COMPROTEIN: A COMPUTER PROGRAM TO AID PRIMARY PROTEIN STRUCTURE DETERMINATION*

Margaret Oakley Dayhoff and Robert S. Ledley
National Biomedical Research Foundation
Silver Spring, Maryland

INTRODUCTION

Among the main chemical constituents of the human body—and, in fact, of all living things—are proteins. In addition to serving as component structural parts of many types of living tissues, the proteins are enzymes that are necessary in order that the chemical reactions which comprise the life processes may occur. The protein enzymes act to "decode" the message of the genes, interpreting this message in terms of specific chemical reactions which determine the physical and functional characteristics of the organism. Thus proteins play a uniquely vital role in the evolution, ontogeny, and maintenance of living organisms. It therefore becomes important when studying the basis of life processes to know the structure of the proteins themselves.

In spite of their highly complex role, the molecular structure of proteins is, in principle, relatively simple: they are long chains of only twenty different types of smaller molecular "links" called amino acids (see Figure 1). Each type of protein is characterized by a particular ordering of the amino acid links, and a major problem in finding the exact structure of a protein is to obtain the ordering of the amino acids in the chain. This ordering is of great interest because it is the order of the amino acids in a protein that is determined by the gene. Thus, according to current biological theory, the gene determines which proteins will be made by determining the order of the amino acids in the protein chain and it is these proteins in turn, acting as enzymes, that control the chemical processes that determine the physical and functional characteristics of the organism.

Finding the amino acid order of a protein chain has proved a time consuming process for the biochemist; in fact, only about 6 complete or almost complete protein orderings have been found so far, namely those of insulin [1], hemoglobin [2], ribonuclease [3], tobacco mosaic virus protein [4], myoglobin [5], and cytochrome C [6]. The basic technique used on all these proteins (with the exception of myoglobin) was to breakdown the long chain chemically into smaller fragment chains at several different points, to analyze the amino acids in each fragment chemically, and then to try to reconstruct the entire protein chain by a logical and combinatorial examination of overlapping fragments from the different breakdowns. It is in this reconstruction of the protein that the computer finds its application.

*The research reported in this paper has been supported by Grana GM 08710 from the National Institutes of Health, Division of General Medical Sciences, to the National Biomedical Research Foundation.

	Name	Abbreviation		Name	Abbreviation
1	alanine	ALA	11	leucine	LEU
2	arginine	ARG	12	lysine	LYS
3	asparagine	ASN	13	methionine	MET
4	aspartic acid	ASP	14	phenylalanine	PHE
5	cysteine	CYS	15	proline	PRO
6	glycine	GLY	16	serine	SER
7	glutamine	GLN	17	threonine	THR
8	glutamic acid	GLU	18	tyrosine	TYR
9	histidine	HIS	19	tryptophane	TRY
10	isoleucine	ILU	20	valine	VAL

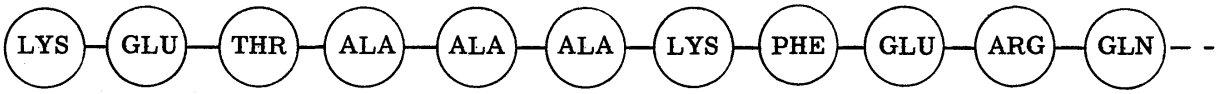


Figure 1. A listing of the amino acids with their abbreviations is shown in the upper section and the lower indicates part of the protein ribonuclease which actually comprises a chain of some 124 amino acids.

As a trivial example, suppose that for a protein one chemical breakdown produced the fragment chains of known ordering,

Breakdown P: AB, CD, and E

Where A, B, C, D and E each occur once and only once in the protein. Let us call this a complete breakdown, and let another breakdown, this time incomplete, produce the fragments

Breakdown Q: BC and DE

where A, B, C, D, and E represent amino acids. Here fragment BC in Breakdown Q (see Figure 2a) clearly overlaps the two fragments AB and CD of Breakdown P, and DE overlaps CD and E of breakdown P, giving as the reconstructed protein

ABCDE.

As another example, consider the more common case where the amino acid components of a fragment are known, but the order of these within the fragment is unknown. Let parentheses indicate that the order they enclose is unknown [e.g., (A, B, C) represents the six permutations of A, B, C; (D, E) represents either DE or ED; (A, B, C) (D, E) represents the $6 \times 2 = 12$ fragments of each of the six permutations in (A, B, C) followed by DE or ED etc.], and suppose that one complete breakdown is

Breakdown P: (A, B, C) and (D, E)

and that another, incomplete, breakdown is

Breakdown Q: (A, B) and (C, D)

Clearly (C, D) of breakdown Q overlaps (A, B, C) and (D, E) of breakdown P but (A, B) is contained within (A, B, C). Hence, since each amino acid has distinct "left" and "right" ends, two possible protein reconstructions result, namely (see Figure 2b)

(A, B) (C) (D) (E) and (E) (D) (C) (A, B)

where in each possibility the order of A, B still remains unknown. Such partial reconstructions frequently occur, and pinpoint for the biochemist that portion of the molecule on which further effort is required.

Unfortunately, however, the problems involved in reconstructing proteins are not as simple as in the examples just given. The largest protein analyzed so far, the tobacco mosaic virus protein, has only 158 amino acids whereas proteins usually have chains of many hundreds of amino acid links. Since the number of combinations on n things taken r at a time ($1 < r < n$) increases more rapidly than does n itself, it is to be expected that the difficulties in piecing together the fragments of a protein will increase proportionally faster than the number of amino acids in the protein. In addition, there may be occasional errors in the fragments reported by the biochemist,

