Report Number: WUCS-92-29

1992-08-01

DNA Mapping Algorithms: Strategies for Single Restriction Enzyme and Multiple Restriction Enzyme Mapping

Will Gillett

An approach to high-resolution restriction-fragment DNA mapping, known as Multiple-Restriction-Enzyme mapping (MRE mapping), is present. This approach significantly reduces the uncertainty of clone placement by using clone ends to synchronize the position in of clones within different maps, each map being constructed from fragment-length data produced by digestion of each clone with a specific restriction enzyme. Maps containing both fragments-length data and clone-end data are maintained for each restriction enzyme, and synchronization between two such maps is achieved by requiring them to have "compatible" clone-end map projections. Basic definitions of different kinds of maps, such as restriction sites maps,... Read complete abstract on page 2.

DNA Mapping Algorithms: Strategies for Single Restriction Enzyme and Multiple Restriction Enzyme Mapping

Will Gillett

Complete Abstract:

An approach to high-resolution restriction-fragment DNA mapping, known as Multiple-Restriction-Enzyme mapping (MRE mapping), is present. This approach significantly reduces the uncertainty of clone placement by using clone ends to synchronize the position in of clones within different maps, each map being constructed from fragment-length data produced by digestion of each clone with a specific restriction enzyme. Maps containing both fragments-length data and clone-end data are maintained for each restriction enzyme, and synchronization between two such maps is achieved by requiring them to have "compatible" clone-end map projections. Basic definitions of different kinds of maps, such as restriction sites maps, restriction fragment maps and clone end maps, are presented. Several specifications notations, such as sequence-set notation and sequence-set-tree notation, for describing the structure of these maps, are defined. Basic concepts, such as the match/merge approach to map incorporation, extension vs. assimilation and ambiguity, are exposed. Supporting techniques, such as window sizing, window placement, and ambiguity resolution, are also discussed. A mathematical analysis of how MRE mapping effects false positives and false negatives is presented. For concreteness, MRE mapping is presented using a specific methodological framework. However, many of the concepts and techniques have a wider range of use than just high-resolution restriction-fragment mapping.

DNA Mapping Algorithms: Strategies for Single Restriction Enzyme and Multiple Restriction Enzyme Mapping

Will Gillett

WUCS-92-29

August, 1992

Department of Computer Science
Washington University
Campus Box 1045
One Brookings Drive
St. Louis MO 63130-4899

This work was supported by the James S. McDonnell Foundation under Grant 87-24 and NIH under Grant R01 HG00180.

ABSTRACT

An approach to high-resolution restriction-fragment DNA mapping, known as Multiple-Restriction-Enzyme mapping (MRE mapping), is presented. This approach significantly reduces the uncertainty of clone placement by using clone ends to synchronize the positioning of clones within different maps, each map being constructed from fragment-length data produced by digestion of each clone with a specific restriction enzyme. Maps containing both fragment-length data and clone-end data are maintained for each restriction enzyme, and synchronization between two such maps is achieved by requiring them to have "compatible" clone-end map projections.

Basic definitions of different kinds of maps, such as restriction site maps, restriction fragment maps and clone end maps, are presented. Several specification notations, such as sequence-set notation and sequence-set-tree notation, for describing the structure of these maps, are defined. Basic concepts, such as the match/merge approach to map incorporation, extension vs. assimilation and ambiguity, are exposed. Supporting techniques, such as window sizing, window placement, and ambiguity resolution, are also discussed. A mathematical analysis of how MRE mapping effects false positives and false negatives is presented.

For concreteness, MRE mapping is presented using a specific methodological framework. However, many of the concepts and techniques have a wider range of use than just high-resolution restriction-fragment mapping.

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF TABLES

1. Introduction

1.1. An Overview of DNA Restriction Mapping

Deoxyribonucleic acid (DNA) is the genetic material that supplies the blueprint for an organism's development. A DNA molecule is composed of **nucleotides**, each nucleotide consisting of a sugar, a phosphate, and one of the four bases: A (Adenine), T (Thymine), C (Cytosine), and G (Guanine). A nucleotide is distinguished by the base it contains. Sugar-phosphate bonds link the nucleotides into strands, and a base on one strand can "bond" with a base on another strand. However, only certain base bondings are allowed: A bonds with T, and C bonds with G. Thus, A and T are known as **complementary bases**, as are C and G. A DNA molecule is made of two complementary DNA nucleotide strands bound together by this base pairing, the base sequence on one strand determining the complementary sequence on the other strand.

DNA restriction mapping [1-8] deals with determining the *positions* of specific sites of interest along a given DNA strand, or **genome**. The sites of interest are called **restriction sites**, and consist of a specific subsequence of DNA, often six nucleotides long. These restriction sites are recognized by specific enzymes, known as **restriction enzymes**; a restriction enzyme cleaves (or cuts) DNA that it encounters at exactly these restriction sites. Thus, given sufficient time a restriction enzyme reacting with a strand of DNA will completely **digest** it, producing fragments of DNA whose lengths are exactly the distance between two successive restriction sites. The process of **electrophoresis** can be used to measure the approximate lengths of these fragments, which are known as **restriction fragments**. If it were possible to (a) identify each restriction fragment present in the genome, (b) determine the length of each restriction fragment, and (c) determine the order of the restriction fragments in the genome, then it would be possible to construct the **map** of the restriction sites.

The mechanism for obtaining this information is somewhat indirect. Ordering of the restriction fragments is achieved by cleaving multiple copies of the original DNA at random positions to produce randomly overlapping strands of DNA, known as **clones**. Each clone is then completely digested by the restriction enzyme (of interest), and electrophoresis is used to determine the lengths of the restriction fragments within it. This set of restriction fragment lengths is known as the **fingerprint** of the clone. Overlap between the clones is inferred based on the similarity of the fingerprints, and the order of the clones is inferred based on multiple clone overlap. As overlap between the clones is inferred due to a significant number of restriction fragments of similar (within measurement error bounds) lengths, the exact order of the restriction fragments within each clone may remain unknown; only the relative (partial) order of large groups of fragments may be inferrable. As more clones are found to overlap a specific region of the original genome, the random positions of the clone ends are used to refine the original partial order (of the restriction fragments) by reducing the size of the groups for which the fragment order is unknown.

This process of DNA restriction mapping is analogous to solving a large jigsaw puzzle. However, the uncertainty of where a clone should be placed can be significant, due to measurement error (produced during electrophoresis), experimental error (produced during cloning or digestion with the restriction enzyme), and certain biological properties of the DNA being mapped (e.g., two fragments of the same length do not necessarily contain the same sequence of nucleotides). When putting together a jigsaw puzzle, the pieces of the puzzle have several cues (shape, color, pattern on the surface) which can be used to guide their ultimate positioning in the final solution. In DNA restriction mapping, the clones have no shape or color, but the fingerprint information can be viewed as a "pattern" to be matched between potentially overlapping clones. The objective is to find a consistent positioning of clones with respect to one another in which fragments in different clones can be identified with one another, while all fragments of each clone remain contiguous (i.e., no "gaps" or unpaired fragments are present internally). There may be multiple "solutions" to this restriction map puzzle, and, in the absence of supplementary data that distinguishes between them, the one (or ones) which is most compact is preferred.

1.2. Formalization of the DNA Mapping Problem

The DNA mapping problem can be abstracted to the Shortest Common Matching String (SCMS) problem, as defined by Turner[9]. Within this abstraction the clones are abstracted to **bags**, and the lengths of the restriction fragments in the clone are abstracted to **symbols**. (In this abstraction of the lengths to symbols, it is assumed that there is no measurement error and that all fragments of the same length are identified with the same symbol.) A **bag** b = $<a_1, a_2, \cdots, a_h>$ is a multi-set in which a symbol ($a_i \in \Sigma$, where Σ is some finite alphabet) can occur more than once. This mathematical formalism is appropriate because a clone (i.e., a bag) can contain more than one fragment (i.e., a symbol) of the same apparent length (i.e., fragments of the same length, within measurement error bounds).

If $s = a_1 \cdots a_h$ is a sequence of symbols, then let $<s>$ denote the bag $<a_1, a_2, \cdots, a_h>$. The same bag $<s>$ represents all strings that can be obtained by permuting the symbols in s. s can be thought of as a sequence of fragments (in the order that they actually occur in the genome), and $<s>$ can be thought of as the same data from which the ordering information has been removed. A bag b and a string s are said to **match** if s contains a substring s' such that $<s'> = b$. The SCMS [9] problem can now be defined as follows:

Given a set of bags $B = \{b_1, \cdots, b_n\}$, find a minimum length string s that matches every bag in B.

The sequence of symbols (i.e., fragment lengths) in s constitutes a solution to the DNA mapping problem, given the bags (i.e., clones) in B. Turner[9] proved that the SCMS problem is NP-hard. Similar work has been done by Turner[10], Rhee[11, 12], and Lewis and Gillett[13].

1.3. Data Collection

The type of data considered in DNA mapping is the clone fingerprint data. Prior to any mapping, the original genome to be mapped is duplicated using "traditional biological means". Then, the DNA is randomly cleaved into smaller sections by partially digesting it with a **cloning restriction enzyme**; this produces random **clone inserts**. The partial digestion process causes different copies of the DNA to be cleaved at some (randomly selected) restriction site, but not at all restriction sites. This tends to produce clone inserts which have random overlap with one another. This is depicted in Figure 1.1, where four clone inserts (which will be used later in a running example) are shown in their positional context within the genome from which they come. The ends of these clones correspond to sites randomly cleaved by the cloning restriction enzyme during the partial digestion; other sites are present within the clones at which cleavage did not occur.

The cloning restriction enzyme is usually selected to be different than any of the restriction enzymes being mapped; in the protocol described here, it is *assumes* that they are different. This implies that clone-end sites do not coincide with the sites of any of the restriction enzymes being mapped. Clone inserts are inserted into a biological organism known as the **λ phage**, which is a virus used as a cloning vector (i.e., a mechanism used to reproduce many copies of a clone insert). This is depicted in Figure 1.2. The body of the λ phage is removed (leaving a left and a right λ arm) and a clone insert is replaced in its stead. The λ arms are engineered so that they do not contain restriction sites corresponding to any of the restriction enzymes being mapped. Since the site at the end of the clone insert does not correspond to a site of any restriction enzyme being mapped, there is always a **partial fragment** at the end of each clone insert which remains attached to the λ arm as the clone is digested with a restriction enzyme being mapped. The λ arms (with the partial fragment attached) are large, and are thus easily identified during subsequent processing. Only the **complete fragments** (i.e., those for which there are two delimiting restriction sites within the clone insert itself) are selected for inclusion in the mapping activity.

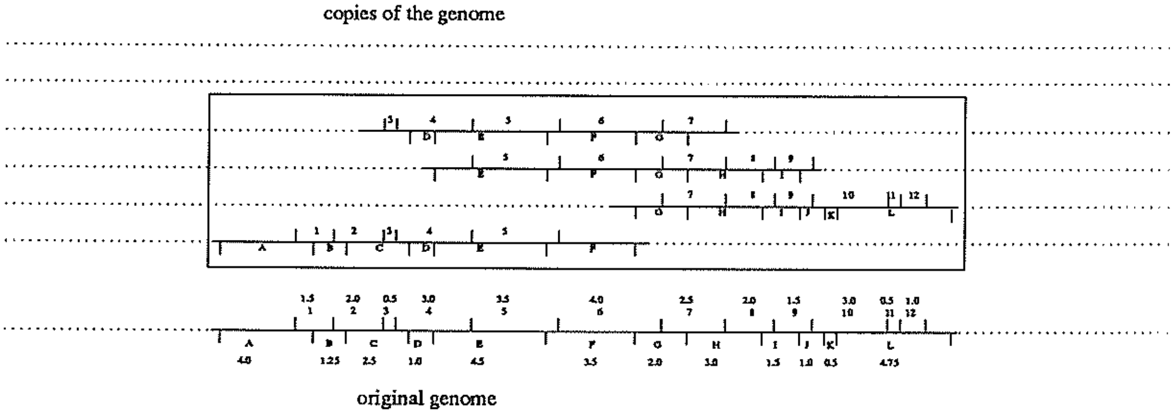
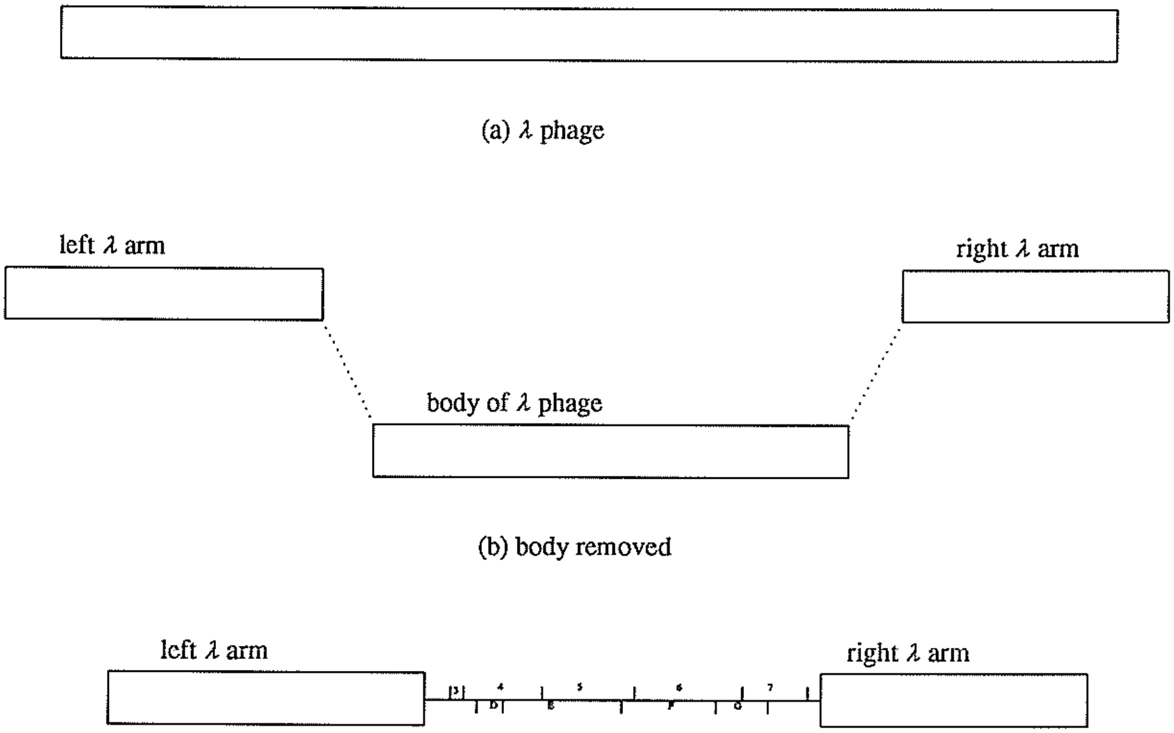


