(31) Rose, S. L.; Jurs, P. C. "Computer-Assisted Studies of Structure Activity Relationships of N-Nitroso Compounds Using Pattern Recognition". J. Med. Chem. 1982, 25, 769-776.
(32) Jurs, P. C.; Ham, C. L.; Brugger, W. E. "Computer Assisted Studies of Chemical Structure and Olfactory Quality Using Pattern Recognition Techniques". ACS Symp. Ser. 1981, 148, 143-160.
(33) Norden, B.; Edlund, U.; Wold, S. "Carcinogenicity of Polycyclic Aromatic Hydrocarbons Studied by SIMCA Pattern Recognition". Acta Chem. Scand., Ser. B 1978, B32, 602-608.
(34) Wolff, M. E. "Steroids and Other Hormones". In "Quantitative Structure-Activity Relationships of Drugs"; Topliss, J. G., Ed.; Academic Press: New York, 1983.
(35) Bodor, N.; Harget, A. J.; Phillips, E. W. "Structure-Activity Relationships in the Antiinflammatory Steroids: A Pattern-Recognition Approach". J. Med. Chem. 1983, 26, 318-328.
(36) Stouch, T. R.; Jurs, P. C. "Monte Carlo Studies of the Classifications made by Nonparametric Linear Discriminant Functions". J. Chem. Inf. Comput. Sci. 1985, 25, 45-50.
(37) Hansch, C.; Leo, A. "Substituent Constants for Correlation Analysis in Chemistry and Biology"; Wiley: New York, 1979.
(38) Kier, L. B.; Hall, L. H. "Molecular Connectivity in Chemistry and Drug Research"; Academic Press: New York, 1976.
(39) Tou, J. T.; Gonzalez, R. C. "Pattern Recognition Principles"; Addison-Wesley: Reading, MA, 1974.
(40) Moriguchi, I.; Kopmatsu, K.; Matsushita, Y. "Adaptive Least-Squares Method Applied to Structure-Activity Correlation of Hypotensive N-Alkyl-N''-cyano-N'-pyridylguanidines". J. Med. Chem. 1980, 23, 20-26.
(41) Zander, G. S.; Stuper, A. J.; Jurs, P. C. "Nonparametric Feature Selection in Pattern Recognition Applied To Chemical Problems". Anal. Chem. 1975, 47, 1085-1093.
(42) Sporn, M. B.; Roberts, A. B.; Goodman, D. S., Eds. "The Retinoids"; Academic Press: New York, 1984.
(43) Newton, D. L.; Henderson, W. R.; Sporn, M. B. "Structure-Activity Relationships of Retinoids: Tracheal Organ Culture Assay of Activity of Retinoids"; Laboratory of Chemoprevention, Division of Cancer Cause and Prevention, National Cancer Institute: Bethesda, MD.
(44) Lindahl-Kiessling, K.; Bhatt, T. S.; Karlberg, I.; Coombs, M. M. "Frequency of Sister Chromatid Exchanges in Human Lymphocytes Cultivated with a Human Hepatoma Cell Line as an Indicator of the Carcinogenic Potency of Two Cyclopenta(a)phenanthrenes". Carcinogenesis 1984, 1, 11.
(45) Soper, K. A.; Stolley, P. D.; Galloway, S. M.; Smith, J. G.; Nichols, W. W.; Wolman, S. R. "Sister-Chromatid Exchange (SCE) Report on Control Subjects in a Study of Occupationally Exposed Workers". Mutat. Res. 1984, 129, 77.
(46) Latt, S. A.; et al. "Sister Chromatid Exchanges: A Report of the GENE-TOX Program". Mutat. Res. 1981, 87, 17.

Chemometrics and Distributed Software

D. L. MASSART* and P. K. HOPKE†
Farmaceutisch Instituut, Vrije Universiteit Brussel, Laarbeeklaan 103, B-1090 Brussels, Belgium

Received February 1, 1985

An extrapolation is made of trends in chemometrics and analytical chemistry. This leads to a picture of the analytical laboratory of 1990. The importance of software is underlined, and it is concluded that the present situation is not what it should be and that the entrance of publishers and/or learned societies could remedy that situation.

INTRODUCTION

The aims of chemometrics have been defined as follows:[1,2] "Chemometrics is the chemical discipline that uses mathematical, statistical and other methods employing formal logic (a) to design or select optimal measurement procedures and experiments, and (b) to provide maximum chemical information by analyzing chemical data."

If one eliminates words such as "mathematical", this definition could just as well be a definition of analytical chemistry itself. In fact, chemometrics from being a mere subdiscipline of analytical chemistry is evolving to be the basis of modern analytical chemistry to such a degree that many excellent analytical chemists are also excellent chemometricians, even if they do not recognize this fact. To show the all-pervading role of chemometrics in analytical chemistry, let us consider what the analytical laboratory of the future might be.