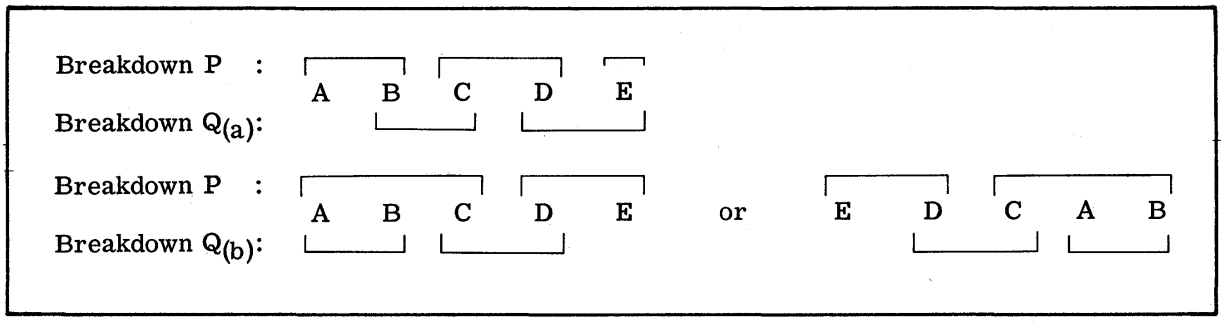


Figure 2. Illustration of two different breakdowns of a protein into amino acid fragments (see text).

as well as other aberrations in the data. Hence the logical and combinatorial problems can become severe, and a computer is then required to assist in the analysis.

The advantage of computer aid is that it may help significantly to extend the current chemical analysis methods of determining the amino acid sequences of proteins to many more and much larger proteins. By exhaustively analyzing the possibilities of protein reconstructions, the computer may assist in determining the best next step to try in the chemical analysis processes. In addition, it should be noted that presently the chemical analysis is carefully planned to produce results that will be logically simple for mental analysis. The use of a computer to perform the logical analysis may thus allow significant simplification and further systematization of the chemical experimental procedures by placing more of the burden on the automated logical and combinatorial analysis and less on the experimental procedures.

In this paper we shall describe a completed computer program for the IBM 7090, which to our knowledge is the first successful attempt at aiding the analysis of the amino acid chain structure of protein [7]. The idea was conceived by us in 1958, but actual programming was not initiated until late 1960. D. F. Bradley, S. A. Bernhard, and W. L. Duda have independently reported, in an as-yet unpublished paper [8], progress in approaching a similar problem. R. Eck has reported on a system for using marginal-punch cards to aid in certain aspects of the logical analysis problem [9].

A SIMPLIFIED ILLUSTRATION

Discussion of the programming methods utilized will be clarified if we first consider a simple illustration. Suppose a complete breakdown P is made by the biochemist as in Figure 3, and that another breakdown Q is also known but not complete (see Figure 3).

Complete breakdown P LIST		Incomplete breakdown Q LIST	
p_1	(R)(A,B)	q_1	(A)(B,B,D)
p_2	(D)(B)(C,A)	q_2	(A)(C,A,C)
p_3	(A)(C)(D)(X,A)(C)	q_3	(X)(A,C,B)
p_4	(B)(D)(B)(D)(A,B,D)	q_4	(B)(A,A,C,D)
p_5	(C)(A)(Z)	q_5	(A)(B,C,D)

Figure 3. Breakdowns of protein fragments for the illustration in the text.

In Figure 4 we show how such breakdowns P and Q can occur from our hypothetical protein, but the problem is to reconstruct this from the fragments given in Figure 3. Since each fragment q_i of breakdown Q must either overlap several fragments of P, or else be included within some fragment of P, let us start by making a list for each q_i of all possible such associated P fragments; Figure 5 shows such lists for our illustration. As an example of how each entry in a list is found, consider the test of whether or not q_4 overlaps p_4p_5 where

q_4 is (B)(A,A,C,D)

and where

p_4 is (B)(D)(B)(D)(A,B,D), p_5 is (C)(A)(Z)

The problem is to determine if each acid of q_4 can be accounted for in p_4 and p_5. First note that the maximum overlap between q_4

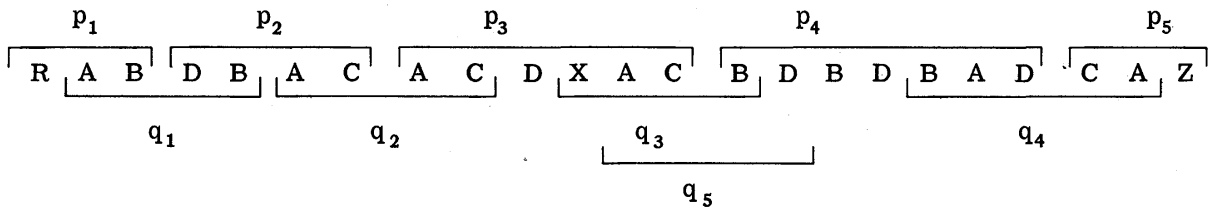
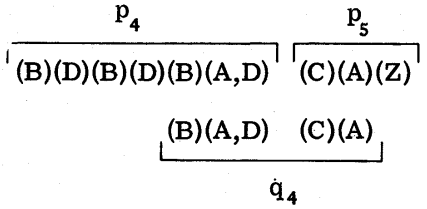


Figure 4. Illustration of sources of peptide fragments from protein molecule illustrated in the text.

q_1	q_2	q_3	q_4	q_5
$p_1 p_2$	$p_2 p_3$	$p_3 p_4$	$p_1 p_3$	$p_1 p_2$
$p_1 p_4$	$p_2 p_5$		$p_4 p_3$	$p_2 p_4$
$p_2 p_4$	$p_3 p_5$		$p_4 p_5$	$p_3 p_2$
$p_4 p_2$				$p_3 p_4$
				$p_4 p_2$
				$p_4 p_5$

Figure 5. The q lists for the illustration in the text.

and p_4 is (B,A,D), on the right of p_4. This leaves (A,C) of q_4 "hanging over" on the right of q_4, to be accounted for in p_5. This is clearly possible, resulting in the overlap:



In order to determine all the entries in all the lists, such trials must be made by the computer for every pair of fragments $p_i p_j$, for each q_k.