Figure 1.1: Random Clone Inserts in Context



(a) λ phage

(b) body removed

(c) clone insert replaced

Figure 1.2: Making a Clone from a Clone Insert

The size of the clone inserts is limited by the packaging mechanism of the λ phage. This size range lies roughly between 10,000 and 25,000 base pairs (bp); The combination of the λ phage and the inserted DNA is known as a **clone**, because the λ phage will be used to reproduce multiple copies of the clone insert.

During the creation of these initial biological data, enough independent λ clones must be created so that randomly selected clones will cover all (or almost all) of the genome. The overlapping regions of clones can be viewed as redundant copies of the underlying genome. The **redundancy** factor of a set of clones with respect to the underlying genome is the average (over all nucleotides) of the number of times the underlying genome is duplicated. The higher the redundancy factor, the higher the probability of "covering" the entire genome. A redundancy factor of between five and ten is not uncommon. This implies that any region of DNA from the original genome is likely to appear in between five to ten clones, on the average.

Since the inserts of DNA are the result of random cleavings, each insert may or may not contain some overlap with another insert from roughly the same region. This overlap may range from **partial overlap**, where each insert contains DNA besides the region of overlap, to **total overlap**, where one insert is a subsection of another. The success of DNA mapping depends on the fact that the clones contain these overlapping regions of DNA. It is this overlap which will allow the clones to be "rejoined" in the order in which they existed in the original genome.

After the clones are formed, further processing is done on them. First, the clones are separated by a multi-level dilution process, and *in vivo* DNA reproduction is employed to obtain enough DNA for subsequent processing. For each clone, the clone DNA extracted from this amplification process is completely digested by a restriction enzyme (the restriction enzyme being mapped), producing fragments of DNA called restriction fragments. The lengths of these fragments (in base pairs) are then measured using **agarose electrophoresis gel** technology. When an electric current is passed through an agarose gel in which DNA fragments have been placed, the fragments will migrate down the gel. It is easier for smaller fragments to move through the gel than it is for larger ones, so the fragments arrange themselves in order of decreasing length. This creates **lanes** of DNA fragments in which **bands** of DNA of the same length have migrated to the same position on the gel. After the gel has been stained, these bands can be detected and their positions on the gel determined. **Reference lanes**, containing DNA fragments of known length, are also present on the gel. Using the positions of the bands present in these reference lanes and the process of interpolation, it is possible to estimate the lengths of restriction fragments in the data lanes. Unfortunately, standard agarose electrophoresis technology is limited to detecting fragments in a particular size range; here the range of detectable fragments is taken to be approximately 400 bp to 15 kilobase pairs (kb). However, restriction enzymes can be chosen whose specificity assures that most of the restriction fragment-length data fall in this range.

There are (at least) two significant sources of error which create uncertainty about the data produced by electrophoresis. The first is the classical problem of measurement error. From experimental evidence[14], it is known that the measured lengths of the same fragment measured multiple times (say as it occurs in different clone inserts) are normally distributed about the true length of the fragment. This normal distribution is often characterized by giving an error window (a percentage difference around the true length) into which almost all measured lengths of the fragment will fall. Under some circumstances, it is possible to obtain a 3% error window around the true length of the fragment, 1.5% on either side of the actual length. Thus, a fragment which is actually 1000 bp in length may be measured as anywhere from 985 bp to 1015 bp. The second deals with determining the multiplicity of different genomic fragments of similar length; these are referred to as comigrating fragments. Two fragments of identical (or nearly identical) length will comigrate to the "same" location on the gel. Thus, it is possible for two (or more) fragments to be in the same band when the gel is stained. If this is not taken into account, the set of fragment lengths will not accurately reflect the number of fragments present in the clone. It is possible but difficult to identify this situation reliably. The intensity of the stained DNA bands should decrease along the expanse of the gel, due to the fact that there is less DNA material to stain in smaller fragments. Deviation from this expected intensity distribution can be used to estimate the number of multiple restriction fragments present in a band.

1.4. Definitions

In this section an example genome is presented. A number of important definitions are given, and data from the example genome are used to clarify and illustrate their meanings.

1.4.1. An Example Genome

For simplicity, all examples will be based on one DNA sample and set of accompanying clones. This hypothetical example is represented in Figure 1.3. A DNA segment is shown, along with a number of random clones which have been derived from the original DNA sample. In this example, the sites of two restriction enzymes, named α and β, are represented as "ticks" along the sample DNA. In this presentation, α sites are represented as ticks *above* the base line and β sites are represented as ticks *below* the base line. Note that α fragments are labeled using numbers and β sites are labeled using capital letters. Details about the lengths of the fragments (and possible confusion between them) are presented in Table 1.1. The specific composition of the clones is presented in Table 1.2. (The symbol "≈" indicates possible fragment confusion.) Apparent fragment overlap between the clones C1, C2, C3 and C4 is shown in Table 1.3. Various examples will rely on different aspects of the data present in this example genome.

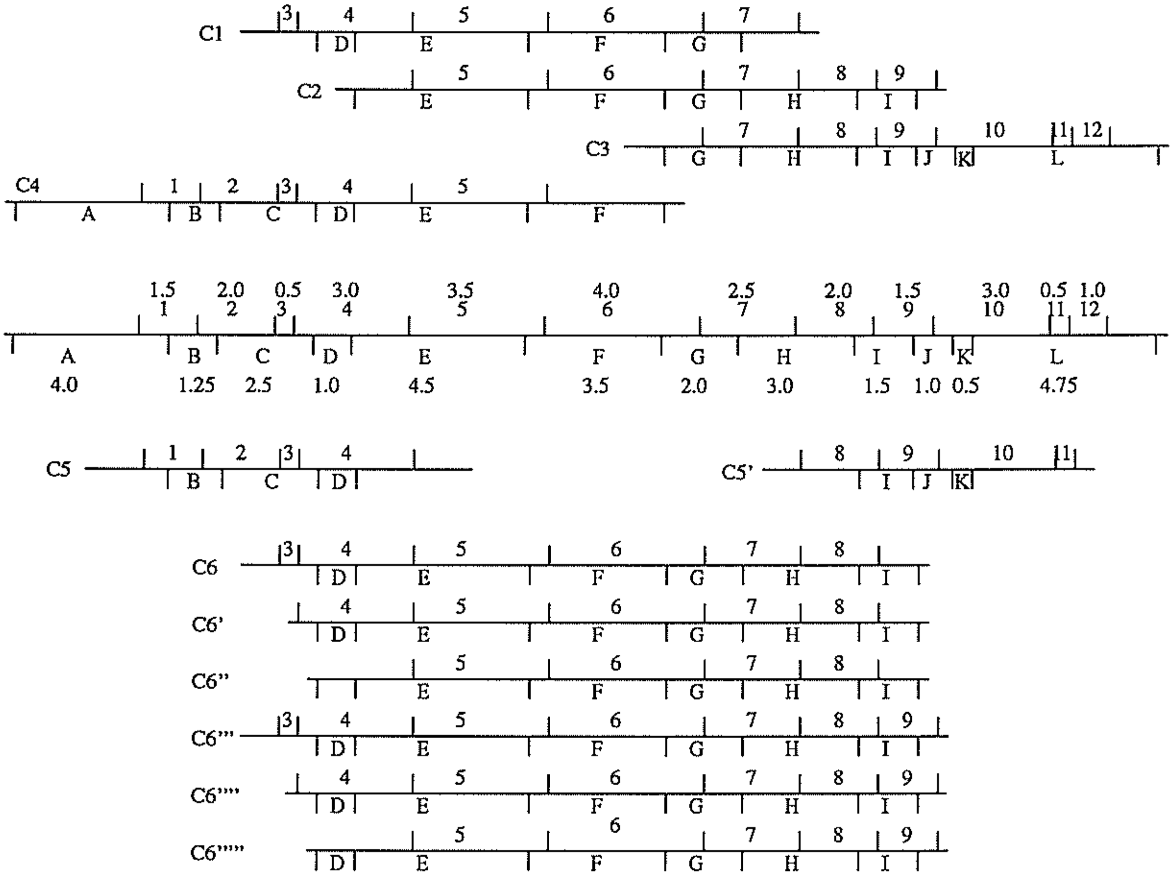


Figure 1.3: Example Genome and Random Clones

Table 1.1
Fragments from the Example Genome

Underlying Reality about Fragments		
	Fragment Name	Fragment Length (kb)
α	1(≈9)	1.5
	2(≈8)	2.0
	3(≈11)	0.5
	4(≈10)	3.0
	5	3.5
	6	4.0
	7	2.5
	8(≈2)	2.0
	9(≈1)	1.5
	10(≈4)	3.0
	11(≈3)	0.5
	12	1.0
	Fragment Name	Fragment Length (kb)
β	A	4.0
	B	1.25
	C	2.5
	D(≈J)	1.0
	E	4.5
	F	3.5
	G	2.0
	H	3.0
	I	1.5
	J(≈D)	1.0
	K	0.5
	L	4.75

Table 1.2
Clones from the Example Genome

Underlying Reality about Clones		
Clone	Fragment-length Data	
	α	β
C1	3(≈11), 4(≈10), 5, 6, 7	D(≈J), E, F, G
C2	5, 6, 7, 8(≈2), 9(≈1)	E, F, G, H, I
C3	7, 8(≈2), 9(≈1), 10(≈4), 11(≈3), 12	G, H, I, J(≈D), K, L
C4	1(≈9), 2(≈8), 3(≈11), 4(≈10), 5	A, B, C, D(≈J), E, F
C5	1(≈9), 2(≈8), 3(≈11), 4(≈10)	B, C, D(≈J)
C5′	8(≈2), 9(≈1), 10(≈4), 11(≈3)	I, J(≈D), K
C6	3(≈11), 4(≈10), 5, 6, 7, 8(≈2)	D(≈J), E, F, G, H, I
C6′	4(≈10), 5, 6, 7, 8(≈2)	D(≈J), E, F, G, H, I
C6″	5, 6, 7, 8(≈2)	D(≈J), E, F, G, H, I
C6‴	3(≈11), 4(≈10), 5, 6, 7, 8(≈2), 9(≈1)	D(≈J), E, F, G, H, I
C6′′′′	4(≈10), 5, 6, 7, 8(≈2), 9(≈1)	D(≈J), E, F, G, H, I
C6′′′′′	5, 6, 7, 8(≈2), 9(≈1)	D(≈J), E, F, G, H, I

Table 1.3
Apparent Clone Overlap in the Example Genome

	C1		C2		C3	
	α	β	α	β	α	β
C4	3-3 4-4 5-5	D-D E-E F-F	5-5 2-8 1-9	E-E F-F	1-9 2-8 3-11 4-10	D-J
C3	7-7 10-4 11-2	G-G D-J	7-7 8-8 9-9	G-G H-H I-I		
C2	5-5 6-6 7-7	E-E F-F G-G				

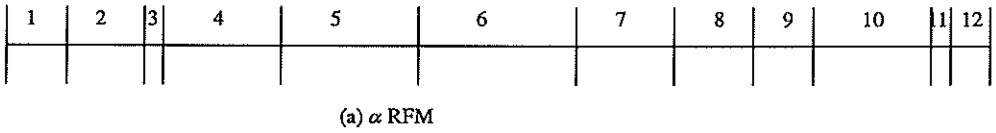
In this example, the data are assumed to be perfect. Specifically, there are no missing fragments, and for simplicity of presentation there is no measurement error. Although experimental error is of paramount pragmatic importance, it is not important to the exposition of the basic concepts presented here.

1.4.2. Maps

There are several different kinds of maps that can be constructed, given random clones extracted from a genome. Several of these are discussed here. They include: restriction-site maps, restriction-fragment maps, clone-end maps, and compositions of these.