ANALYTICAL LABORATORY OF 1990

To extrapolate, one first must look at the present situation. So, let us look at the way analytical chemists proceed from the point where they have been given an analytical problem to the point where they deliver the information that they were seeking.

Figure 1 gives the main steps; namely, the development of a method, its execution, and the obtaining of information from the determination. This process is considered in some more detail in Figure 2. The very first problem facing the analytical chemist is to select a method. For example, let us suppose the analyst needs to determine a certain drug in blood. He will then need to decide first whether to use TLC, HPLC, or GLC. If, for instance, his choice is HPLC, he will then have to decide whether to try a reversed-phase or normal-phase column with UV, fluorometric, electrochemical, or post-column derivatization detection, etc. This process usually leads to the selection of an initial procedure, which is then optimized. In this step the analyst starts from initial parameter values (such as parameters describing the composition of a mobile phase, elution temperature, gradient, etc.) and by logical reasoning, which as we will show later can be formalized, arrives at the final optimal or, at least, acceptable values of the parameters.

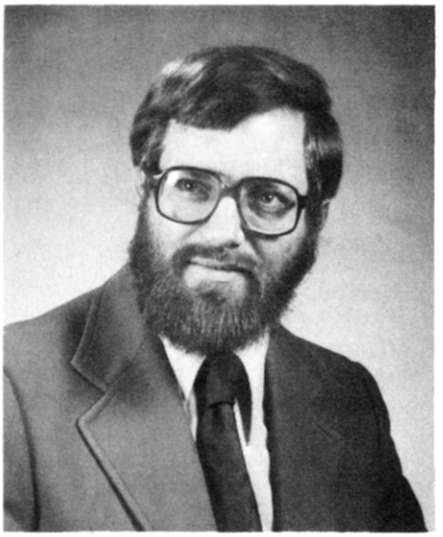
Main step 2 of Figure 1 also can be divided into two steps (Figure 2). Usually, one needs to carry out some pretreatment of the sample, such as weighing, drying, extracting, etc., that is then followed by the actual determination. The pretreatment step is usually the more difficult one (at least in pharmaceutical or biomedical analysis). It involves the most time and cost and very often determines the quality of the final result in terms of precision and accuracy.

The third main step of Figure 1 starts with the data acquisition (i.e., the collection of data by computer). This step is followed by the transformation of the signal to chemical information. This information consists of a list of chemical identities and concentrations. Sometimes, this is the end result, but much more often this chemical information has to be translated into what could be called diagnostic or user information. Does the result of a patient's blood test indicate illness; does the result of the analysis of an air sample signify that the air that was sampled is polluted and that a certain industry has contributed to this in a significant way, etc.?

Now, let us see how a chemometrician views and contributes to this process and at the same time what tomorrow's analytical laboratory might be like. The first step is probably the most

† On leave from the Institute for Environmental Studies, University of Illinois, Urbana, IL.

0095-2338/85/1625-0308$01.50/0



Philip K. Hopke is Professor of Environmental Chemistry at the University of Illinois at Urbana–Champaign. He obtained his B.S. in chemistry in 1965 from Trinity College of Hartford, CT, and his M.S. and Ph.D. in chemistry from Princton University in 1967 and 1969, respectively. After a postdoctoral appointment at M.I.T., he held an assistant professorship at the State University College at Fredonia, NY. In 1974 he moved to the University of Illinois at Urbana–Champaign. Professor Hopke has a broad interest in air pollution chemistry with particular emphasis on applying multivariate statistical methods to air-quality data and the study of radon and its decay products. Professor Hopke has published over 80 papers in the refereed journal literature, edited three books, and recently has had his first book, *Receptor Modeling in Environmental Chemistry*, published. He has spent the 1984–1985 academic year on a sabbatical leave at the Farmaceutisch Institute of the Vrije Universiteit Brussel.

difficult one. A few years ago, there was considerable interest in information theory to help analytical chemists select analytical systems, but this interest has abated. In general, it seems that this approach has not led to spectacular results. However, the recent entrance of expert systems into analytical chemistry might lead to a revival of interest. Until now, expert systems have been used mainly in the data treatment step,[3,4] but in our view, expert systems might also become very important in the selection and development of analytical procedures. There is one very successful analogy in organic chemistry, namely, computer-assisted organic synthesis,[5] which, after all, is also concerned with finding the most likely method. In fact, some preliminary applications of analytical method development are appearing now in the literature. A small article by researchers from Varian, contained in one of Dessy's excellent and often prophetic Interface series about applications in HPLC,[6] and a feasibility study about the use of expert systems in pharmaceutical analysis by UV spectrophotometry[7] seem to indicate that expert systems have a future in this field. There generally is a lot of interest and excitement about the possible use of expert systems for this kind of application. One of the problems with expert systems is that they work with a decision tree and that the number of possible alternatives to be considered may not become too large. For instance, it is not a good idea to include all 200+ available GLC phases in such a system. Information theory and some of its earlier results may become important in choosing the critical information subset. Information theory, supplemented with chemical insight, has shown, for instance, that only a limited number of GLC phases are really needed. Similarly, it has been shown that nearly all HPLC applications can be carried out with one single stationary phase and six solvents.[8] Much more has been done about the optimization step: it is one of