However, just forming the lists is but the first step in reconstructing the protein chain. The next step is an elimination process to leave only the consistent possibilities. For instance, q_3 can only arise from p_3p_4; hence p_3 must be followed by p_4, and p_4 must be preceeded by p_3, and therefore all other possibilities in other lists involving p_3 and p_4 can be eliminated—such as p_1p_4, p_2p_4 and p_5p_4 in the q_1 list of Figure 5, p_3p_5 in the q_2 list, p_4p_3 in the q_4 list, and p_2p_4 and p_3p_2 in the q_5 list.* This leaves in the list for q_1 only p_1p_2 and p_4p_2. If we first assume that p_1p_2 is overlapped by q_1, then in the q_4 list only p_4p_5 remains, and hence in the q_2 list only p_2p_3 remains, giving altogether these adjacent fragments,

p_3p_4, p_1p_2, p_4p_5, and p_2p_3

which determine the structure as

$p_1p_2p_3p_4p_5$

On utilizing p_1, p_2, p_3 and p_4 we find as the final structure:

(R)(A)(B)(D)(B)(A)(C)(A)(C)(D)(X)
(A)(C)(B)(D)(B)(D)(B)(A,D)(C)(A)(Z)

Returning to the second possibility in the q_1 overlap list, namely p_4p_2, this leaves only p_1p_3 in the q_4 list, which in turn leaves only p_2p_5 in the q_2 list. Hence a second possibility for adjacent fragments is

p_3p_4, p_4p_2, p_1p_3, and p_2p_5

which gives the structure

$p_1p_3p_4p_2p_5$

On utilizing q_3, q_2, q_4, and q_2, we find

(R)(B)(A)(A)(C)(D)(X)(A)(C)(B)(D)
(B)(D)(D)(A)(B)(D)(B)(A)(C)(C)(A)(Z)

Actually, it is also necessary to eliminate the occurrence of p_4 in the lists by replacing it with p_3, which stands for p_3p_4. This is to insure that an impossible succession of conditions such as p_1p_2, p_2p_3, p_3p_1 is not produced.

COMPUTER HANDLING OF BIOCHEMICAL INFORMATION

The problems involved in writing a computer program, however, are not as straightforward as the above illustration might indicate. The biochemist utilizes enzymes to break up (hydrolyze) the protein into the fragments that we have been considering; these fragments are called peptides by the biochemist. The enzymes commonly used, such as subtilisin and chymotrypsin, produce an assortment of peptides which may overlap each other. Hence a problem arises in actually arriving at a complete set of peptide fragments, as illustrated above in the breakdown P. In addition, for several reasons, the biochemical experiments very often do not result in integer values for the number of amino acids of a particular kind that occur in a peptide fragment. This second uncertainty problem must also be taken into account by the computer program. Furthermore, there may be experimental errors in the amino acid composition and ordering of some peptides.

In the case of overlapping peptide fragments from a hydrolytic breakdown, the computer program tries to reconstruct a complete set of fragments from overlapping subsets of fragments. The procedure of accomplishing this is to look for every group of two, three, or four acids known to be adjacent in some peptide. Then, for each such group, the probability that this particular group will occur again in the protein chain is computed from the amino acid frequency data. For instance, if the ordered pair LYS-PHE occurs, and it is known that there are five LYS residues and four PHE residues in the entire protein chain of, say, 150 amino acid links altogether, then the probability that another such LYS-PHE pair will occur is approximately $4 \times 3/150$.

If the probability is small that another such group occurs, it is most likely that all of the peptides containing this group should arise from the same part of the protein; hence these peptides are sorted out. All possible fragments that can be reconstructed from these (overlapping) peptides are then determined.

It may happen, however, that all these peptides cannot "fit" into any reconstructed fragment; this indicates either that some peptide must arise from a different place on the protein or that there may be an experimental error. In such a case the experimental results are reconsidered from a chemical point of view.

Of course, there is a small but finite probability that a misinterpretation can be made at this point and an erroneous peptide constructed, such as could occur if a highly unlikely configuration actually occurred more than once in the protein or if there were lost peptides from a particular region but the existing peptides fortuitously fit. However, it is likely that in any case, in later building up of the protein, an inconsistency would arise, leading to the rejection of this erroneously constructed peptide.

Some peptides may contain two or more groups on which searches are made. Reconstructed fragments containing these peptides must be joined together themselves. Hence the program merges these fragments to obtain all possible larger fragments. Such procedures will fix the relationships of many amino acids beyond that given in the initial data. These new relationships change the probabilities of occurrence of the two, three, or four amino acid groups. The probabilities are accordingly recalculated, and once more searches on improbable groups are made, leading to further merges of fragments into even larger fragments. This process is repeated by an iterative program until less than about 20 fragments remain.

Further details must be taken into consideration by the program: the set of fragments may still not be complete; there may exist alternative possibilities for a fragment; and there may be gaps in the chain where all the peptides were lost.

After obtaining a complete, or almost complete, set of fragments by iteration of the searching procedure, the program can continue toward reconstructing the entire protein utilizing the remaining peptides not used in the building up of the complete set P as the Q set of peptides (see example above).

It should be noted that in the various phases of the reconstruction of the protein the assumption is made that the total amino acid content of the entire protein and of the fragments is known. There is always, however, some experimental uncertainty in the number of amino acids of each type in the protein, to within a fraction of one amino acid. As a rule, the larger number of amino acids is always chosen initially. If an extra acid is thereby included in the computations, it may

be eliminated at the end, by a procedure described below.

Completing the final reconstruction of the protein again can present further details which must be taken into consideration by the computer program. Some peptides may appear to overlap at only one amino acid. If this occurs it would be unwise to conclude definitely that this represented a true overlap. Hence "pseudofragments" are used which consist of each overlapping fragment without the common amino acid.

Single amino acids to complete the P set, as required by the amino acid constitution of the protein are considered with the larger fragments.

If extra amino acids are so included, the final answer showing which fragments must be attached may place no attachment restrictions on these extra acids. In this case, if the acid arose from a fractional experimental result (see above), one may presume that it does not actually occur in the molecule. Otherwise further experimentation may be required. For example, if amino acid X is added to the P list in Figure 3, the Q lists will be unaffected and the resulting answers will be unchanged. One might conclude that either X really didn't belong in the P list or it could be at either end of the molecule. It is sometimes known which amino acids are on the right and left ends of the protein itself, and this information can further reduce the final possibilities.

