4, but prospectively for bond formation in step 2. In the same way, double bonds could be placed during steps 2 and 3, when bonds are formed. A preliminary work in these directions leads to a much more complex search algorithm with only modest performance improvement.

The normal execution scheme can be modified, either to produce incomplete solutions or for data set debugging in "maximum level" mode. When correlation data does not suffice to limit the number of generated isomers, execution flow may be altered so that only partial structures obtained after step 2 are written on output. It is then assumed that there is at least one way of deriving from it a complete structure that passes the final tests. This result is obtained by activation of the so-called "partial mode". If logical data inconsistency causes the failure of the resolution process, the number of consecutive recursive calls is printed. Then, execution flow can also be altered by a "maximum level" command having this number as argument. Execution stops when the indicated number of nested calls is reached and the obtained structure is printed. This feature may provide some hint about the reason of the failure.

Bonds from correlations

At this stage, unambiguous data from 1D and 2D COSY spectra is already taken into account. The treatment of each inherently ambiguous HMBC correlation is decomposed in two sub-steps: correlation choice and consequence analysis of the choice.

Correlation choice As already mentioned, resolution time strongly depends on the HMBC correlation choice order. Indeed, step 2 of the resolution process deals only with fictitious ^1J or ^2J correlations deduced from HMBC and HMQC data during step 1. In LSD, the highest priority is given to the correlations of the atoms that are complete and then to the closest to be complete. The search of such an atom can be carried out either within the set of all skeletal atoms or, preferentially, only within a smaller subset of it, named the "basis set". The latter contains the recently bonded atoms of the molecule that present unexploited correlations. This search scheme favors the construction of a connected structure piece and therefore the earliest detection of wrong hypothesis. Once a correlating atom is selected, one of its correlations is chosen, with a lower priority given to those with atom groups.

Correlation analysis A fictitious X_1-X_2 correlation is first analyzed as a X_1—X_2 bond, and then as a X_1-X-X_2 fragment. If either X_1 or X_2 is a complete atom, there is no need to build the X_1—X_2 bond because X_1 and X_2 are already neighbors. Indeed, when a Y_1-Y_2 correlation can be interpreted either from the initial set of bonds or from those constructed during resolution, it obtains the "already used" status. This avoids to analyze correlations from which no bond can be deduced and happens when Y_1 and Y_2 are either directly bonded or both bonded to the same atom. The building of the X_1-X-X_2 fragment is also simplified when either X_1 or X_2 is complete. Both cannot be simultaneously complete because this would mean that correlation X_1-X_2 was not previously stamped as useless, as it should have been. If X_1 (resp. X_2) is complete then X is necessarily a neighbor of X_1 (resp. X_2) and there is only one bond to establish between X and X_2 (resp. X_1). If neither X_1 nor X_2 is complete, X is chosen in the set of atoms that still can establish two bonds. Clearly, the number of possible X atoms is smaller when X_1 or X_2 is complete. This observation is at the origin of the correlation selection heuristic. If X has no correlation and no bond, then the equivalence class mechanism plays its role and allows only one atom per class to be involved in a bond. Bond formation is also constrained by atom properties. Atom status information is used to define the number of bonds which an atom can form, and to ensure that a complete sp^2 atom has at least a sp^2 neighbor. Once a bond is established, equivalence classes, atom property data, the basis set and correlation status are updated for the next selection-analysis cycle.

Bonds from incomplete atoms

The atoms left in an incomplete state after step 2 of the resolution process are systematically paired during step 3, until a complete structure is generated. The pairing process is recursive and involves two sub-steps. First, an incomplete atom is selected and is then tentatively bonded to all another ones according to status, property and equivalence constraint.

Final tests

A complete solution from step 3 must be rejected if it does not comply to the tests in step 4 of resolution.

Some structures do not allow double bonds to be placed between sp^2 atoms, even though each sp^2 atom has at least such a neighbor. Therefore, double bond placing may be seen as a filter for some unrealistic structures.[14] Double bonds involving sp^2 atoms with a single sp^2 neighbor are placed first because this task does not introduce any alternative. The other double bonds are placed using a recursive process until each sp^2 atom bears a double bond.

The second test also involves double bonds and eliminates structures that violate the Bredt's rule. This means that a double bond cannot be placed at a bridgehead of a small-sized bicyclic system. This non-energetic criterion was introduced to eliminate most of unrealistic, highly strained structures. The corresponding algorithm is de-

scribed in full detail in Ref. 15.

Sub-structure search in LSD is also achieved by a recursive process whose goal is the search of a relationship between atoms and sub-atoms that respects bond position and status compatibility. In step 1 of the resolution, the sub-bonds are re-ordered so that, when sequentially considered, the bonded sub-atoms redraw the substructure with minimum disconnection. During step 4, the existence of a SX_1-SX_2 sub-bond so that SX_1 is assigned to skeletal atom X_1 imposes SX_2 to be assigned to a neighbor X_2 of X_1. Of course the status of X_2 must be identical or compatible with the one of SX_2.