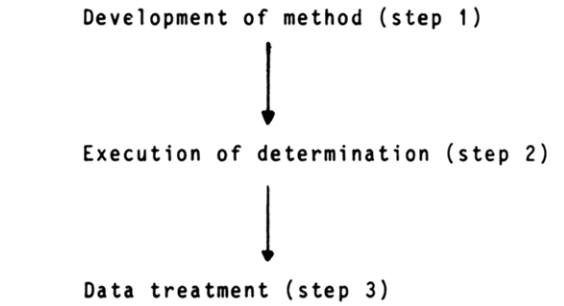


Figure 1. Main steps of the analytical process.

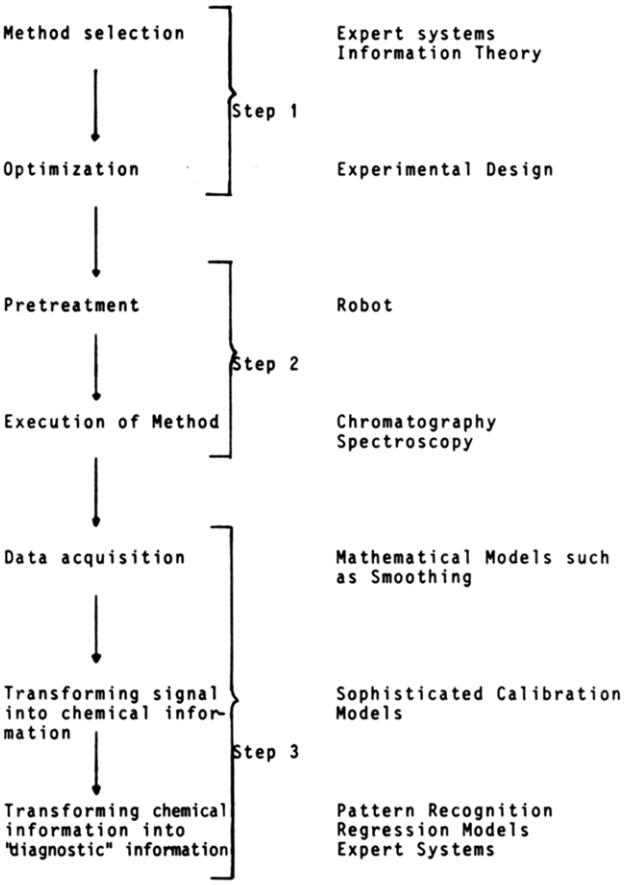


Figure 2. A more detailed view of the analytical process (left) and the techniques used by the chemometrician (right).

the techniques that has been studied most by the chemometrician, and the time appears to be ripe to bring the achievements of chemometrics in this field to a wider public. For this reason, we will discuss this subject to a somewhat greater extent in one of the following sections of this article. As we have already mentioned, the acquisition of the chemical data may really represent only the beginning of the process of gaining chemical insights into the system under study. With the trend toward multiple variable measurement methods, greater computer control of the analytical method and data acquisition, and the more complete automation of procedures through expert systems and robotics, the possibility exists that one may be overwhelmed with data unless efficient and effective chemometric methods are available to put those data into appropriate contexts of decision making. There has been a great deal of progress in the area of multivariate data analysis that basically parallels the improvements in the speed and size of available computers. This aspect of chemometrics will also be discussed in detail in a subsequent section.

In the pretreatment step a great advance has been made by the recent introduction of robots in the laboratory.[9,10] A robot is really a computer-controlled arm, at least as it is used in