1.4.2.1. Restriction-Site Maps and Restriction-Fragment Maps

Given a restriction enzyme whose sites are to be mapped, the objective is to determine the sequence of restriction sites along the genome, i.e., their placement in terms of base pairs within the genome. If all the restriction fragments produced by a complete digestion of the genome could be totally ordered, and their lengths precisely determined, then the map could easily be constructed. Thus, a Restriction-Site Map (RSM) can be characterized as a sequence of integers, each of which expresses the distance (in base pairs) from one restriction site to the previous restriction site. A Restriction-Fragment Map (RFM) will be constructed in lieu of an RSM. An RFM is similar to an RSM, except that the positions of the restriction fragments are determined instead of those of the sites. Given a total ordering on the restriction fragments (along with their lengths), the RFM can be expressed as the sequence of the fragment lengths, as defined by the total ordering. A common mechanism for expressing the map is graphical, in which line segments (whose lengths are proportional to the fragment lengths) are concatenated with one another, separated by vertical bars. Figure 1.4(a) shows an example of this notation for the α RFM of the genome shown in Figure 1.3; Figure 1.4(b) shows the corresponding β RFM. In general, it may not be possible to reconstruct the completely refined RFM, given the fingerprint data. It may only be possible to determine a partial ordering of the restriction fragments in which the relative order of groups of fragments is known, but the order of the fragments within the group is not known. Such a partial ordering constitutes a Partially-Ordered Restriction-Fragment Map (PORFM). Figure 1.5(a) shows the most refined α PORFM that can be produced given the fingerprint data for clones C1 through C4 of Figure 1.3; Figure 1.5(b) shows the corresponding β PORFM. Here, the vertical bars delimit the groups of fragments, and the tick marks inside a group indicate fragment boundaries if multiple fragments are present. The order of the groups is known, but the order of the fragments within the groups is not known. Of course, a clone by itself is a PORFM; it is very unstructured, consisting of a single group of fragments about which no order information is known.



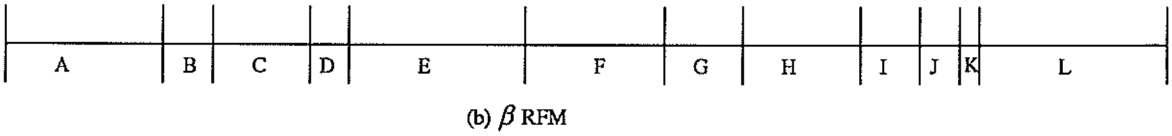
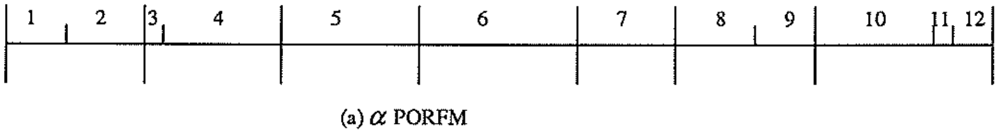


Figure 1.4: α and β RFMs for the Example Genome

A completely refined RFM can be produced if enough clones can be incorporated so that there exists a clone end between every pair of adjacent restriction sites.

Besides this graphical notation for representing PORFMs, two other notations are of interest. The first notation is the **sequence-set** notation. In this notation a *sequence* of objects is represented by placing the objects between square brackets ([ ]) (in the order that they occur in the sequence) and separating the objects with commas. A *set* of objects is represented by placing the the objects between set brackets ({ }) (in any order) and separating the objects with commas. These two notations are mutually recursive, and can be "mixed" in any way to any level of inclusion. For example the α PORFM of Figure 1.5(a) can be denoted by [{1,2}, {3,4}, {5}, {6}, {7}, {8,9}, {10,11,12}]; the β PORFM of Figure 1.5(b) can be denoted by [{A,B,C}, {D}, {E,F}, {G}, {H,I}, {J,K,L}].

The second notation is the **sequence-set-tree** (SST) notation[15] This is a graphical tree notation which is conceptually identical to the sequence-set notation. Here, a **sequence-node** denotes sequence information, and a **set-node** denotes set information. The order of the children of a sequence-node implies the sequence of the corresponding object, whereas the order of the children of a set-node implies no knowledge of their order. For example the α PORFM of Figure 1.5(a) can be denoted by the SST shown in Figure 1.6(a); the β PORFM of Figure 1.5(b) can be denoted by the SST shown in Figure 1.6(b).



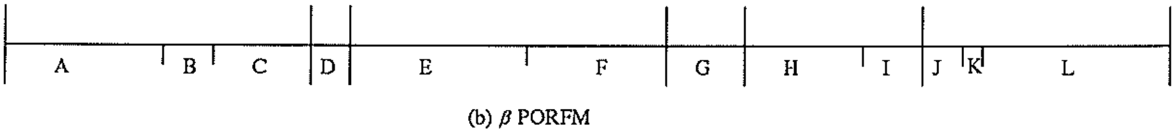
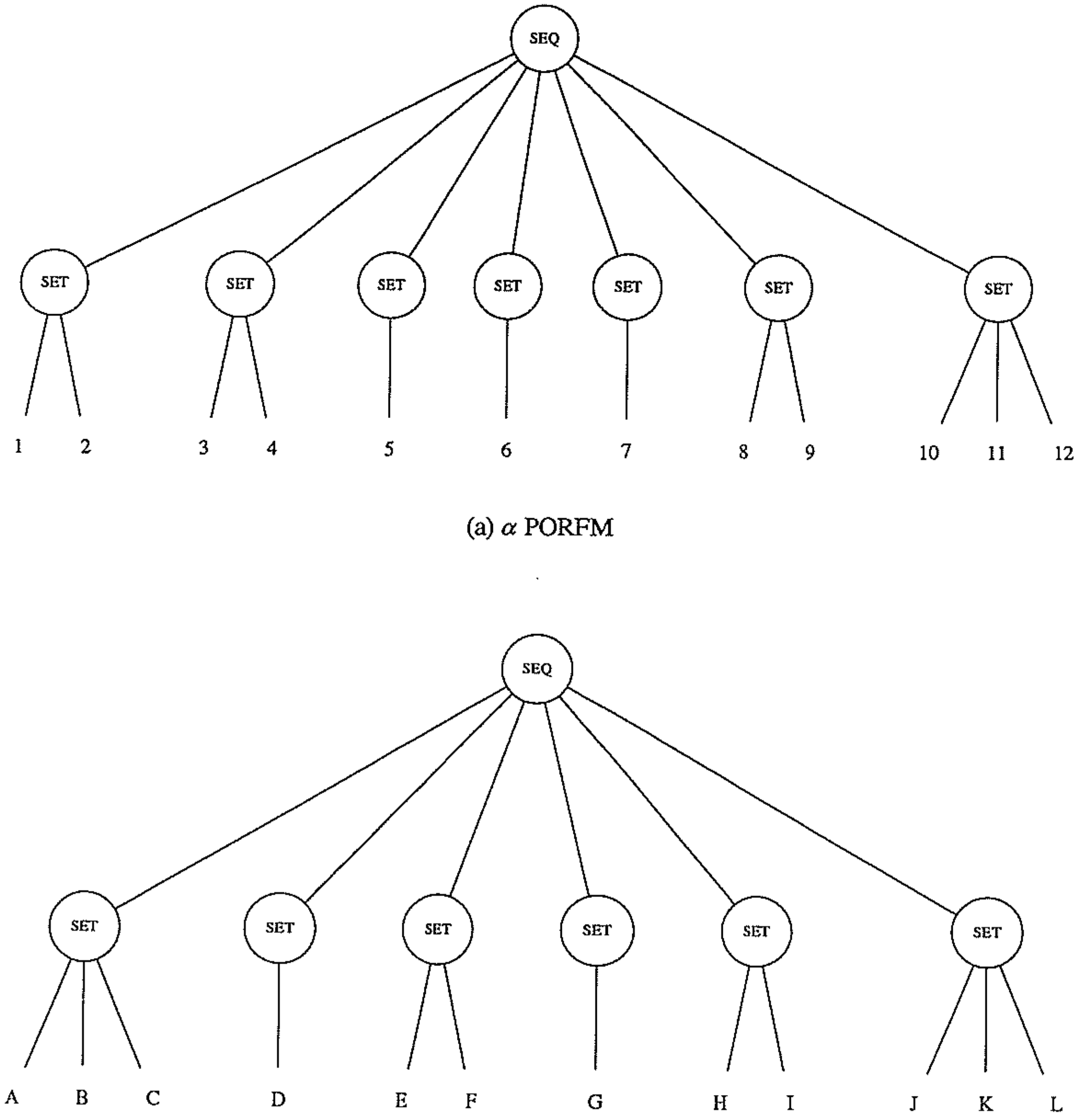


Figure 1.5: α and β PORFMs for the Example Genome



(a) α PORFM

(b) β PORFM

Figure 1.6: α and β SSTs of the PORFMs for the Example Genome

1.4.2.2. Clone-End Maps

In a similar way that restriction sites can be delineated along the expanse of the genome in an RSM, the clone ends (which are also sites) can be mapped. Here, the term **clone end** refers to the *site* at the end of a clone insert. It is *not* the partial fragment which remains attached to the λ-arm (one of two DNA components of the λ phage used to reproduce the clone insert) as the clone is completely digested with one of the restriction enzymes being mapped; it is instead the site at which the clone insert and the λ-arm are joined. (The best analogy is that of a *point* and a *line segment*, as in Geometry. A restriction fragment corresponds to a line segment; a restriction site and a clone end correspond to a point.) A Clone-End Map (CEM) is the sequence of left and right clone ends as they occur along the expanse of the underlying genome. Notice that the CEM does not give the exact position (in base pairs) along the genome; it only gives the relative position of the clone-end sites along the genome. Given the usual clone fingerprint data, in which the restriction enzyme used to produce the clones by partial digestion is not any of the restriction enzymes being mapped, it is not possible to determine the exact placement of the clone ends along the genome. For instance, using the sequence-set notation, the α CEM for the genome in Figure 1.3 is given by [{L_{C4}}, {L_{C1}}, {L_{C2}}, {L_{C3}}, {R_{C4}}, {R_{C1}}, {R_{C2}}, {R_{C3}}]; the β CEM is, as it must be, identical. This can, of course, be transformed to the SST notation. Here, the notation L_X stands for *left clone end of clone X*, and R_Y stands for *right clone end of clone Y*. As with RFMs, the clone fingerprint data may not be sufficient to construct the total ordering. A Partially-Ordered Clone-End Map (POCEM) may be the best that can be constructed. The most refined α POCEM that can be constructed for the example genome and clones C1 through C4 of Figure 1.3 is [{L_{C4}}, {L_{C1}}, {L_{C2}}, {L_{C3}, R_{C4}}, {R_{C1}}, {R_{C2}}, {R_{C3}}]; the most refined β POCEM is the totally ordered CEM shown above.

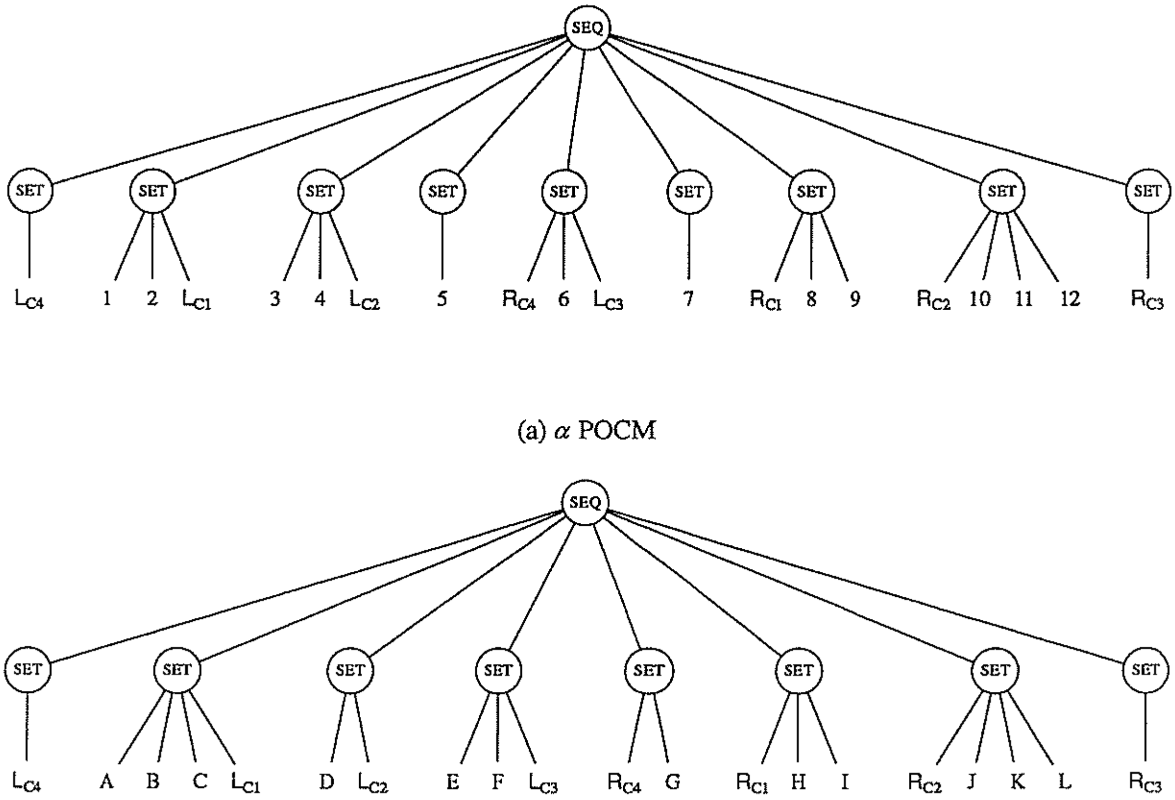
1.4.2.3. Composite PORFMs and POCEMs

It is possible to combine PORFMs and POCEMs into a single notation in which the relative order of fragments and clone ends is well-defined; this will be referred to as a Composite Map (CM and POCM). For instance, the most refined α POCM (using only α fragment-length data) that can be produced using clones C1 through C4 is [{L_{C4}}, {1,2,L_{C1}}, {3,4,L_{C2}}, {5}, {L_{C3},6,R_{C4}}, {7}, {R_{C1},8,9}, {R_{C2},10,11,12}, {R_{C3}}]; the most refined β POCM that can be produced is [{L_{C4}}, {A,B,C,L_{C1}}, {D,L_{C2}}, {E,F,L_{C3}}, {G,R_{C4}}, {R_{C1},H,I}, {R_{C2},J,K,L}, {R_{C3}}]. The corresponding SSTs are shown in Figure 1.7. Note that with the introduction of clone end information into the map, the graphical form of maps becomes less effective, because there is no "natural" notation for combining fragment information and site information. However, the sequence-set and SST notations are abstract enough that such a combination can easily be expressed. The SST for an unstructured POCM (the α digestion of Clone C1) is shown in Figure 1.8.