Finally, each structure is transformed to a vector of integers that contains the bond information in a compact and univocal format. Each time a new solution structure is found, its vector is formed and compared to those of the previously found solutions. In case of equality, the new solution is rejected, unless this test is disabled by the appropriate execution control command. If not, the vector of the original solution is stored for future comparison.

Once all tests are successfully passed, the structure is written to the output device (either computer screen or text file) in the required format.

Conclusion and perspectives

The reader of the present paper could expect here the treatment of an example. However, there will be none, because some of them were already proposed in previous papers[4,14,16-20]. I strongly recommend the reading of ref. 16 because it deals with the resolution of a truly complex problem. I also suggest the interested reader to consult the on-line documents in the LSD web site.

If a single improvement had to be brought to the LSD algorithm, it would certainly deal with the introduction of atoms with initially unknown status. A straightforward way of doing it would be to generate all status sets compatible with the gross formula and with the proposed alternative status. The execution of the search algorithm on them necessarily provides the desired solutions. This crude generate-and-test option can certainly be refined by defining unknown status when needed during resolution, instead of doing it for all atoms at the beginning of the search process. Automatic status definition must also benefit in some way from the knowledge of chemical shift values.

Ranking of structures in decreasing likelihood order is also an important feature for the structure elucidation program end-users. Again, chemical shift is certainly the key of the solution. The comparison of the experimental values with those provided by a chemical shift prediction tool produces a numerical value suitable for structure likelihood ranking. Of course, the important point is then the availability of an efficient and possibly low-cost prediction tool. Prediction is far beyond the scope of this article, but it seems to be obvious that its quality strongly depends on the size of spectroscopy database it relies on. Moreover, the three-dimensional nature of organic molecules makes it difficult to predict chemical shifts only on the basis of planar structure. As a partial conclusion on this topic, structure set ranking is still an open problem for the molecules that are not already registered in spectroscopic data bases.

The LSD program was not written in such a way that it can solve all structure problems, but it solves many problems that are likely to appear in a laboratory involved in plant secondary metabolites study. This explains why sp hybridized atoms were voluntarily left out, even though they can occur in natural products. Such a limitation, as well as the previously mentioned ones, will have to be corrected in future versions of LSD. Its initial design and implementation in Prolog on incredibly slow computers (according to actual standards) strongly favored the search of decent execution times at the expense of program application scope completeness. One can expect this situation to be corrected in a near future.

References

1 Jaspars, M. *Nat. Prod. Rep.* **1999**, *16*, 241.
2 Claridge, T. D. W. *High-resolution NMR Techniques in Organic Chemistry*, Pergamon, Amsterdam, **1999**.
3 Christie, B. D.; Munk, M. E. *J. Am. Chem. Soc.* **1991**, *113*, 3750.
4 Nuzillard, J. M.; Massiot, G. *Tetrahedron* **1991**, *47*, 3655.
5 Peng, C.; Yuan, S.; Zheng, C.; Hui, Y. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 805.
6 Funatsu, K.; Sasaki, S. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 190.
7 Steinbeck, C. *Angew. Chem., Int. Ed. Engl.* **1996**, *35*, 1984.
8 Lindel, T.; Junker, J.; Köck, M. *J. Mol. Model.* **1997**, *3*, 364.
9 Eliashberg, M. E.; Blinov, K. A.; Williams, A.; Martirosian, E. R.; Molodtsov, S. G. *J. Nat. Prod.* **2002**, *65*, 693.
10 Bax, A.; Subramanian, S. *J. Magn. Reson.* **1986**, *67*, 565.
11 Bodenhausen, G.; Ruben, D. *J. Chem. Phys. Lett.* **1988**, *69*, 185.
12 Aue, W. P.; Bartholdi, E.; Ernst, R. T. *J. Chem. Phys.* **1976**, *64*, 2229.
13 Bax, A.; Summers, M. F. *J. Am. Chem. Soc.* **1986**, *108*, 2093.
14 Nuzillard, J. M. *J. Chim. Phys.* **1998**, *95*, 169.
15 Nuzillard, J. M. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 723.
16 Ley, S. V.; Doherty, K.; Massiot, G.; Nuzillard, J. M. *Tetrahedron* **1994**, *50*, 12267.
17 Almanza, G.; Balderama, L.; Labbé, C.; Lavaud, C.; Massiot, G.; Nuzillard, J. M.; Connolly, J. D.; Farrugia, L. J.; Rycroft, D. S. *Tetrahedron* **1997**, *53*, 14719.
18 Nuzillard, J. M.; Connolly, J. D.; Delaude, C.; Richard, B.; Zèches-Hanrot, M.; Le Men-Olivier, L. *Tetrahedron* **1999**, *55*, 11511.
19 Mulholland, D.; Randrianarivelojosia, M.; Lavaud, C.; Nuzillard, J. M.; Schwikkard, S. L. *Phytochemistry* **2000**, *53*, 115.
20 Mulholland, D.; Schwikkard, S. L.; Sandor, P.; Nuzillard, J. M. *Phytochemistry* **2000**, *53*, 465.

(E0304045 PAN, B. F.; FAN, Y. Y.)