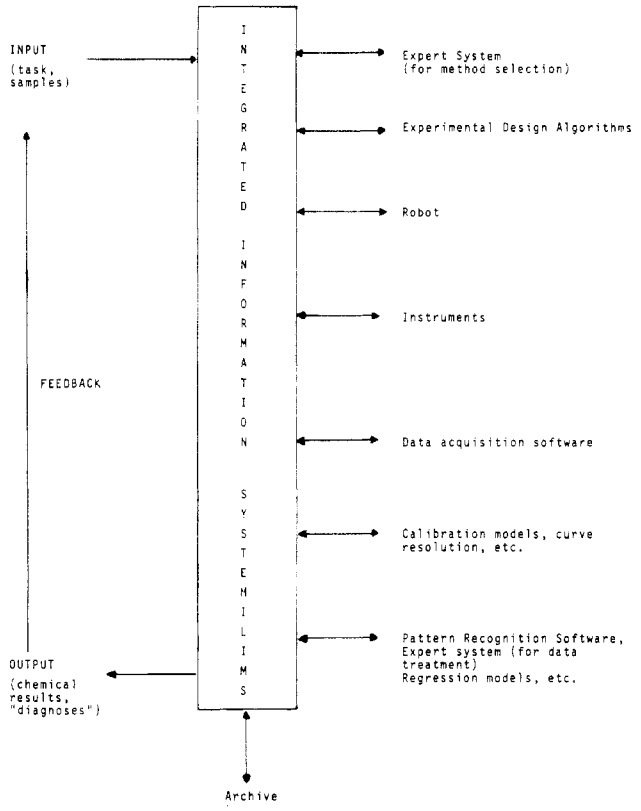


Figure 3. The "intelligent" laboratory concept.

the laboratory. Chemometricians have for a long time been hampered in applying their theories in practice by the need to leave their computers and go into the laboratory to do some dirty chemistry. Now comes their chance. More seriously, the robot is a natural extension of the software approach promoted by the chemometrician.

Figures 1 and 2 consider analytical chemistry as a process. This means that interaction with the environment and feedback have also to be considered. All this leads to the concept of the intelligent laboratory (Figure 3), a future laboratory that, given a certain problem, is able to select its own methods, carry them out, and analyze the data in such a way that useful information is presented to the user.

There is no doubt that software will play an extremely important role in this field. To emphasize this role and to review the current state of such software, we will discuss in greater detail the two fields that we have outlined in this introduction.

EXPERIMENTAL OPTIMIZATION

An often recurring problem in the development of analytical procedures is experimental optimization, the object being to maximize (or minimize) the response of the procedure in question. By response, one must then understand the quantity used to evaluate the method, i.e., the optimization criterion. In chromatography the criterion might be a quantity describing the separation quality such as the resolution or one of many others that have been described.[11] Alternatively, it could be the sensitivity of the procedure. The optimization consists in the selection of the values of parameters influencing the system such that an optimal response is obtained. An optimization strategy usually consists of three stages: the choice of the optimization criterion; the selection of the parameters to be considered; the actual optimization. There are two kinds of optimization programs available: dedicated optimization programs and general ones. In the dedicated programs, a specific solution has been found for a particular application. Examples in the field of chromatography are the overlapping

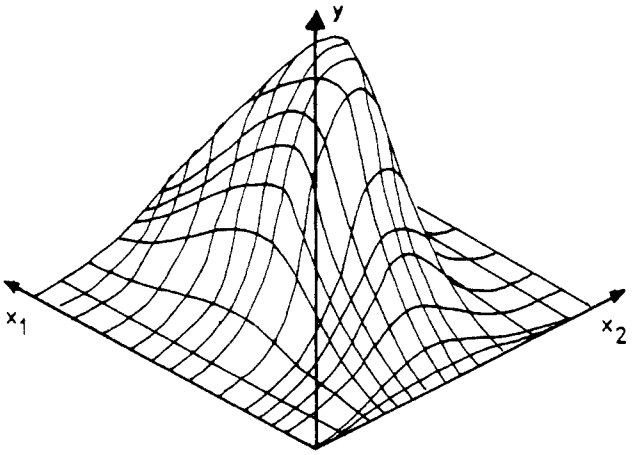


Figure 4. A response surface describing the optimization criterion y in function of the two parameters x_1 and x_2.

resolution maps (ORM) of Snyder[12] and the window diagrams of Laub.[13] These programs integrate the three stages outlined above in the sense that the criterion is fixed. For some parameters, this may also be the case (as in the ORM), and the optimization obtained can then only be applied to the particular situation studied.

In more general programs, usually only the third step is considered, and the user is asked to insert his own criteria and selection methods. The optimization itself can make use of two basic designs. The first design is often known under the name Simplex. It is a sequential design, meaning that, after each experiment, the parameter values are determined for the next one. The second approach is sometimes known as factorial experimental design and consists in the carrying out of a certain number of experiments according to a prearranged plan.

To understand the difference, it is necessary to examine the simple case of an optimization involving two parameters, x_1 and x_2. In that case, a response surface such as that in Figure 4 results. The optimization consists in finding the maximum of the resulting "hill". The philosophy of the Simplex method is to conduct a few experiments, drop the one with the worst results, and replace it with a new one in another (in the simplest case: opposite) direction. By doing this iteratively, one climbs the hill in an efficient way. When using the factorial design, one uses the results of the experiments to fit a second-order polynomial through the observed results to describe the surface. The more sophisticated designs usually have features of both kinds of designs. The supermodified Simplex,[14] for instance, is a Simplex method that climbs the hill of Figure 4 in a way determined by the local gradient and the existing knowledge of the shape of the surface. Alternatively, the steepest ascent[15] method uses small factorial designs in a sequential way.