Note that the clone ends and fragments have an order relation with respect to each other; clone ends and fragments can co-exist within sets or set-nodes. The analogy of a fragment to a line segment and a clone end to a point is very useful and appropriate here. In fact, the actual clone end resides in the interior of one of the fragments present (assuming that there are no undetected fragments). More specifically, if a left clone end is present in a set (or set-node) with fragments, then it resides in the interior of the rightmost fragment present in the set. For instance, consider the third set-node from the left in Figure 1.7(a). There are three objects present: fragment 3, fragment 4, and L_{C2}. L_{C2} resides on either fragment 3 or fragment 4, whichever is the rightmost fragment in the underlying genome. Similarly if a right clone end is present in a set (or set-node) with fragments, then it resides in the interior of the leftmost fragment present in the set.

1.4.3. Incorporating PORFMs and POCMs

In DNA mapping, the ultimate goal is to uncover the *actual* sequence of the restriction fragments in the original genome. For various reasons it may not be possible to reconstruct the actual ordering of the fragments present in the original genome, given only the fingerprint fragment-length data of the clones. For instance, there may be missing undetectable fragments (because they are too short or too long for electrophoresis technology to detect) or there may be regions of the genome which are unclonable. There may be regions of the genome in which a number of restriction sites are distributed in such a way that there is no set of clones with clone ends which can separate the restriction fragments; thus, a completely refined RFM will be impossible to construct, and a partially refined PORFM must suffice. It may also be that the fingerprint data are consistent with many different underlying genome configurations containing different distributions of restriction sites. Some of these alternate configurations may be more compact that the one from which the fingerprint data actually came. Thus, a map constructed by even the most exhaustive and



(a) α POCM

(b) β POCM

Figure 1.7: α and β SST POCMs for the Example Genome

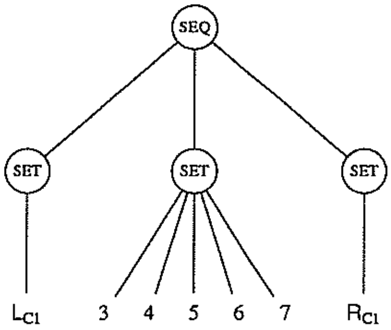


Figure 1.8: α SST POCM for Clone C1

meticulous algorithmic approaches may still produce a map which is different from the underlying reality.

Knowledge about what restriction fragments are present in the genome is distributed throughout the fingerprint data of the clones. Two fragments (in different clones) will be referred to as **identifiable** with one another if they are equal in length within measurement error bounds. Any specific restriction fragment from the underlying genome (i.e., a **genomic** fragment) will be represented by an instance of a restriction fragment in each clone (referred to as a **real** fragment) which overlaps the same region of the genome. Thus, two real fragments of identical length (within measurement error bounds) in different clones are potentially instances of the same genomic fragment, and thus should be identifiable with one another. Unfortunately, all fragments of the same length (within measurement error bounds) do not represent instances of the same genomic fragment. The choice as to which fragments actually will be *declared* to represent the same genomic fragment is determined during the process of building the PORFM. When two fragments are declared to represent the same underlying genomic fragment they will be referred to as being **paired** with one another.

Given these observations, the objective is the following: Find as compact an "overlaid" positioning of the clones as possible in which (a) the fragments within each clone can occur in any order, (b) the fragments of each clone remain contiguous with each other, and (c) paired fragments in all the clones which have overlap align with one another. The general approach to discovering the desired PORFM is to attempt to put clones together incrementally while complying with the above constraints. The term **incorporate** will be used to denote the process of taking two PORFMs and "putting them together" to form a third, potentially more refined, PORFM. At any point in the overall mapping process, the two PORFMs being incorporated might be two clones, two structured PORFMs, or a clone and a structured PORFM. Many different strategies can be employed; one will be presented in Section 2.

Restriction fragments in the original clones will be referred to as **real fragments** because they represent a specific instance of a restriction fragment from the underlying genome and have an *actual* measured length. As fragments from different clones are combined (by pairing during incorporation) to represent an inferred fragment from the underlying genome, the resulting fragment will be referred to as a **virtual fragment**. A virtual fragment has no actual measured length; its length will be taken to be the average of the measured lengths of the real fragments which compose it. Thus, many of the fragments in a structured PORFM will be virtual fragments. Even the real fragments present in a PORFM can be considered to be virtual fragments composed of only one real fragment.

1.4.4. The Working Assumption

In general, two PORFMs will be allowed to incorporate if there is sufficient *apparent* evidence that there is significant overlap between them. One approach for declaring that sufficient evidence is present is based on the fundamental working assumption that fragments which are identifiable with one another correspond to the same genomic fragment. Of course, this working assumption is often false. This working assumption is based on a combination of the converses of the following two theorems; neither of the converses are true.

Theorem 1:

If two instances of restriction fragments in two different clones are instances of the same genomic fragment, then they have the same sequence.

The converse is not true. Just because two fragments have the same sequence does not imply that they are instances of the same genomic fragment.

Theorem 2:

If two instances of fragments have the same sequence, then they have the same measured length (within measurement error bounds).

The converse is not true. Just because two fragments have the same measured length does not imply that they have the same sequence.

Although this fundamental working assumption has a significant probability of being wrong for individual fragments, the joint probability of being wrong is significantly reduced by requiring multiple fragments within the perceived region of overlap to be identifiable with one another. Often, requiring some specific number of fragments to be present in the *apparent* overlap increases the probability that the joint working assumption is true to a sufficient extent that it is appropriate to declare the presence of significant *actual* overlap (even though the actual overlap may be less than the apparent overlap).

1.5. The Process of DNA Mapping

1.5.1. Mapping Two Clones Together

The reason that clone data can be used to create a map of a genome is the fact that fragments which come from a single clone must be contiguous in the original DNA sequence. Given fragment-length data for just one clone, it is impossible to know the ordering of the fragments within it; it is only known that they *are* contiguous in a certain region of the original DNA. A more refined view of that region can be created by considering other clones which are suspected to overlap the same region. Consider the α data for C1 and C2 from the example genome. C1 has fragment-length data {4000, 3500, 3000, 2500, 500}, and C2 has fragment-length data {4000, 3500, 2500, 2000, 1500}. Since these two clones appear to share three fragments of the same lengths (4000, 3000, and 2500), it may be considered probable that they are partially overlapping clones from the same general region of the original DNA. However, it is impossible to determine how they actually overlap without doing more biological work. Simply because they contain three fragments having the same length is no guarantee that they *actually* overlap, since two fragments of the same (apparent) length do not necessarily correspond to the same genomic fragment. One of the ways that this is taken into account while mapping is to require multiple (apparent) fragment overlap before assuming an actual overlap is present. Often, given the type of data presented here (i.e., fragment-length data for λ clones) the minimum number of fragments which must seem to overlap (before actual overlap is inferred) is taken to be 4 or 5; here, the value is taken to be 3 to reduce the amount of data required to expose the concepts. The probability that two clones actually share some region of the underlying genome increases as the number of fragments of apparent overlap increases.

Returning to the example, it is known that the five α fragments in C1 must be contiguous; similarly, the five α fragments of C2 must be contiguous. This is all that can be determined from examining the clones independently of each other. However, more information can be extracted by examining the two clones in concert.

There appear to be three genomic fragments in common. Assuming the postulate that these three fragments should be paired is accepted, these three fragments must also be contiguous. This means that each clone can be divided into two sets, one set containing the fragments which apparently overlap and the other set containing all the remaining fragments in the clone. In C1, these two sets are {4000, 3500, 2500} and {3000, 500}, while in C2 they are {4000, 3500, 2500} and {2000, 1500}. Since each of the two clones contains an overlapping region with the other clone, it is possible to "fit" the two back together into one partial sequence. This sequence is shown in Figure 1.9 and corresponds to the PORFM shown in Figure 1.10. In Figure 1.10, the fragments are displayed in the order that they occur in the underlying example genome, in order to correlate the PORFM with the underlying reality; this ordering is not actually available during the mapping process, so the fragments in each group are usually sorted in descending order, as depicted in Figure 1.9.

The ordering in Figure 1.9 contains more information than either of the original two clones provided. Specifically, it is now known that there is a restriction site a distance of 3500 (3000+500) bp in from the first restriction site occurring at the left end of C1. (The concepts of "left" and "right" are, of course, *arbitrary* orientations which come from the human-oriented notations used to present the structure of the data. Given any map, the reflection, exchanging "left" with "right", is just as valid. The important relationship

```
{3000, 500} {4000, 3500, 2500} {2000, 1500}
|----------------------------|
             C1

           |------------------------------|
                        C2
```

Figure 1.9: Two Overlapping Clones

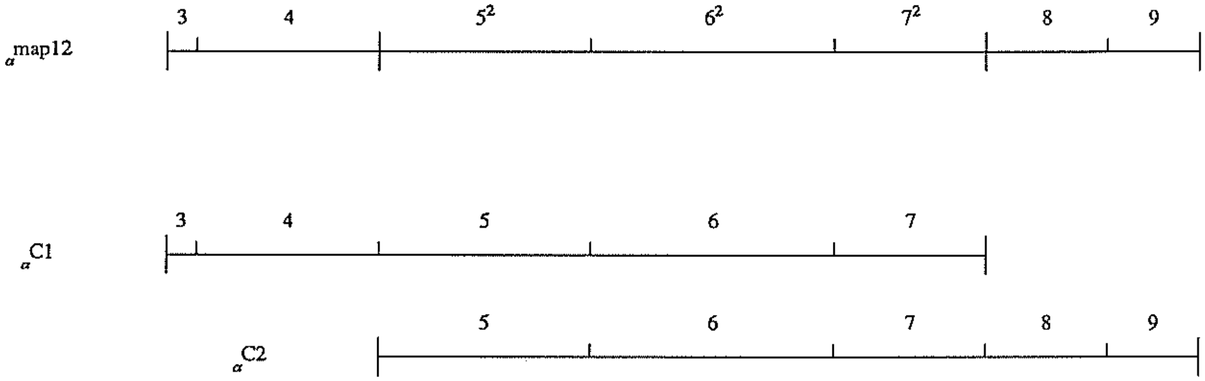


Figure 1.10: PORFM for C1 and C2

here is the partial ordering of the fragments, not the orientation.) Similarly, there is a restriction site a distance of 3500 (2000+1500) bp in from the first restriction site occurring at the right end of C2. The information about this particular region of the genome is still relatively unrefined. It is known that there are three sets of fragments, with two fragments in the first set, three fragments in the second set, and two fragments in the last set. These subdivisions, or sets, will be referred to as **groups** in the abstract PORFM. It is known how the three groups are positioned in relation to each other. However, the exact order of the fragments in any one of the groups is not known.