DESCRIPTION OF COMPUTER PROGRAMMING SYSTEM

The computer programming system to aid protein analysis has been written in a flexible manner. The computer input and output is in terms of three letter abbreviations for the amino acids, with the parentheses notation for ordered and unordered sets as described above. Intermediate results are printed out for examination by the biochemist; in fact the entire process is geared for a close cooperative effort between the computer and the biochemist during the entire analysis. This is necessary in order to take advantage of the special conditions presented by any particular protein and type of chemical experimental procedures. For example it might be convenient to omit all prolines from the peptides, or not to consider a distinction between Glu and Gln. Special rules might be introduced regarding end-groups from hydrolyses by certain enzymes, etc. Such special considerations can be handled by the programming system, and make it easier to spot errors in the experimental data.

The programming system is based on the following six programs:

(1) MAXLAP: Program to find the maximum possible overlap between any two peptides with any amount of ordering information known.

(2) MERGE: Program to find all possible overlapping configurations of two peptides.

(3) PEPT: Program to find all possible fragments that are consistent with the overlapping of any number of peptides.

(4) SEARCH: Program to search on probabilistic considerations all peptides which contain an unusual group of amino acids.

(5) QLIST: Program to generate the Q-lists of possible associated sets of P peptides over which each q_i fragment can fit.

(6) LOGRED: Program to perform the logical reduction of the Q-lists to obtain all possible protein structures that are consistent with the data.

Since detailed flow diagrams would consume too much space and not be appropriate for the present discussion, we have included here only gross overall flow diagrams of these programs. Each of the six programs will now be described, and simple examples illustrating some of the methods involved will be given. (For further details see "Sequencing of Amino Acids in Proteins Using Computer Aids," Report No. 62072/8710, National Biomedical Research Foundation, Silver Spring, Md., July 1962.)

Program MAXLAP (Figure 6). In this program p and q are peptide fragments, PCOM and QCOM are lists of acids from these peptides respectively which may provide the maximum overlap. After setting up a tentative maximum number of positions (i.e., amino acids) of overlap, three cases may be distinguished as illustrated by the three examples of Figure 7. In the first example there is the successful maximum overlap situation where all of p or q is overlapped. Here MV is the list of all the acids from PCOM and QCOM which match; with this maximum overlap, the complete maximum overlap protein fragment is shown. In

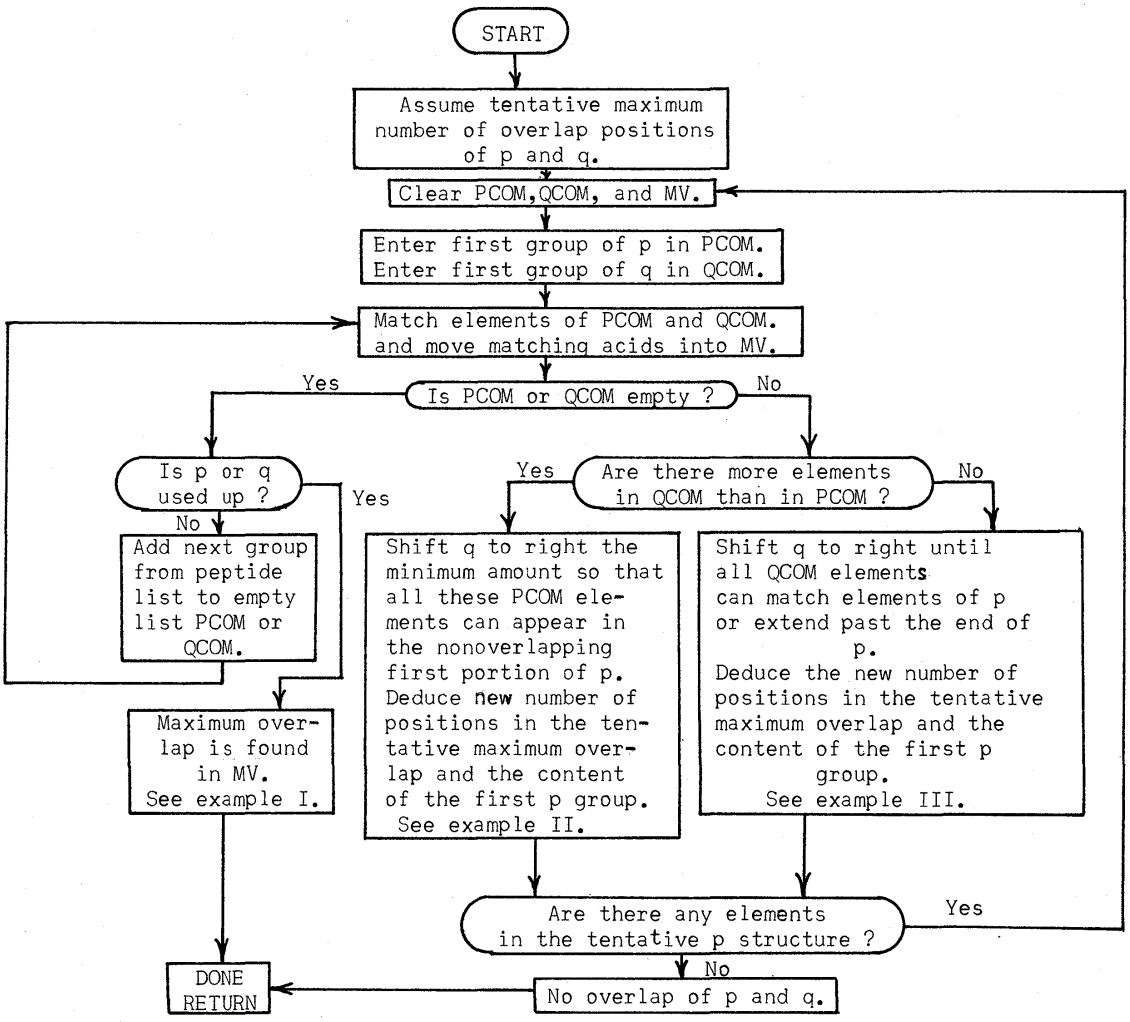


Figure 6. Flow Chart for Program MAXLAP.