A question that is often asked is which method is the best one. The answer is really that it depends upon circumstances since all optimization problems are different! A recent case study of our own experience illustrates some of the problems that may occur. The study concerned the optimization of the gas chromatographic determination of polychlorinated biphenyls (PCB's) after direct injection and was performed for the BCR (the European equivalent of the U.S. National Bureau of Standards). In most cases, experimental optimization in chromatography is best achieved with the Simplex methodology. However, the participants indicated that there were three variables to be taken into account and that there were not less than five criteria to be optimized: the peak height, width, symmetry, area, and resolution. This example, by the way, indicates one of the more difficult problems encountered when applying optimization methods. All of the methods are

constructed for the optimization of a single variable, but very often, there are several optimization criteria. Such a situation cannot be handled at all with a Simplex approach since a separate optimization procedure would be needed for each criterion. Factorial designs are more suitable. In one run information can be obtained that permits the computation of the response surface. Theoretically, this surface should then allow the user to find an optimum for each criterion and yield enough information to find a good compromise for the five criteria. However, one of the problems of the factorial design is that one must be well aware of the mathematical limitations. If the optimum is not within the boundaries of the experimental region and if parameters are included that are not relevant, absurd results are obtained. Simplex does not have these problems but, as explained before, could not be used here for other reasons.

The practical problems encountered with the software in this case study are a good example of what one must expect, if one wants to apply these (and most other) mathematically oriented methods in analytical chemistry. We would have preferred to use existing software. However, although the design we chose is well documented in the literature, we could not find immediately available software and had to write the program ourselves. Since such software certainly has been written for this problem by others, we will not be able to write a publication about our program. Thus, the postgraudate student who wrote it cannot be rewarded for her effort. The users for whom the optimization was calculated received a complimentary copy of our program, but to use it, they will need to study it carefully. Although the program works, it was not cleaned up completely and not made as user friendly as possible. In the field of optimization in general, few off-the-shelf programs exist. A notable exception is Simplex programs, of which several are now commercially available on a worldwide basis.[16,17] This clearly is an important step forward, but one really would need to have a number of software modules available to adequately cover most optimization problems, and one should be able to combine these modules at will. This situation certainly does not exist at this moment. If it did, it would be a solution to the problems of chemometricians. However, it would not suffice for those analysts that want to use the tools created by chemometricians without having to become specialists. One of the most frustrating problems is the lack of communication between the chemometric specialists and the analytical method user. It is not possible here to go into detail about the reasons for this lack of communication. Certainly, one way of helping to overcome this situation certainly is to have easily available and very user-friendly software available, so that it helps the user select the correct options.

DATA ANALYSIS

The other major aspect of chemometrics is the determination of the relationships that exist in a set of collected data. Major advances have been made in the spectroscopic and analytical instrumentation that can be used to obtain large data sets. These data can be used to characterize individual compounds or analytical samples. It is easy to collect enough data to make them incomprehensible without some computer-assisted data analysis method. Fortunately, at the same time that our ability to obtain data has so dramatically increased, there have been concurrent developments in computer systems and associated software that permit more complete data analyses and sophisticated interpretations of experimental results than were previously possible.

The two major types of data analyses that are performed are the identification of groups of "similar" objects (or variables) in the data set and the determination of the relationships among the variables (or objects) within a defined group. Once these relationships have been characterized, new objects can be classified as belonging or not belonging to any particular group, and the properties of new objects can be predicted based on the variable interrelationships for that class of objects. Thus, chemometric methods include unsupervised and supervised pattern-recognition procedures, eigenvectors and regression analysis, and multivariate analysis of variance.

There are many, many publications regarding the application of these methods to a wide variety of chemical, biomedical, and environmental problems. In contrast to the optimization area where there is very little available software that is widely distributed, in this area of chemometrics, there is a very large and bewildering assortment of software with widely varying availability, quality of documentation, support and maintenance, and ease of implementation. A major question arises as to how an individual can develop the collection of programs and the expertise in their use to perform the necessary data analysis tasks and to be certain that the mathematical methods are applied to problems appropriate to a particular technique.

A first step is typically to employ one of the several large packages of statistical programs such as the Biomedical Computer Program (BMDP), the Statistical Package for the Social Sciences (SPSS), and the Statistical Analysis System (SAS). In each of these cases, the programs have been developed over a period of time by a number of individuals with the primary intent of batch processing of nonchemical data. In many cases, the programs are designed with default parameter values that may not be appropriate for the interpretation of chemical data. For example, in the eigenvector analysis (principal component analysis) of a correlation matrix, it is a common default to only retain components with eigenvalues greater than 1. This criterion can be shown to represent a lower bound to the number of components needed to describe the system[18] but can also be shown to represent a simultaneous upper bound.[19] Thus, there are many instances where a reexamination of a factor or components analysis has shown that more valid information can be obtained from the data by retaining factors beyond those with eigenvalue greater than 1. Similar problems can arise in the use of multiple linear regression in these packages since there is little or no direct provision for the inclusion of diagnostic methods to identify outliers or collinear variables.