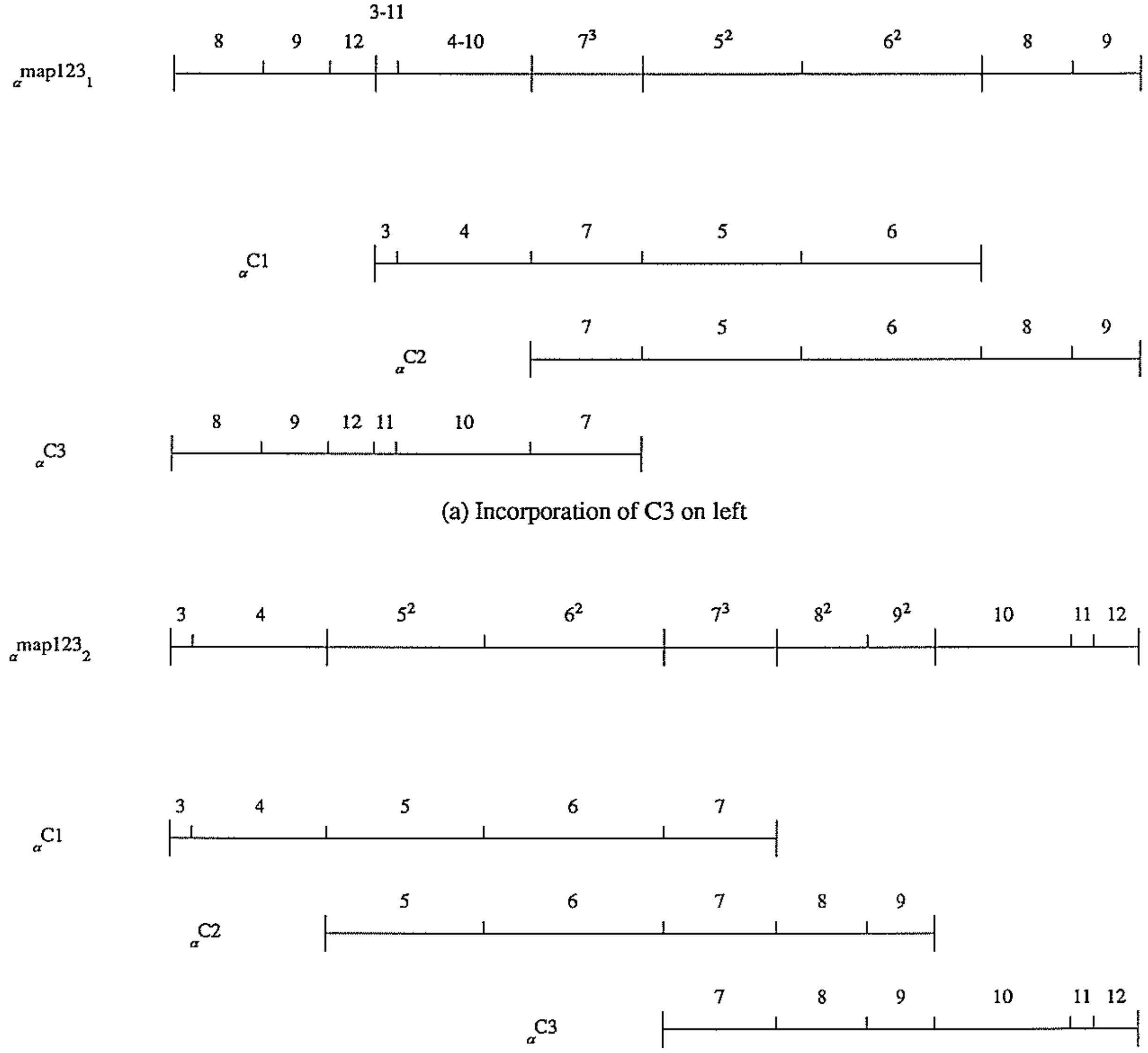
1.5.2. Mapping a Set of Clones

Given a set of clones, perceived to come from a contiguous region of the underlying genome, it is possible to continue to refine the positioning of the fragments by incorporating more clones into the PORFM. New clones can be incorporated into a PORFM in two different ways: by **extension** or by **assimilation**. An **extension** occurs if the number of fragments in the resulting PORFM is greater than the number of fragments in the previous PORFM, i.e., some fragment of the clone extends beyond the boundaries of the original PORFM. An **assimilation** occurs if the number of fragments in the resulting PORFM is equal to the number of fragments in the previous PORFM, i.e., every fragment in the clone pairs with an already existing fragment in the original PORFM.

It might be possible to incorporate a clone into a PORFM in more than one position. Such a situation is referred to as **ambiguous**. Ambiguous incorporations are, in general, not allowed; they must be resolved (by some external means) or be deferred until more structure is available (which might eliminate the ambiguity).

Consider attempting to incorporate C3 into the PORFM in Figure 1.10. Unfortunately, C3 can be incorporated in two different positions, on the left (as shown in Figure 1.11(a)) or on the right (as shown in Figure 1.11(b)). Note that in Figure 1.11(a) it is necessary to reorder (i.e., change the order from that in the underlying genome) the fragments in each of the three clones in order for the incorporation to take place.

Here, assume that Figure 1.11(a) is discarded by some external mechanism and Figure 1.11(b) is selected as the appropriate PORFM. Then C4 can be incorporated in only one way, as shown in Figure 1.12. Note that Figure 1.12 represents the same underlying reality as that shown in Figure 1.5(a).



(a) Incorporation of C3 on left

(b) Incorporation of C3 on right

Figure 1.11: PORFMs for C1, C2, and C3

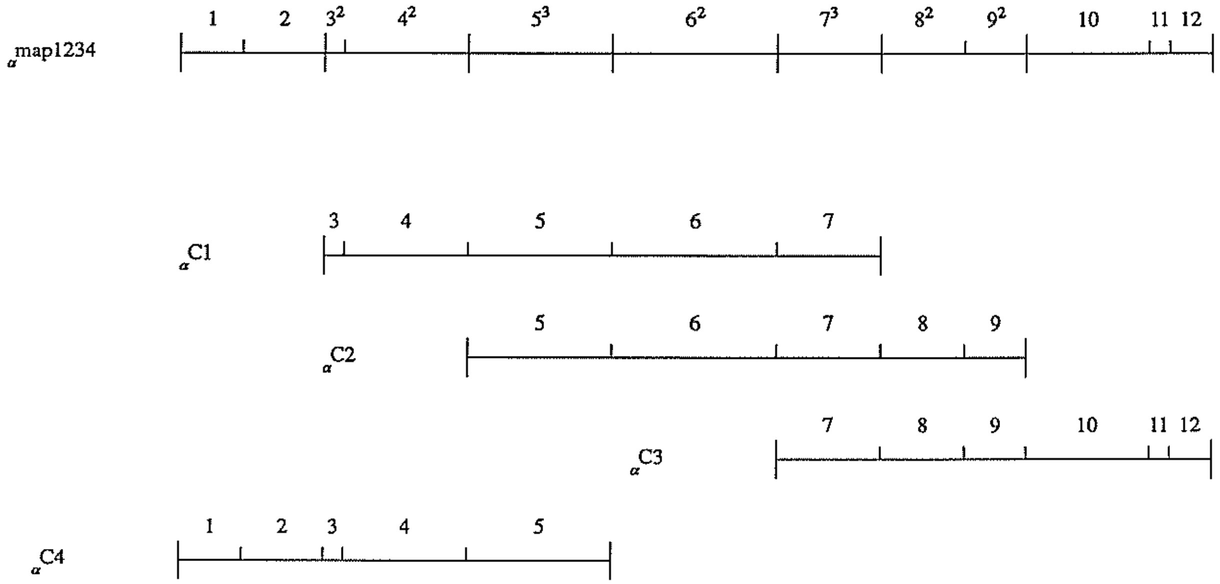


Figure 1.12: PORFM for C1, C2, C3, and C4

There are several types of ambiguity. **Global ambiguity** occurs when it is possible to incorporate a clone into a PORFM in distinctly different regions of the PORFM (i.e., regions which have no overlap). **Local ambiguity** occurs when the clone can be incorporated in the same general region in a number of different ways (i.e., the regions of incorporation have non-null overlap).

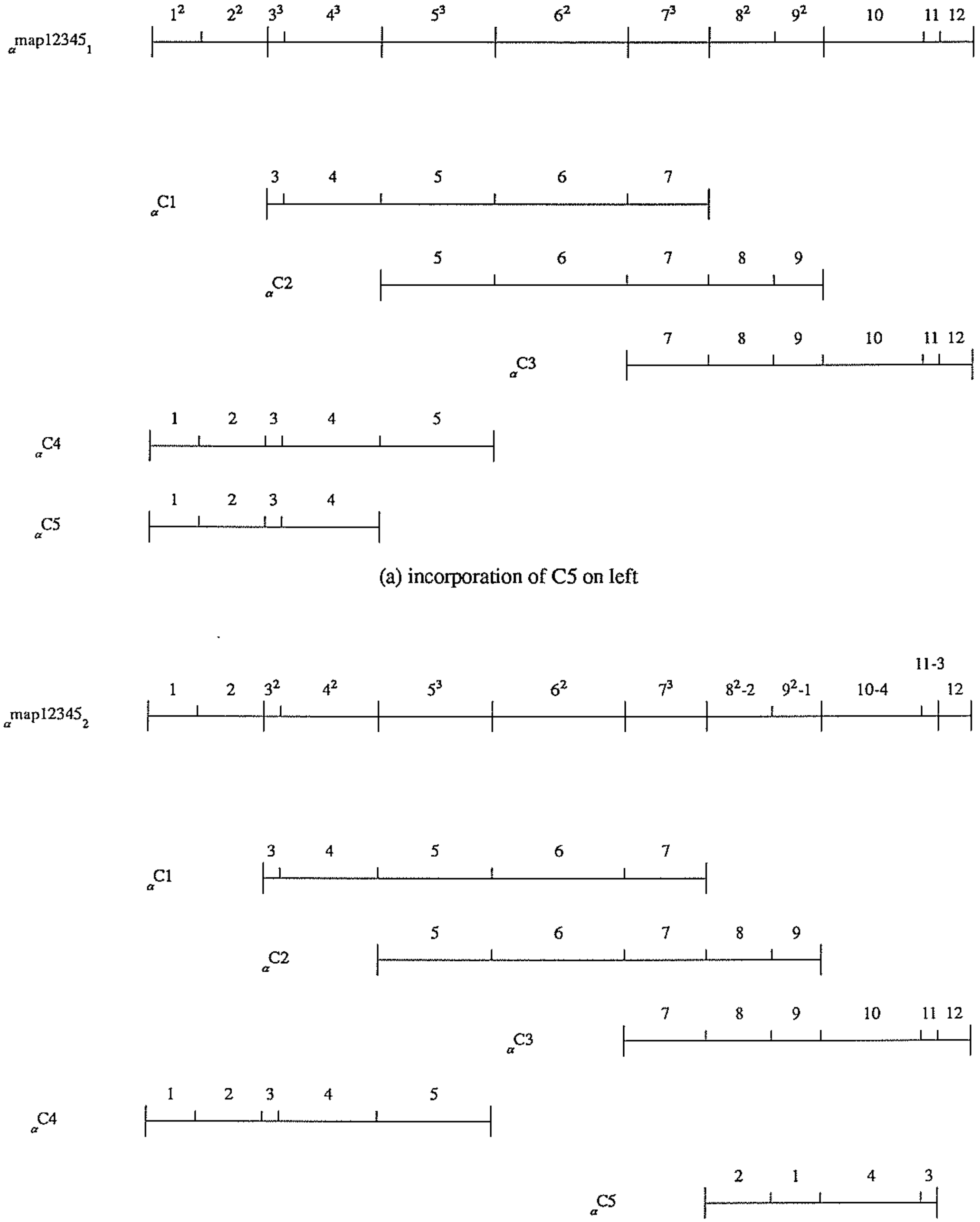
As an example of global ambiguity, consider the incorporation of C5 into the PORFM in Figure 1.12. C5 can be incorporated in two distinctly different regions, as shown in Figure 1.13.

As an example of local ambiguity, consider the incorporation of C6 into the PORFM in Figure 1.12. C6 can be incorporated into the same general region in two different positions, as shown in Figure 1.14; note that the regions of incorporation have significant overlap. This often occurs when there are two different genomic fragment of roughly the same size, approximately one clone length apart; in this case these are fragments 2 and 8, each of length 2.0 kb.

1.6. Puzzle Construction

The process of performing DNA restriction mapping can be likened to that of putting together a large jigsaw puzzle. In a jigsaw puzzle the pieces usually have a variety of shapes (which help determine their placement) and a pattern on the top surface (which helps to discriminate between pieces with "similar" shape). Every piece of the puzzle is present once and only once, and there is usually a composite picture created by the aggregate of the patterns once the pieces have been put together completely. A variety of heuristic techniques can be used to sort the pieces into collections which have higher than random probability of having close proximity. Selecting pieces with similar color is one such technique; separating pieces which have at least one straight edge (straight edges occur on the boundary of the puzzle) is another.