Example I	**p:**	**(A,B)(C,D)**
	q:	**(C,B,A)(F,G,D)**
	MV:	**(A,B)(C)(D)**
	Protein fragment with maximum overlap:	**(A,B)(C)(D)(F,G)**
Example II	**p:**	**(C)(D,E,F)**
	q:	**(D,C,E)(G,H)**
	New tentative maximum Overlap structures	**(C)(F,D,E) (D,C,E)(G,H)**
	Protein fragment with maximum overlap:	**(C)(F)(D,E)(C)(G,H)**
Example III	**p:**	**(A,B)(B,E,A)(D)**
	q:	**(B)(A,D,E)(F)**
	New tentative maximum Overlap structures	**(A,B)(B,E,A)(D) (B)(A,D,E)(F)**
	Protein fragment with maximum overlap:	**(A,B)(B)(A,E)(D)(F)**

Figure 7. Examples of the three cases considered in flow chart of program MAXLAP.

the second example all of p cannot be overlapped by q, and hence new tentative overlap positions must be assumed. Here F is the limiting acid since it does not appear in q. In the third example even though D appears in both p and q, new tentative overlap positions must be assumed, because in p the D acid is restricted in position at the right.

Program MERGE (Figure 8). Once the maximum overlap has been determined, all other possible overlaps can be determined. Several cases can occur. First the essentially trivial cases of p and q entirely disjoint, as in

p: (A,B,C) q: (D,E)

or else q a subset of p as in

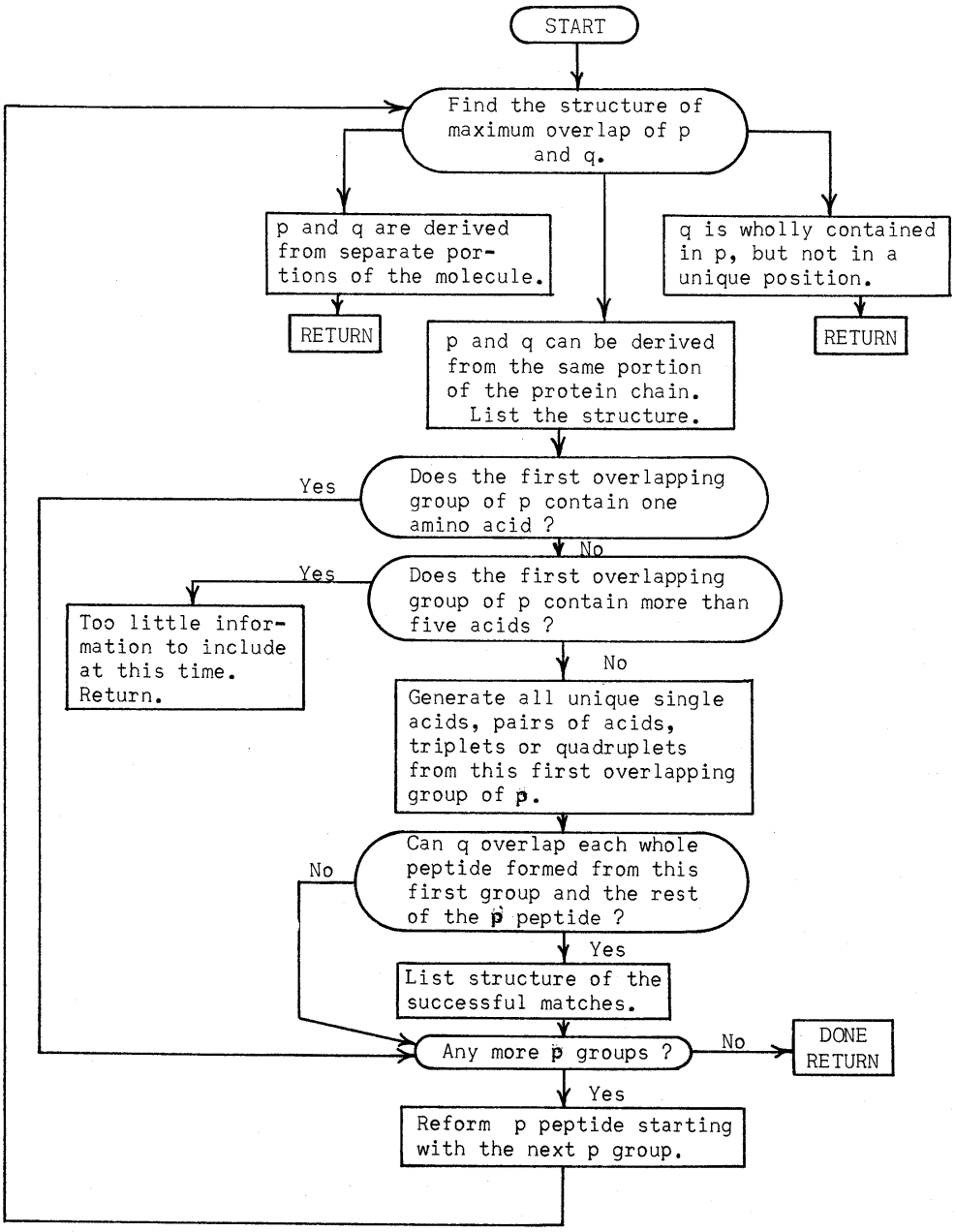


Figure 8. Flow Chart for Program MERGE.

p: (A,B,C,D) q: (B,C)

If neither of these is the case, as for example in

p: (A,B,B,D) q: (B)(B,D)(G)

the first overlapping group of p is considered, which for our later example is

(B,B,D)

If this first overlapping group contains not more than five nor less than two acids, it warrants further consideration. A list is made of all singles, pairs, triplets and quadruplets of acids that can be formed from this overlapping group, which for our example is

(B) (D) (B,D) (B,B)

Next each of these is examined to see if it can overlap with q. For an example we have respectively

(A,B,D)(B)(B,D) G, none,
(A,B)(B)(D)(B)(G), and (A,D)(B)(B)(D)(G)

where we have underlined the overlapping group to the left of which p fits and to the right of which q fits. Finally the peptide is re-formed starting with the next group, and so forth, until all the overlapping groups of p and q have been considered.