Because of some of the limitations of these packages for analysis of chemical data, a variety of special programs have been developed for particular kinds of data analysis. A system of data manipulation and analysis methods has been developed at the University of Washington under the direction of Professor Bruce Kowalski. This system, ARTHUR, has been successfully applied to the interpretation of many chemical data problems. A system of cluster-analysis programs originally developed for numerical taxonomy studies, CLUSTAN, has also been applied in a number of chemical studies.

All of these packages, SPSS, SAS, ARTHUR, and CLUSTAN, are distributed and maintained by private companies with fairly high levels of support and cost. BMDP is similar although still distributed by the University of California at Los Angeles. They are described in more detail with illustrations of their use by Wolff and Parsons.[20] Most reasonable-sized minicomputer or mainframe computer systems will have subscribed to one or more of these systems. Several of these systems are now being rewritten and distributed for some of the more powerful microcomputers. Other complete statistical packages have been written expressly for microcomputers so that almost any procedure that has been available on a large system is probably available on a microcomputer subject to core memory and execution time constraints. The more complex problem is that of the number, quality, and

accessibility of special-task programs. The current chaotic situation is that programs can be obtained through a variety of channels from public-access repositories, from commercial software vendors, and directly from the authors if one knows that the program, in fact, exists. For example, it may be of interest to have a program to perform Target Transformation Factor Analysis (TTFA), an eigenvector method developed by Professor Edmund Malinowski and co-workers.[21] They made their program, FACTANAL, available through the Quantum Chemistry Program Exchange (QCPE) at Indiana University. Anyone can obtain the program for a small fee to pay for the tape and copying charges. The program is delivered as is. There is some information on input formats and use, but it is the user's problem to implement it on his particular system. There is no support from the QCPE. User support is not their function. The programs obtained in this way were generally written for a particular computer to solve particular problems and may not be sufficiently flexible for other types of data. In the case of applying TTFA to airborne particulate data, it was found more convenient to write another program, FANTASIA,[22] that performs basically the same tasks but is directly applicable to particular kinds of data. This program is available directly from its authors, but also comes as is. For most of these kinds of programs, there is little incentive to make the programs transportable and user friendly since they are written by and for a research group where all of the users will be aware of its idiosyncracies.

There has emerged several journals that will publish descriptions and even listings of programs. For example, *Computers & Geosciences* recently published a constrained, least-squares regression program with some self-contained diagnostics.[23] The program was written in reasonably standard FORTRAN and is thus relatively easy to implement once you have typed in many lines of code and corrected your errors. However, it also requires the availability of several library routines from a particular library, the International Mathematics Subroutine Library (IMSL). Unless your computer has IMSL implemented on it or you have access to the IMSL documentation, it is difficult to find alternative, equivalent library routines. Thus, the utility of the publication has been diminished to the extent that the program is not completely available on computer-compatible media.

A final consideration in data analysis software is a problem that has arisen with the advent of microcomputer systems. This problem is compiled, special-purpose programs where the code is not available to the end user. A number of programs that have been developed by research groups for particular data analyses have taken this approach. It protects their programming concepts but can often leave the user bewildered when the program crashes with only a cryptic error message. It also means that he cannot take advantage of improvements in systems software like new, more efficient compilers or the ability to redimension data arrays to fit unusual needs. It can often be very frustrating to have bought a program to solve a problem and not to be able to fully test the method on your data because you cannot make small modifications to the code to make it fit your problem constraints.

Thus, the situation in data analysis is that there are many programs available if you can find out about them, if you can get them running on your system, and if your problem fits their constraints rather than the other way round.

WHY DISTRIBUTED SOFTWARE

In the previous sections, we explained why the present software situation is not at all what it should be. From the viewpoint of the programmer, it is annoying that he cannot always publish his product. The person who optimizes an analytical procedure can easily publish his work, but the programmer who optimizes a software program usually cannot. From the viewpoint of the user, it is regrettable that he cannot find easily implemented and accessible, user-friendly software. Except perhaps for standard statistical applications, he either has to write to a colleague, who probably will send him an incompletely tested and documented product with not all necessary options available, or write it himself. Therefore, there is no question that, if there was a way to publish good software, both the programmer and the user would be satisfied. The next questions then are how to guarantee that only good-quality software is published and what exactly is meant with "good quality".