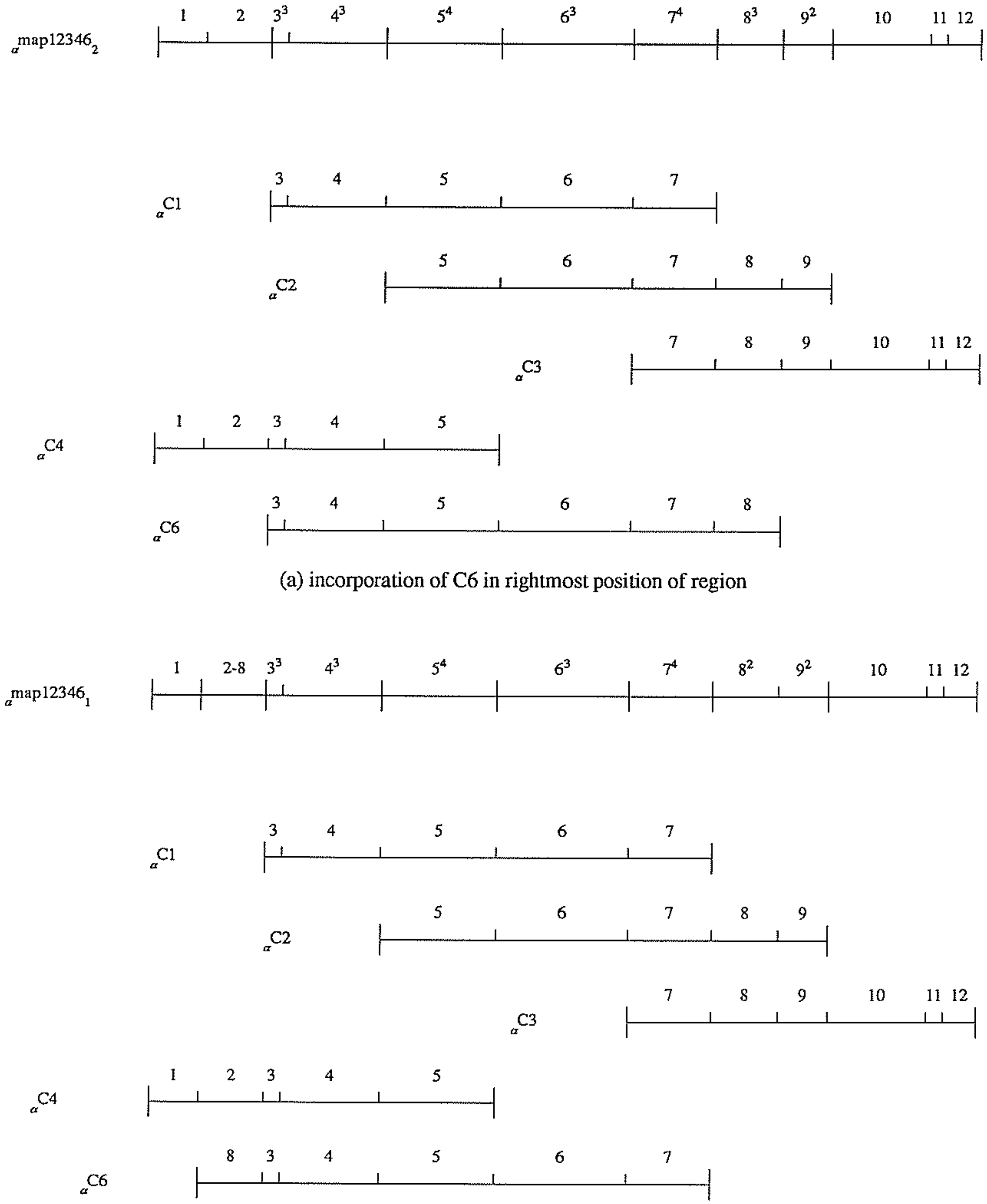
In DNA mapping, the jigsaw pieces correspond to the clones. Their shape is not distinctive (which might be considered analogous to jigsaw pieces in which all shapes are constructed from straight lines), but there is a pattern associated with each piece, the fragment-length data (which might be considered



(a) incorporation of C5 on left

(b) incorporation of C5 on right

Figure 1.13: Global Ambiguity



(a) incorporation of C6 in rightmost position of region

(b) incorporation of C6 in leftmost position of region

Figure 1.14: Local Ambiguity

analogous to a random pattern, such as popcorn). A piece may be missing completely, present multiple times, or partially overlap with other pieces. (In fact this third situation predominates and is what allows the map to be assembled.) No information (or at least little information) is available *a priori* about how the final map should look. Contig building (the process of identifying sets of clones which statistically have a high probability of coming from a common *contiguous* region of the genome) is often used as a heuristic for subdividing clones into collections which have higher than random probability of having close proximity.

In DNA mapping, the objective is to find as compact an alignment of the clones as possible such that, as fragments within each clone are allowed to rearrange in any order, the fragments of each clone remain contiguous with each other while simultaneously pairing with identifiable fragments in all the aligned clones.

Given a jigsaw puzzle whose surface pattern is relatively random (say something like popcorn) and whose pieces do not have distinctive shapes (because all their edges are straight), there is a high probability of misplacing a piece during the incremental construction of the jigsaw puzzle. If a piece "seems to fit" at a specific position (because the surrounding pattern in the puzzle as constructed so far roughly matches that on the piece being considered) but, in fact, belongs somewhere else, the piece may inappropriately be placed at this wrong position. Construction of the puzzle may continue for quite some time before the inconsistency becomes apparent. However, if the piece truly has been placed in the wrong position, the puzzle cannot be correctly completed until this error has been identified and corrected.

In DNA mapping, similar pitfalls occur with the placement of clones. If a clone is placed in the wrong position in the map, it may not be possible to complete the map.

In the "end game" of constructing a jigsaw puzzle, it is often easy (a) to determine that a piece has been placed incorrectly and (b) to determine what piece is in the incorrect position. However, in DNA mapping these determinations are not so obvious. Backtracking is sometimes attempted as a remedy for this problem. However, the computational complexity of this solution makes it completely unacceptable, especially since the erroneous placement often occurs very early in the construction of the map. Instead, a mechanism for reducing the uncertainty of placing a clone in a map is what is truly needed. MRE mapping supplies just such a mechanism.

Given the previous analogy to a jigsaw puzzle, whose picture is a random pattern and whose pieces have nondistinct shape, what could be done to reduce the possibility of placing a piece in the wrong position? The jigsaw puzzle actually has two surfaces, a top surface and a bottom surface. If another random pattern (independent from the first) were placed on the bottom surface, this second pattern could be used to verify the placement of any specific piece. Specifically, if a given piece seems to fit at a specific position because the pattern on the top surface matches the surrounding pattern, but it fails to match the surrounding pattern on the bottom surface, then this piece clearly does not belong at this specific position.

MRE mapping constitutes the DNA mapping counterpart of the two-sided jigsaw puzzle. Assume that each clone has been digested with two independent restriction enzymes, say α and β, and that as mapping proceeds, two maps are constructed in parallel, one using the α fragment-length data and the other using the β fragment-length data. Assume further that each of these maps contains exactly the same clones, and the clones occur in "the same positions". As a new clone is considered for insertion into the maps, one of two maps is selected as a primary search domain. If a clone seems to incorporate at a specific position in the α map (based on the α fragment-length data), it may be (but not necessarily will be) merged into the compound POCM only if it also incorporates into "the same position" in the β map (based on the β fragment-length data). The resolution of whether two clone are in "the same position" can be achieved by synchronizing the two maps with respect to the clone ends present prior to attempting to incorporate the new clone. This is valid because the clone ends occur at specific sites along the original underlying DNA. The juxtaposition of these sites, although originally unknown, is fixed and independent of any specific restriction enzyme fragment-length data. In fact, the juxtaposition of these clone end sites is discovered during the process of mapping.

Unlike a jigsaw puzzle, which can only use two surfaces to help reduce the uncertainty of piece placement, it is possible to use virtually any number of restriction enzymes to help reduce the uncertainty of clone placement. A probabilistic analysis will be presented (in Section 4) showing how the uncertainty of clone placement, jointly consistent across multiple restriction enzymes, is related to the uncertainty of clone placement for each restriction enzyme independently. The decrease in uncertainty is exponential in the number of restriction enzymes used. Thus, given reasonable estimates of the uncertainty of placing a clone using only one restriction enzyme (based on specific criteria by which clones are allowed to be incorporated into maps), it is possible to determine *a priori* how many restriction enzymes should be used in order to achieve any specific desired level of joint uncertainty.

The remainder of this report presents general approaches and algorithms for reducing the uncertainty of clone placement. Section 2 presents many of the general techniques used in SRE mapping; these include window tiling and the match/merge approach. Section 3 shown how these incorporation techniques can be extended to facilitate MRE mapping. Section 4 presents a Bayesian analysis of the probability of false positives and false negatives in MRE mapping, as a function of the corresponding probabilities in SRE mapping. Section 5 presents a number of details which were suppressed in previous sections (in order to expose the concepts without excessive detail). Section 6 takes a broader view, discussing variations and extensions to the MRE mapping technique and showing how it can be effectively applied within a number of strategies.

2. Single-Restriction-Enzyme Mapping

In general, the mapping approach uses a greedy algorithm[16] in which two POCMs (or PORFMs) are selected for an attempt at incorporation. For simplicity, the selection criterion is suppressed here, but can be based on many different properties of the POCMs involved. One POCM is incorporated into another; the **incorporation candidate** (IC) is incorporated into the **incorporation target** (IT). Although the approach works for ITs and ICs having arbitrary structure, it is probably easiest to understand the approach by perceiving the IC to be a clone (with no structure) and the IT to be a highly structured POCM containing many clones.

There are two fundamental ways that the IC can incorporate into the IT: it can assimilate within the IT, or it can extend the IT. Assimilation occurs when every fragment of the IC pairs with a fragment in the IT. Extension occurs when there is at least one fragment in the IC that fails to pair with a fragment in the IT; any unpaired fragment in the IC will constitute an extension onto the IT after incorporation has occurred.

Assimilation of the IC into the IT is always attempted first. If this succeeds, extension is not attempted. This approach is consistent with preferring POCMs which are as compact as possible.

2.1. Assimilation

In order to determine whether or not the IC assimilates into the IT, the IC must be allowed to be positioned anywhere within the internal boundaries of the IT. For each such possible positioning, it is determined whether or not there is a topologically valid set of pairings (i.e., pairings which allow the fragments of each clone to remain contiguous) between the fragments of the IC and the fragments in the region of the IT in which the IC has been positioned.

2.1.1. Window Tiling

The concept of a "window" in the IT is used to achieve this goal of considering every possible positioning of the IC within the IT. A **window** is a contiguous set of groups in a POCM (i.e., a contiguous set of set-node siblings in an SST). The window can be specified by designating the two outermost groups in the set of contiguous groups that constitute the window. A window (of size just slightly larger than the IC) is positioned within the IT, and an attempt is made to incorporate the IC within the confines of the window. (Details about how to select a window of the appropriate size are given in Section 5.2.1.1.1.) The IC either does or does not incorporate within the window; in either case, the window is "moved" to an "adjacent" position and incorporation is attempted there. (Details about how windows are "moved" across the IT are given in Section 5.2.1.1.2.) In essence, the window is "dragged" across the IT, incorporation being attempted at each discrete position. The collection of different window positions constitute a "tiling" of the IT; this "tiling" must allow the IC to attempt to incorporate in any position of the IT. If the IC can incorporate in different positions within a window, then **internal ambiguity** is present. Internal ambiguity usually implies local ambiguity, because the incorporation regions within a single window usually have some overlap (if the window is chosen appropriately). If an IC can incorporate in different positions in different windows, then **external ambiguity** is present. External ambiguity usually implies global ambiguity, because there is usually no overlap in the incorporation regions in different windows. (However, since adjacent windows do have overlap, it is possible for incorporation regions in different windows to have overlap.) The presence of any form of ambiguity causes the incorporation attempt to fail and to be deferred until later. The introduction of more structure into the IT (or IC), caused by incorporation with other POCMs, may eliminate the current ambiguity. Thus, a simple deferral strategy often is sufficient to resolve ambiguity problems.

The use of windows serves two purposes. First, windows supply a conceptual tool for organizing the computation and focusing on a specific region of interest within the IT. Second, their use tends to make the underlying computation *linear* with respect to the length of the IT, instead of *exponential*. If windows were not used, and fragments in the IC were allowed to attempt to pair with all fragments in the IT at the same time, then an exorbitant number of fragment matchings (see next section) could be created, most of which are topologically infeasible. The number of fragment matchings tends to be exponential in the number of fragments available for pairing. Since the maximum size of a clone is effectively bounded by a constant, the maximum size of a window is similarly bounded. Thus, there is effectively an upper bound on the number of fragment matchings that can be produced within a window. The number of windows in a tiling of an IT is linear in the length of the IT. Thus, the number of fragment matchings produced across the entire IT using windows is linear in the length of the IT.

When the IC being incorporated is not a clone, but is instead a structured POCM, there is no upper bound on the size of the IC. Thus, this claim of linearity on the number of fragment matchings produced does not hold. However, the use of windowing significantly reduces the number of fragment matchings produced in this situation also.

A two-phase approach is used to determine whether or not an IC incorporates into the current window of the IT. These are known as the **match** phase and the **merge** phase.

2.1.2. Match

During the match phase, all of the structure present in the IC and in the window of the IT is ignored, and the fragments in each are aggregated together to create multisets (or bags). The individual fragments in the IC are compared to the fragments in the window of the IT to determine fragments that can be identified with one another. All possible individual pairings between fragments in one set and fragments in the other set are considered. An aggregate of such pairings, which allows the use of any specific fragment no more than once, is referred to as a **fragment matching** or **fragmat**, for short. A fragmat whose cardinality is greater than or equal to the cardinality of all other possible fragmats is known as a **maximum** fragmat. All maximum fragmats are considered, and any "similar" fragmats are represented by one of the "best" fragmat among those which are similar. (For instance, if two fragments in the same group in the IC pair with a single fragment in the IT, then the two fragments in the IC are considered "indistinguishable" for this purpose, and the fragment whose length is closest to the length of the fragment in the IT is selected as the representative for pairing.) All fragmats are created ignoring any information about the topology or structure in the region from which the fragments come.