Program PEPT (Figure 9). This program extends the previously discussed program MERGE in that it finds all of the possible structures of the protein chain consistent with the overlapping of all the peptides obtained from a search for all experimental peptides with a rare configuration of amino acids. The overlapping portion of these acids must contain the group of rare amino acids, called SAA in the flow chart. The list resulting from the search is called MCO in the flow chart.

Program SEARCH (Figure 10). This program systematically looks at each pair, triplet, and quadruplet of amino acids that are known to occur together from the experimental peptide fragment data. For each such group, the probability of its occurrence is computed from the amino acid frequency data as described above. A list, called Num(L) in the flow chart, is made of these groups of amino acids known to occur together which are improbable of occurrence (i.e., less probable than some chosen value). The letter L is used to index the elements of the list Num(L). Finally the program utilizes PEPT to generate and print out all possible merged structures. For example suppose a search was made on the ordered pair

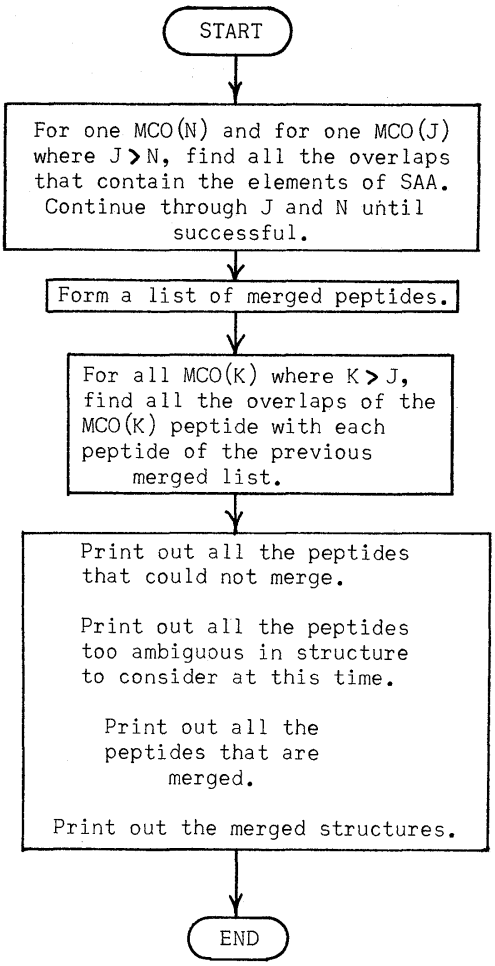


Figure 9. Flow Chart for Program PEPT.

LYS—PHE

and there resulted the following fragments:

(ALA)(ALA, ALA, LYS)(PHE)
(ALA)(ALA, LYS)(PHE)
(ALA)(LYS, PHE, GLU, ARG)(GLU)
(LYS)(PHE)

The merged structure becomes

(ALA)(ALA)(ALA)(LYS)
(PHE)(GLU,ARG)(GLU)

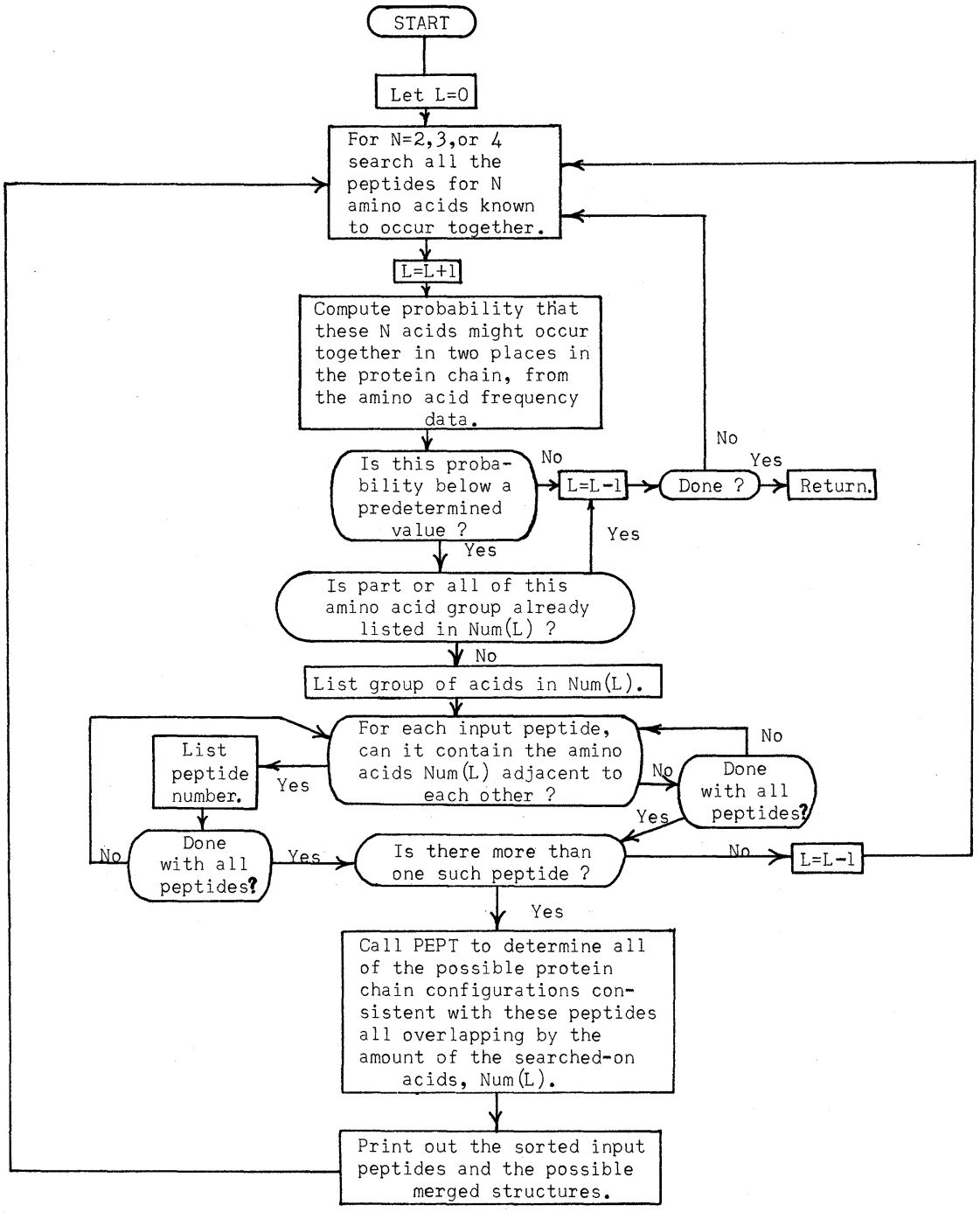


Figure 10. Flow Chart for Program SEARCH.