The most important quality criterion is of course the scientific correctness of the program. There are two aspects to this. One is the validation of the software: how can one be sure that the program yields correct results? One possible answer to this is to consider software as a publication. All good scientific journals have reviewing procedures, and the more thorough they are in their review processes, the better the scientific reputation of the journal usually is. One publisher (Elsevier Scientific Software) has recognizied and implemented such a process. Before a program is published, it first asks an anonymous referee to give his opinion on the scientific quality and the relevance of the program submitted. If the advice of this expert is that the program is acceptable, then an official referee is appointed. He runs the programs on his own computer and is asked to prepare a detailed report about all aspects of the program. The program is then published only after the reviewer has finally agreed to it. To guarantee thorough reviewing and to reward the pains he has taken, the reviewer's name is mentioned on the title page of the manual. It is too early to known whether Elsevier's experiment will succeed, but we feel very strongly that the publication of reviewed and tested, in short, validated, software is essential. We are also convinced that there is a role for learned societies in this field. Isenhour's editorial[24] in this journal in which he states that the American Chemical Society should consider playing a role in this process indicates that we are not alone with this opinion. An additional advantage of subjecting programs to a reviewing procedure is that it meets one of the earlier discussed difficulties. If a refereeing procedure is followed and the program is published officially by a respected publishing house or a learned society, then there is no reason why one could not mention it on a curriculum vitae or in the list of publications added to one's yearly activity report.

A second aspect is that software should be open, i.e., that the user should be able to know exactly what the program does and that the source code should be made available. This very rarely is the case with programs that run with an instrument. Of course, it is understandable that instrument manufacturers do not want to make their know-how easily available to their competitors and that software houses selling software do not want copying to be made easy. Nevertheless, scientists should insist that they need to have all the documentation available to know exactly what a program does. Again, there might be a role here for learned societies. They could, on the one hand, try to instill in their members the notion that indiscriminate copying of scientific software is morally wrong and does not serve the interests of the scientific community and, on the other hand, explain to those people who produce scientific software that they have the obligation to disclose what exactly their program does.

A second important quality criterion is easy implementation of the program by the user. This means that he should be able to let the program run correctly on the first trial and without the help of a software or hardware expert. One of the biggest problems in this respect is compatibility. About 2 years ago, there seemed to be some hope that software would grow to

be more easily transportable by the introduction of the IBM PC. This has led to a generation of compatibles, and for a while, it seemed that this machine would become a de facto standard to which other manufacturers would have to adhere. However, at least one manufacturer (Apple with the Macintosh) has successfully challenged IBM's position and produced a machine that is not compatible at all. It seems now that the universally transportable software ideal is still far away. There is also another factor that is responsible for this phenomenon. Theoretically, by not using some statements and by adding a few subroutines (for instance to translate the CLEAR statement into HOME and vice versa according to the needs of a particular computer), it should be possible to produce a kind of midstream BASIC that could be used by all microcomputers. The Dutch broadcasting company NOS has developed such a program so that it can broadcast programs compatible with most computers. However, to produce user-friendly programs that incorporate aids such as a division of the screen such that questions are always asked in the same location and answers always appear in another location, one must use all the capabilities of a machine including the CALL, PEEK, and POKE instructions that are usually not transportable. Therefore, at this moment there seems to be only one way of ensuring easy transportability and that is to try out the product on several machines and to add this information in the manual.

Similar problems exist in other computer languages. For example, there are several microcomputer implementtions of both PASCAL and FORTRAN that represent different subsets of the full language. Although there are standards for FORTRAN, many people take advantage of special features of a particular implementation and therefore reduce the transportability of the final code. For languages where there are less formal standards, the problems are often greater. There are, however, quite reasonable subsets that could be fully agreed upon to provide a basis for transportable codes. In this case, the learned societies can take a lead by defining the acceptable scientific subsets just as officially approved chemical nomenclature is defined. Organizations like the American Chemical Society or IUPAC could take the lead in standardizing these languages and force more uniformity in coding practices to make programs easier to follow.

An extremely important quality criterion is also the user friendliness of the program. There are several aspects to this. A first aspect is that there are two categories of users: specialists who understand all the details of the mathematical mechanics and also the software details and the nonspecialist who do not have, and should not need to have, an intimate knowledge of the mathematics and are mainly interested in obtaining correct results. Even for the first category of users, user friendliness is necessary. Use of the program always entails inputting information, which is a boring operation. A good screen layout helps to speed up this operation and helps one avoid making errors. Routines should be available to spot errors and to easily correct them. Providing this kind of facilities is a time-consuming business: we estimate from our own experience that it requires at least 3 times the programming time to make a user-friendly program compared to one that simply works. For the nonspecialist, there often is an additional problem, namely, the selection of the correct option. There are many excellent, mostly mainframe or at least minicomputer, statistical programs, but their potential is used only to a small degree by most users because of the time and the statistical expertise needed to learn to use them well. For nonspecialists, we think that one should avoid having too many options in a program or, when this is unavoidable for sound scientific reasons, one should make the program in such a way that the user is guided toward the correct option. For instance, a simple statistical problem such as the comparison of the means of two series of observations needs to be solved in different ways according to whether the number of observations is lower or higher than 25, the variances may be pooled or not, the assumption of normality is correct or not, the observations are paired or unpaired, and the test is one or two sided. For a person with sufficient knowledge about statistics, it is not difficult to select the correct procedure, at least in principle, because one often notices that users of statistics forget that real-world results do not always obey the ideal of normality. For the nonspecialist, the jargon is an unsurmountable difficulty. It is, however, possible to build a decision tree into the program that leads the user to the choice of the correct solution procedure, as we have tried to show with a program called BALANCE.[25] A final aspect of user-friendliness is the documentation available with the program. This constitutes another reason why software should be distributed in a more organized way than it is now. One cannot blame the author of a program distributed on a noncost basis if he does not put much work in a manual. However, to easily implement and to understand what the program does, a good manual is absolutely necessary. A reviewing procedure here is also of great help.