As a first example, consider attempting to assimilate the α digestion of C3 into ${}_{\alpha}$map12 (shown in Figure 1.10). Assume that the incorporation window in ${}_{\alpha}$map12 is the entire map. There is one and only one maximum fragmat possible:

$fm_1 = \{(3,1),(4,10),(7^2,7),(8,8),(9,9)\}$.

Note that fragment 12 from C3 is not represented in this fragmat, fragment 3 of ${}_{\alpha}$map12 is confused with fragment 11 of C3, and fragment 4 of ${}_{\alpha}$map12 is confused with fragment 10 of C3.

As a second example, consider attempting to assimilate the α digestion of C5 into ${}_{\alpha}$map1234, shown in Figure 1.12. Here, for illustrative purposes, assume that the incorporation window in ${}_{\alpha}$map1234 is chosen to be the entire map. (A more rational tiling is shown in Figure 5.6) Three maximum fragmats (of the 16 maximum fragmats possible) are shown here:

$fm_2 = \{(1,1),(2,2),(3^2,3),(4^2,4)\}$,
$fm_3 = \{(8^2,2),(9^2,1),(10,4),(11,3)\}$, and
$fm_4 = \{(1,1),(8^2,2),(3^2,3),(10,4)\}$.

fm_2 corresponds to the one used in Figure 1.13(a), and fm_3 corresponds to the one used in Figure 1.13(b). Note that fm_4 contains two fragment pairings which involve fragment confusion.

2.1.3. Merge

During the merge phase, the topologies of the regions are reintroduced. Each fragmat constructed during the match phase is checked, one at a time, to determine whether or not the pairings specified by the fragmat allow the existence of a topologically feasible sequence of the fragments in both POCMs, i.e., whether or not the pairings in the fragmat allow the real fragments of each clone to remain contiguous. Incorporation occurs for any fragmat which is topologically feasible. If two or more fragmats allow incorporation, then internal ambiguity is present.

Consider the two examples presented in the previous subsection. First, consider the assimilation of C3 into ${}_{\alpha}$map12. fm_1 has a cardinality of 5, one less than the cardinality of the fragments in the α digestion of C3. Since there exists a fragment in C3 (the IC),in this case fragment 11, which does not pair, the assimilation of C3 into ${}_{\alpha}$map12 is not feasible. (Even if this fragmat is considered for the purposes of extension, i.e., fragment 12 of C3 would be an extension beyond ${}_{\alpha}$map12, the fragmat is topologically infeasible, since the presence of fragments 5^2 and 6^2 of ${}_{\alpha}$map12 would not allow the fragments of C3 to remain contiguous). Thus, this fragmat is discarded. Since this is the only maximum fragmat, assimilation is declared to be impossible

Next, consider the assimilation of the α digestion of C5 into ${}_{\alpha}$map1234. fm_2 and fm_3 are topologically feasible, as depicted in Figure 1.13(a) and 1.13(b), respectively. However, fm_4 is topologically infeasible, because this fragmat does not allow the fragments of C5 to remain contiguous. Thus, fm_4 is discarded. The successful incorporations of C5 using fm_2 and fm_3 signal ambiguity; this is global ambiguity, since the regions of incorporation have no overlap.

2.2. Extension

If incorporation by assimilation was not successful, then incorporation by extension is attempted. Here, an attempt is made to use the IC to extend the IT. Excluding trivial POCMs with no structure, there are four possible ways that the IC might extend the IT: (1) the left end of the IT might incorporate with the left end of the IC, (2) the left end of the IT might incorporate with the right end of the IC, (3) the right end of the IT might incorporate with the left end of the IC, and (4) the right end of the IT might incorporate with the right end of the IC. All of these possibilities are considered.

In general, the process of extension is similar to that of assimilation. A window of appropriate size and position is constructed at the end of each POCM. Several different approaches to determining the placement of these windows are presented in Section 5.2.1.2. Given these windows, a match/merge approach is again used. During the match phase, knowledge of the structure of the fragments in the windows is suspended. Again maximum fragmats are created. However, in the case of extension, fragmats with cardinality less than that of the maximum fragmats may be required to allow the POCMs to incorporate. An example of why this is true is shown in Section 5.2.1.2.

The process of mapping POCMs together using maximum fragmats is known as 0-bye mapping, because none of the fragments which potentially can pair are allowed not to pair. However, the inclusion of a specific pairing of fragments in a maximum fragmat may cause the corresponding topological structure to be infeasible. The deletion of that one pair of fragments may produce a topological structure which is feasible, thus allowing the two POCMs to incorporate; the region of overlap which allows the two POCMs to incorporate will be less (by one fragment) than that which would have been produced by a maximum fragmat, but as long as the minimum overlap requirement is still met, the incorporation is a valid one. The term 1-bye mapping is used to describe the process when fragmats with cardinality one less than that of the maximum fragmats are created and used. In this form of mapping, all maximum fragmats are created; also for each maximum fragmat, all sub-fragmats containing one less pair are also created. Thus, for each maximum fragmat of length n present in 0-bye mapping, there are an additional n fragmats present in 1-bye mapping. Similarly, 2-bye mapping allows two arbitrary pairings form each maximum fragmat to be deleted, resulting in an expansion factor of $1 + n + \frac{n(n-1)}{2}$. The combinatorial explosion implied here continues for k-bye mapping, where k > 2. The actual expansion factor applied over all maximum fragmats can, in fact, be either larger or smaller than the factor suggested above. For example, the factor can be smaller because two different maximum fragmats may produce a common sub-fragmat when k-bye mapping is used. The factor can be larger because some k-bye fragmats may not be subsets of *any* maximum fragmat. However, it should be clear that 0-bye mapping is preferable.

The merge phase used during extension is identical to that used for assimilation. Each fragmat produced in the match phase is checked, one at a time, to determine whether or not it produces a topologically feasible incorporation. If the POCMs can be incorporated together in more than one way, then ambiguity is present, and the attempt at incorporation is deferred.

3. Multiple-Restriction-Enzyme Mapping

Multiple-Restriction-Enzyme (MRE) mapping is a technique which jointly (concurrently) uses the fragment-length data from multiple digestions of the clones to attempt to reduce the uncertainty of placing a clone. In this technique, each clone is digested separately by two or more restriction enzymes. In this presentation for illustrative purposes, only two restriction enzymes, named α and β, are assumed. The mapping process builds **companion POCMs** in both the α domain and the β domain; the resulting pair of companion POCMs will be referred to as a **compound POCM.** These two companion POCMs are constructed in unison, and there are two important invariants which must be maintained. (1) Each companion POCM must contain exactly the same clones. (2) Each clone must occur in "the same position" in both POCMs. (The concept of "the same position" will be based on the consistency of the relative positions of a clone's clone ends with respect to the other clone ends present.)

3.1. The Approach

Assume that two companion IT POCMs, ${}_{\alpha}$map and ${}_{\beta}$map, have been previously constructed which satisfy the two invariants above. (Note that a companion pair of POCMs consisting of the α digestion of a clone and the β digestion of the same clone satisfied these invariants.) When a new clone is introduced as an IC for incorporation into this pair of companion POCMs, one of the two domains is chosen as the primary search domain; here, α is chosen as the primary search domain. The α digestion is "dragged" across ${}_{\alpha}$map, searching for a window in which it can incorporate. If such a window is found in ${}_{\alpha}$map, a **companion window** is found in ${}_{\beta}$map which covers roughly (at least as much, but possibly more) the same region of the genome as the original window in ${}_{\alpha}$map. The combination of the two companion windows is referred to as a **compound window.** An attempt is made to incorporate the clone within the companion window in ${}_{\beta}$map using the β digestion of the clone. If the incorporation in ${}_{\beta}$map is successful, then the two newly created POCMs are checked to determine if their clone-end maps are compatible. (The exact details of clone-end map compatibility will be discussed in detail in Sections 3.3 and 5.3.) If they are compatible (indicating that all clones are in "the same position"), then a successful compound incorporation is declared and the resulting companion POCMs constitute a new pair satisfying the two invariants above and having one more clone present than the original ITs. If the incorporation in ${}_{\beta}$map is not successful or the resulting clone-end maps are not compatible, then it is assumed that the apparent incorporation of the clone in this window in ${}_{\alpha}$map was in error due to the fact that the working assumption was false.

Although the technique of MRE mapping has been expressed in terms of incorporating an IC *clone* into an IT, the technique works equally well for incorporating an IC with structure into an IT.

3.2. The Two-sided Puzzle

The requirement that clones occur in the same position in both of the companion POCMs is analogous to the construction of a jigsaw puzzle with patterns on both its top and bottom surface. As in Section 1.6, assume that each pattern is somewhat random, such as a picture of popcorn, and that the edges of each piece are straight, so that the shape of the piece supplies little or no information about its placement in the overall puzzle. In such a puzzle, each piece would have a pattern on each side. (Assume that it is possible to differentiate the top side of the piece from the bottom side.) A specific piece may *appear* to fit in a particular position based on the correspondence between the pattern on the top of the piece and the pattern on the top of the partially constructed puzzle so far. This apparent incorporation may be (a) correct, caused by the fact that the piece actually originated from this position, or (b) incorrect, caused by random chance, given the high uncertainty of placing a piece. A partial confirmation of whether the placement of this piece is correct or not is to determine whether the patterns match on the bottom surface. If the patterns *do not* match on the bottom surface, the placement is probably incorrect (situation (b) above). If the patterns *do* match, there is a high probability that this is the correct placement of the piece; at least the uncertainty that this is the correct placement of this piece has been significantly reduced. An analysis of how the uncertainty is reduced by MRE mapping is presented in Section 4.

3.3. Clone-end Maps and Their Compatibility

Clone ends can be used as a *synchronizing* mechanism to find corresponding regions in companion POCMs. This synchronization is based on the following observation.

> Given a genome and a set of clones (from which the POCMs will be constructed) there must be a total order (≤) for the clone ends in the underlying genome. The order of these clone ends is independent of the placement of any specific restriction site of any kind. Thus, in a POCM for any specific restriction enzyme, the placement of the (abstract) clone ends in the POCM should be consistent with the order of the actual clone ends in the underlying genome. As POCMs are constructed for two or more restriction enzymes, this consistency constraint should be true of all the POCMs constructed. If there is *no* total order for which the (abstract) clone ends of both POCEMs are consistent, then there is no concrete underlying reality from which both POCEMs can validly have been abstracted; thus, at least one of the POCEMs must be incorrect. If there is *some* total order for which the (abstract) clone ends of both POCEMs is consistent, then there is a concrete underlying reality from which both POCEMs *could* validly have been abstracted. (The two POCEMs may not correspond to the *actual* underlying reality, but at least there is a potential reality for which they are valid abstractions.)

Thus, given two POCMs, ${}_{\alpha}$map and ${}_{\beta}$map, both assumed to be valid abstractions of same region of an underlying genome, the POCEMs extracted from both must be consistent with the underlying genome, and thus compatible with one another. Two POCEMs are **compatible** if there *exists* a total ordering of the clone ends which is consistent with both POCEMs. The term "consistent" means that there is a refinement of the POCEM which *is* the total order. Here, the term "refinement" refers to the process of ordering the members of a set into a specific order. This is best shown by example. Consider the refinement lattice[17] of three (abstract) clone ends, as shown in Figure 3.1. Here, a line between two sequence-sets means that the sequence-set at the top of the line is a refinement of the sequence-set at the bottom of the line. In this example, A is consistent with H through M; B is consistent with H and I; C is consistent with K and M; etc. B is compatible with E because there is a total ordering of the clone ends (I) which is consistent with both B and E. B is compatible with G because there is a total ordering of the clone ends (H) which is consistent with both B and G. However, E is not compatible with G, because there is no total ordering of the clone ends which is consistent with both E and G. (Note that this compatibility relation is reflexive and symmetric, but not transitive.)

As a more concrete example, consider three sequence-sets of clone ends:

$$ss1 = [\{L_{C4}\},\{L_{C1}\},\{L_{C6},L_{C2}\},\{R_{C4},L_{C3}\},\{R_{C1}\},\{R_{C6}\},\{R_{C2}\},\{R_{C3}\}]$$
$$ss2 = [\{L_{C4}\},\{L_{C6},L_{C1}\},\{L_{C2}\},\{L_{C3}\},\{R_{C4}\},\{R_{C1}\},\{R_{C6},R_{C2}\},\{R_{C3}\}]$$
$$ss3 = [\{L_{C4}\},\{L_{C1}\},\{L_{C6},L_{C2}\},\{L_{C3}\},\{R_{C4}\},\{R_{C1}\},\{R_{C2}\},\{R_{C6}\},\{R_{C3}\}]$$

These are extracted from Figures 3.21, 3.18, and 3.19, respectively. *ss1* is compatible with *ss2* because there is a total order, $[\{L_{C4}\},\{L_{C1}\},\{L_{C6}\},\{L_{C2}\},\{L_{C3}\},\{R_{C4}\},\{R_{C1}\},\{R_{C6}\},\{R_{C2}\},\{R_{C3}\}]$ with which both *ss1* and *ss2* are consistent. However, *ss1* and *ss3* are not compatible, because *ss1* requires that R_{C6} precede R_{C2}, and *ss3* requires that R_{C2} precede R_{C6}. There is no total order which is consistent with both of these.