Program QLIST (Figure 11). This program forms the lists of peptides related by each fragment q_1. It is to be noted that each element of a Q list may contain up to five p fragments (although in our example of section 2 only two peptides appeared in each element of the Q lists). The input to this program is a list P of peptides which in some order will reconstruct the original protein and a list Q of peptides which give additional ordering information about the protein. In the flow chart P' is a hypothetical peptide

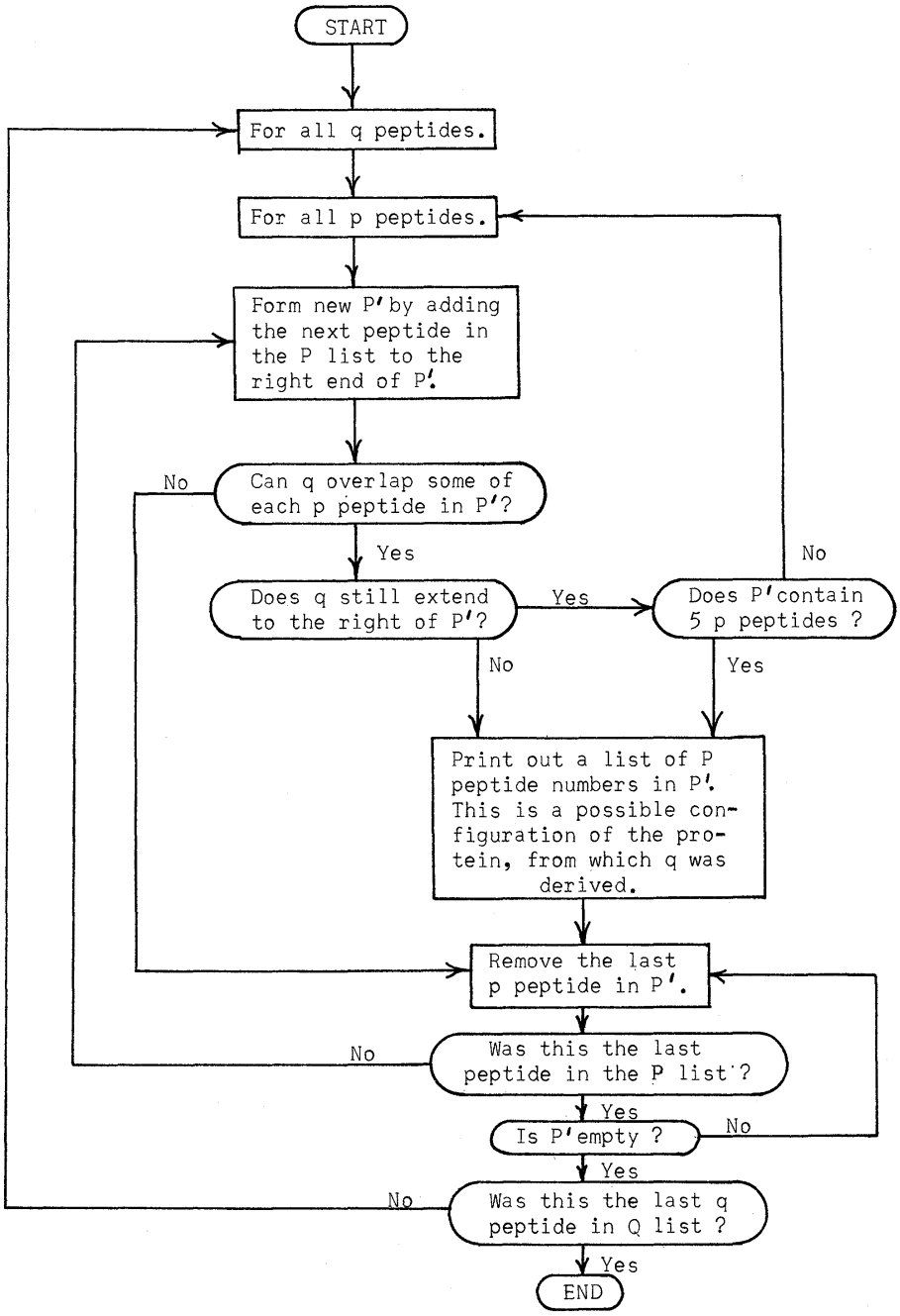


Figure 11. Flow Chart for Program Q LIST.

constructed by the juxtaposition of up to five P peptides.

Program LOGRED (Figure 12). In this program the Q lists are given. Calling each term of a Q lists a condition, the flow chart involves the lists: MQ(M) of conditions on the assumption of the Mth tentative condition; MQI of conditions being considered; and IR(M) of tentative conditions which determine a possible protein structure. The symbol MTI(J) is the last condition considered in the Jth Q list. The program follows a tree structure of possibilities, keeping tract of tentative conditions until a branch is eliminated or comes to a successful conclusion. The program follows with greater generalization