Returning to our original premise of the laboratory of 1990, we foresee the much greater use of chemometric methods to develop and optimize the procedures that collect and analyze the data required to solve future problems. The availability of well-documented, easy to obtain, implement, and use software will greatly facilitate these developments. We are of the opinion that the usual way of distributing software, i.e., to send it more or less free to friends and colleagues who ask for it, is outdated and does not respond to the requirements for user-friendly, easy to obtain, and validated software. It is high time that professionals take this situation in hand. We repeat that the entrance of publishers and learned societies, and preferably some combination of both, in this field would be an important service to the scientific community.

REFERENCES AND NOTES

(1) The Chemometrics Society *Chemometrics News* Bull. **1981**, *No. 7*.
(2) Massart, D. L.; Kaufman, L. "The Interpretation of Analytical Chemical Data by the Use of Cluster Analysis"; Wiley: New York, 1983.
(3) Hippe, Z. *Anal. Chim. Acta* **1983**, *150*, 11.
(4) Lindsay, R. K.; Buchanan, B. G.; Feigenbaum, E. A.; Lederberg, J. "Application of Artificial Intelligence for Organic Chemistry, The Dendral Project"; McGraw-Hill: New York, 1980.
(5) W. T. Wipke, see elsewhere in this journal.
(6) Dessy, R. E. *Anal. Chem.* **1984**, *56*, 1201A.
(7) Detaevernier, M. R.; Massart, D. L. *J. Pharm. Biomed. Anal.*, in press.
(8) De Smet, M.; Hoogewijs, G.; Puttemans, M.; Massart, D. L. *Anal. Chem.* **1984**, *56*, 2662.
(9) T. L. Isenhour, see elsewhere in this journal.
(10) Kool, P.; Michotte, Y. *TrAc* **1985**, *4* (2), 44.
(11) Debets, H. J. G.; Bajema, B. L.; Doornbos, D. A. *Anal. Chim. Acta* **1983**, *151*, 131.
(12) Glajch, J. L.; Kirkland, J. J.; Squire, K. M.; Minor, J. M. *J. Chromatogr.* **1980**, *199*, 57.
(13) Laub, R. J.; Purnell, J. H. *J. Chromatogr.* **1975**, *112*, 71.
(14) van der Wiel, P. F. A.; Maassen, R.; Kateman, G. *Anal. Chim. Acta* **1983**, *153*, 83.
(15) Brooks, S. H. *Oper. Res.* **1959**, *7*, 430.
(16) Deming, S. N.; Morgan, S. L. "INSTRUMENTUNE-UP"; Elsevier Scientific Software: Amsterdam, The Netherlands, 1984.
(17) van der Wiel, P. F. A.; Kateman, G. "CHEOPS"; Elsevier Scientific Software: Amsterdam, The Netherlands, 1985.
(18) Guttman, L. *Psychometrika* **1954**, *19*, 149.
(19) Kaiser, H. F.; Hunka, S. *Educ. Psychol. Meas.* **1973**, *33*, 99.
(20) Wolff, D. D.; Parsons, M. A. "Pattern Recognition Approach to Data Interpretation"; Plenum Press: New York, 1983.
(21) Malinowski, E. R.; Howery, D. G. "Factor Analysis in Chemistry"; Wiley: New York, 1980.
(22) Hopke, P. K.; Alpert, D. J.; Roscoe, B. A. *Comput. Chem.* **1983**, *7*, 149.
(23) Ghiorso, M. S. *Comput. Geosci.* **1983**, *9*, 391.
(24) Isenhour, T. L. *J. Chem. Inf. Comput. Sci.* **1984**, *24*, 2A.
(25) Massart, D. L.; Derde, M. P.; Michotte, Y.; Kaufman, L. "BALANCE"; Elsevier Scientific Software: Amsterdam, The Netherlands, 1984.