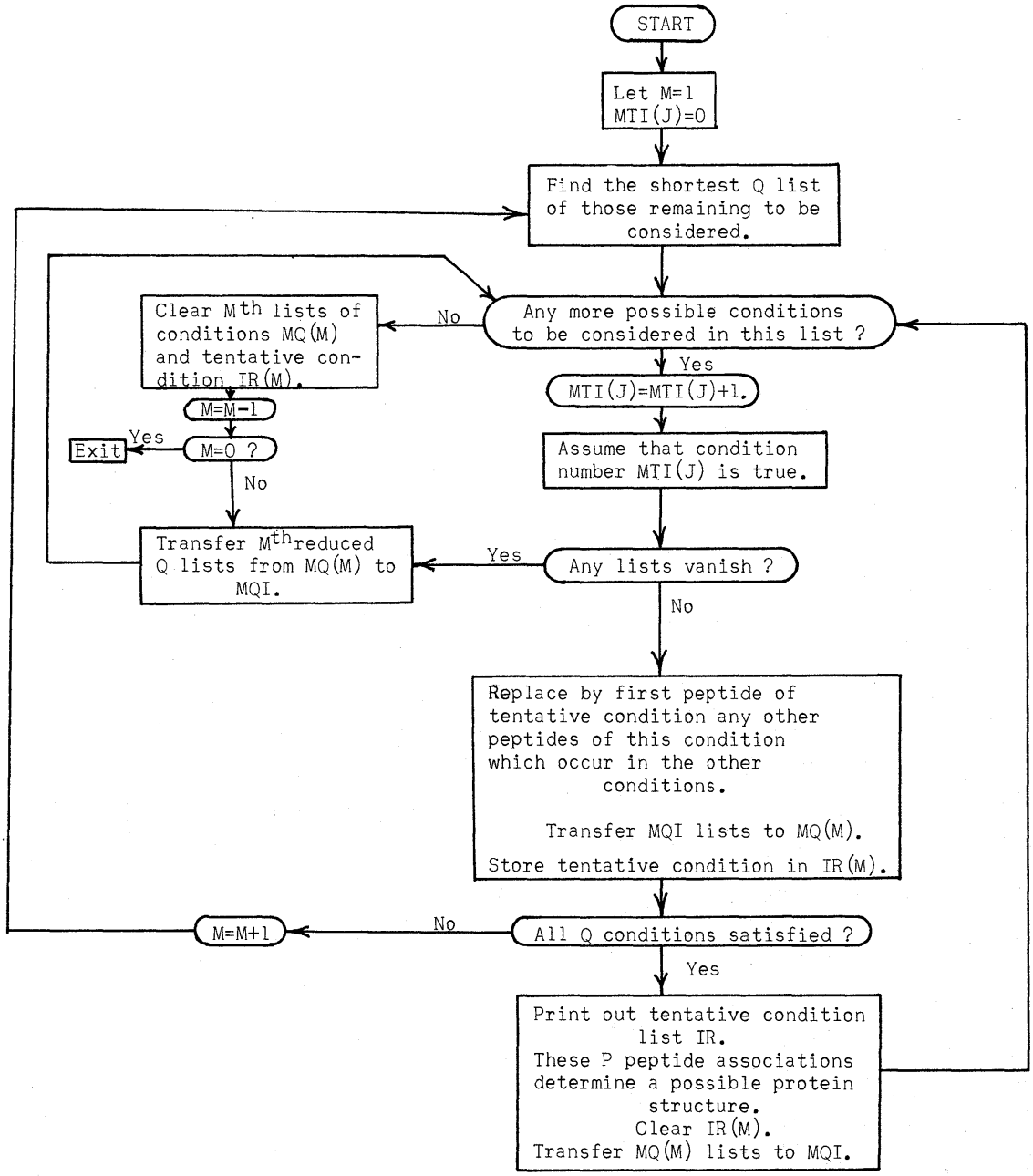


Figure 12. Flow Chart for Program LOGRED.

the method of the "simplified illustration" of the second section of this paper.

SUMMARY AND CONCLUSION

The computer program described in this paper becomes useful when there is a large number of small peptide fragments resulting from the breakdowns which are to be woven into a consistent and unique structure. This is a long tedious process when carried out by hand, and is subject to careless errors and impatient overlooking of all alternative

possibilities.* The completed IBM 7090 Computer program has been successfully tested on a hypothetical subtilysin breakdown of ribonuclease into over eighty fragments.

Just as the proteins are composed of chains of the same types of molecules, the genetic substances desoxyribonucleaic acid (DNA) and ribonucleaic acid (RNA) are composed of chains of only four different types of molecules called the nucleotide bases. It is possible that the order of the molecules in these substances can also be determined by the aid of this computer program and some computer experiments in this direction have been made. However, application of these techniques to DNA and RNA still awaits further development in the chemical experimental methods.

*Dr. Wm. Dreyer, of National Institutes of Health, has developed chemical techniques for isolating a large percentage of the peptides formed in a subtilisin hydrolysis, and for determining the total amino acid content and identifying the right and left ends of the peptide fragments. This experimental technique is very rapid and can be mechanized to a large extent; thus data taking should be reduced to months instead of years. The computer program is ideally suited for analysing this type of data.

The computations in this paper were done in the Computing Center of the Johns Hopkins Medical Institutions, which is supported by Research Grant, RG 9058, from the National Institutes of Health and by Educational Contributions from the International Business Machines Corporation.

REFERENCES

1. Sanger, F., Science, 129, 1340 "1959." Hirs, C. H. W., Moore, S., Stein, W. H., J. Biol. Chem. 235, 633, "1960" Spockman, D. H., Stein, W. H., Moore, S., J. Biol. Chem. 235, 638 "1960."
2. Braunitzer, G., Gehring-Mueller, R., Helschmann, N., Helse, K., Hobom, G., Rudloff, V., and Wittmann-Tiebold, B., Hoppe Seyler Z. Physiol. Chem., 325, pp. 283-6, Sept. 20, 1961.
3. Hirs, C. H. W., Moore, S., and Stein, W. H., J. Biol. Chem. 235, 633 (1960).
4. Tsugita, A., Gish, D., Young, J., Fraenkel-Conrat, H., Knight, C. A. and Stanley, W. M., Pros. Natl. Acad. Sci., Vol. 46, pp. 1463-9, 1960.
5. Kendrew, J., Watson, H., Strandberg, B., Dickerson, R., Phillips, D. and Shore, V., Nature, Vol. 190, pp. 666-70, May 20, 1961.
6. Margoliash, E., Smith, E., Kreil, G., and Tuppy, H., Nature 192, 1121-7, (Dec. 1961).
7. Ledley, R. S., Report on the Use of Computers in Biology and Medicine, Natl. Acad. of Scis.-Natl. Research Council, May 1960.
8. Bradley, D. F., Bernhard, S. A., Duda, W. L., unpublished work.
9. Eck, R., Nature, Jan. 20, 1962, 241-243.