There is a simple algorithm for determining whether or not two POCEMs are compatible; this is shown in Figure 3.2. Here, the symbol "$\Leftarrow$" refers to the operator which extracts one object from the front of a list and "←" is the assignment operator.

This algorithm is based on a technique of rearranging the clone ends to find the least upper bound (lub) in the refinement lattice[17]. A simple variant of this algorithm can construct and return the lub itself.

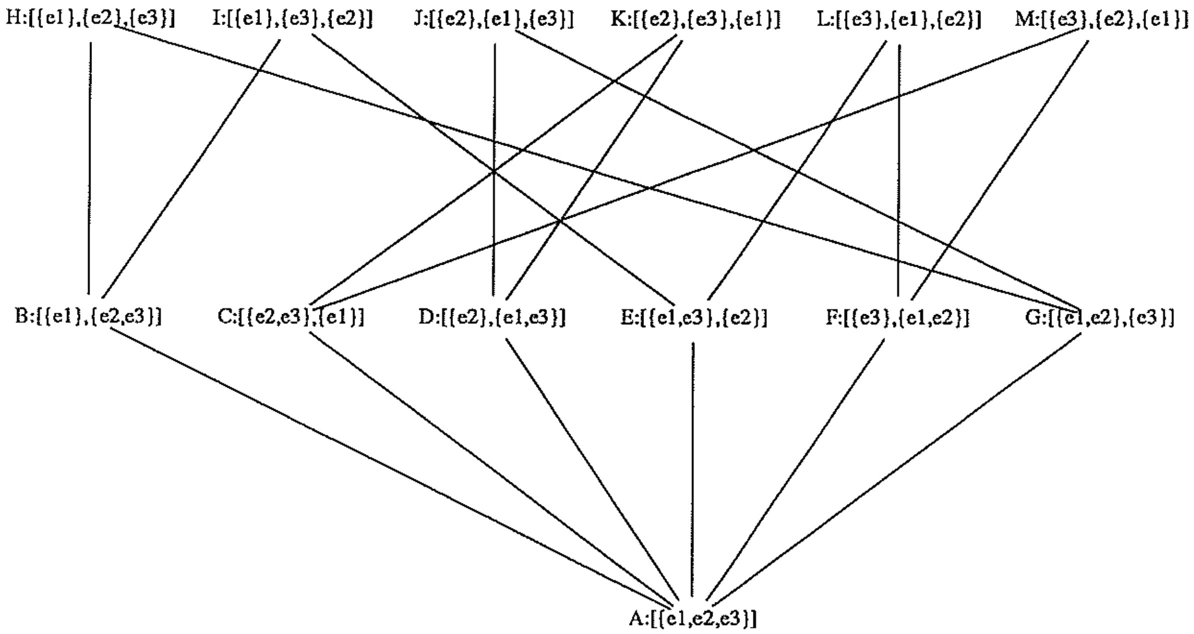


Figure 3.1: Clone End Lattice

In order to understand how the algorithm works, consider a trace of the algorithm when applied to *ss1* and *ss2* defined above. Table 3.1 shows the individual values of `set1`, `set2`, and `inter` at the point just after `inter` is computed. In fact, in general the sequence of values computed for `inter` *is* the lub of the sequences which were input. This algorithm assumes that there are no undetectable (due to the electrophoresis technology) fragments missing in the POCM from which the sequence-sets are extracted. The presence of missing undetectable fragments can disturb the companion POCMs sufficiently that they *seem* to be incompatible. This is discussed in more detail in Section 5.3, where a variation of the algorithm in Figure 3.2, which adjusts for this anomaly, is presented.

3.4. An Example using MRE Mapping

In this section, an extended example of how MRE mapping can help in determining the placement of clones is presented. The example genome shown in Figure 1.3, augmented by Tables 1.1-1.3, are the data used in this example. It will be shown that MRE mapping can be a *significant* help in resolving global ambiguity, and can be of *some* help in resolving local ambiguity. In the scenario to be presented, a clone is always selected as the IC, for simplicity; however, the technique works equally well if the IC has significant structure. First, clones C1 through C4 (in that order) will be incorporated; during the process one instance of global ambiguity will occur (during an extension) and be resolved by the MRE mapping technique. Next, an example of global ambiguity (during assimilation) will be exposed and resolved while attempting to incorporate clones C5 and C5′. Then, local ambiguity will be addressed while attempting to incorporate clones C6 through C6′′′′′; some local ambiguity problems will be resolvable by MRE mapping, and some will not be resolvable.

```
BOOLEAN
cem_compat(ss1,ss2)
    set1 $\Leftarrow$ ss1                                /* the empty set if ss1 is empty */
    set2 $\Leftarrow$ ss2                                /* the empty set if ss2 is empty */
    while (set1 $\neq \phi$ and set2 $\neq \phi$) {           /* not at end */
        inter $\leftarrow$ set1 $\cap$ set2                    /* compute the intersection */
        if (inter = set1) {                     /* if set2 subsumes set1 */
            set1 $\Leftarrow$ ss1                        /* extract new set1 from ss1 */
            if (inter = set2)                   /* if set1 = set2 */
                set2 $\Leftarrow$ ss2                    /* extract new set2 from ss2 */
            else                                /* otherwise */
                set2 $\leftarrow$ set2 - inter           /* construct residue */
        }
        else {
            if (inter = set2) {                 /* if set1 subsumes set2 */
                set2 $\Leftarrow$ ss2                    /* extract new set2 from ss2 */
                set1 $\leftarrow$ set1 - inter           /* construct residue */
            }
            else return(FALSE)                  /* neither set subsumes the other */
        }
    }
    if (set1 = set2 = $\phi$) return(TRUE)
    else return (FALSE)
```

Figure 3.2: Algorithm for Clone End Compatibility

Table 3.1
Trace of cem_compat

set1	set2	inter
$\{L_{C4}\}$	$\{L_{C4}\}$	$\{L_{C4}\}$
$\{L_{C1}\}$	$\{L_{C6},L_{C1}\}$	$\{L_{C1}\}$
$\{L_{C6},L_{C2}\}$	$\{L_{C6}\}$	$\{L_{C6}\}$
$\{L_{C2}\}$	$\{L_{C2}\}$	$\{L_{C2}\}$
$\{R_{C4},L_{C3}\}$	$\{L_{C3}\}$	$\{L_{C3}\}$
$\{R_{C4}\}$	$\{R_{C4}\}$	$\{R_{C4}\}$
$\{R_{C1}\}$	$\{R_{C1}\}$	$\{R_{C1}\}$
$\{R_{C6}\}$	$\{R_{C6},R_{C2}\}$	$\{R_{C6}\}$
$\{R_{C2}\}$	$\{R_{C2}\}$	$\{R_{C2}\}$
$\{R_{C3}\}$	$\{R_{C3}\}$	$\{R_{C3}\}$

Assume that 3-fragment overlap will be required in order to incorporate POCMs (this value is used to keep the example genome small), and that only 0-bye incorporations are considered. Assume that the primary search domain is the α domain. Assume C1 is selected as the original IT, and C2 is selected as the original IC. In attempting to incorporate the α digestions of C1 and C2, only one POCM is possible; this is shown in Figure 3.3. (Note the format of this figure. Component (a) shows how the PORFMs of the IT and IC were incorporated to produce the resulting map. The IT is presented above the IC; this is followed by the resulting PORFM. Component (b) shows the abstracted sequence-set notation for the POCM, which presents the combined relationship between the restriction fragments and clone ends. Component (c) shows the sequence-set-tree notation for the POCM, including the lengths of the fragments. Component (d) shows the sequence-set notation for the POCEM. This format will be used in the suite of figures that follow.) Here, the notation 5^2 means that fragment 5 is present twice in the corresponding virtual fragment. The companion window in the β domain (the entire POCM) is found in the β IT. There is only one way to incorporate the β digestion of C2 into the β IT; this is shown in Figure 3.4. The POCEMs are checked for compatibility, and the result is positive (because the POCEMs are identical). Since there are no ambiguity or compatibility problems and 3-fragment overlap was achieved in both domains, the attempted compound incorporation is successful. Thus, ${}_{\alpha}$map12 becomes the new α IT, and ${}_{\beta}$map12 becomes the new β IT.

Assume that C3 is selected as the next IC. Unfortunately, there are two positions for the α digestion of C3 to incorporate into ${}_{\alpha}$map12; this constitutes external ambiguity. These two incorporations are shown in Figures 3.5 and 3.6. In the POCM in Figure 3.5, C3 has been incorporated onto the wrong end of the IT. (The notation 3-11 indicates that the virtual fragment is composed of real fragment 3 and real fragment 11.) This is caused by confusion of fragment 3 in the IT with fragment 11 in the IC and fragment 4 in the IT with fragment 10 in the IC; fragment 7^2 in the IT was correctly identified with fragment 7 in the IC. (In this figure, the underlying confusion is being exposed, but the incorporation algorithm cannot detect the actual errors because it has only the fragment-length data in which fragment 3 and fragment 11 are perceived to be identical.) In the map in Figure 3.6, C3 has been incorporated onto the correct end of the IT, and in fact corresponds to the underlying reality.

When companion windows for these two maps are found in the β IT and incorporation is attempted, only one possible map can be constructed; this is shown in Figure 3.7. The POCEM of ${}_{\beta}$map123 is compatible with the POCEM of ${}_{\alpha}$map123$_2$, but it is not compatible with the POCEM of ${}_{\alpha}$map123$_1$. Thus, the algorithm discards ${}_{\alpha}$map123$_1$ as a possible solution because it is not compatible with *any* of the β POCMs. It is able to produce a unique pair of POCMs (${}_{\alpha}$map123$_2$,${}_{\beta}$map123) whose POCEMs are compatible, and declares a successful compound incorporation. These POCMs become the new pair of companion ITs.

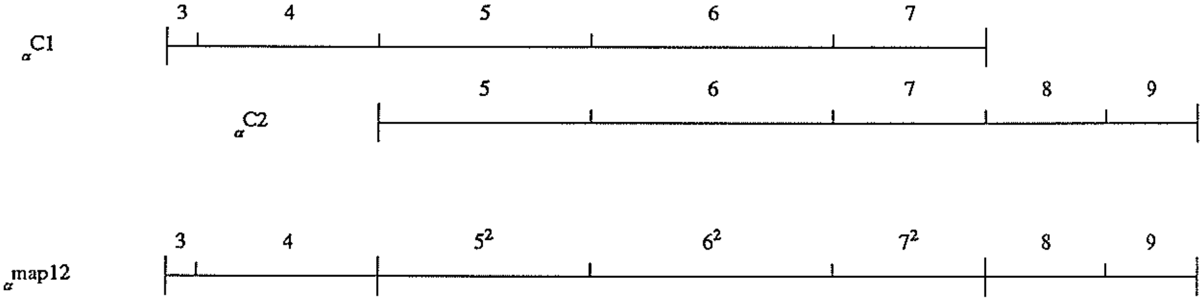
If only α data were being used to map C3, the global ambiguity present would have been unresolvable. The inclusion of β data and the use of clone end compatibility allows resolution of the ambiguity, and allows the appropriate compound POCM to be constructed.

Assume that C4 is selected as the next IC. As the α digestion is incorporated into ${}_{\alpha}$map123$_2$ only one possible result is obtained; this is shown in Figure 3.8. The companion window is found in ${}_{\beta}$map123, and only one incorporation is found possible; this is shown in Figure 3.9. The corresponding POCEMs are found to be compatible. Since a unique pair of maps is found, (${}_{\alpha}$map1234,${}_{\beta}$map1234), for which the POCEMs are compatible, the algorithm declares a successful compound incorporation. These POCMs become the new pair of companion ITs.

In the remainder of this running example, (${}_{\alpha}$map1234,${}_{\beta}$map1234) is repeatedly taken as the compound IT. In the next phase of the example, clones C5 and C5′ will be used to illustrate the use of MRE mapping to resolve global ambiguity. Note that the fragment-length data of the α digestions for C5 and C5′ are identical. However, they originate from different regions of the genome. If only α data were used to attempt to incorporate either C5 or C5′ into ${}_{\alpha}$map1234, global ambiguity would be present and unresolvable. However, inclusion of the β fragment-length data can be used to resolve this ambiguity.

First consider the incorporation of C5 into the IT (${}_{\alpha}$map1234,${}_{\beta}$map1234). As assimilation windows are dragged across ${}_{\alpha}$map1234, two windows are found in which the α digestion of C5 will incorporate. Each of these windows produces one incorporation; the resulting POCMs are shown in Figure 3.10 and 14. Thus, global ambiguity is present in the α domain. When the companion window corresponding to ${}_{\alpha}$map12345$_2$ is found in ${}_{\beta}$map1234, no incorporation of the β digestion of C5 is possible. Thus, this compound window is discarded. When the companion window corresponding to ${}_{\alpha}$map12345$_1$ is found in ${}_{\beta}$map1234, there is exactly one way to incorporate the β digestion of C5 into the window; this is shown in Figure 3.12. The POCEMs of ${}_{\alpha}$map12345$_1$ and ${}_{\beta}$map12345 are compatible, and thus a unique pair of companion POCMs has been found. These POCMs represent the correct positioning of C5.

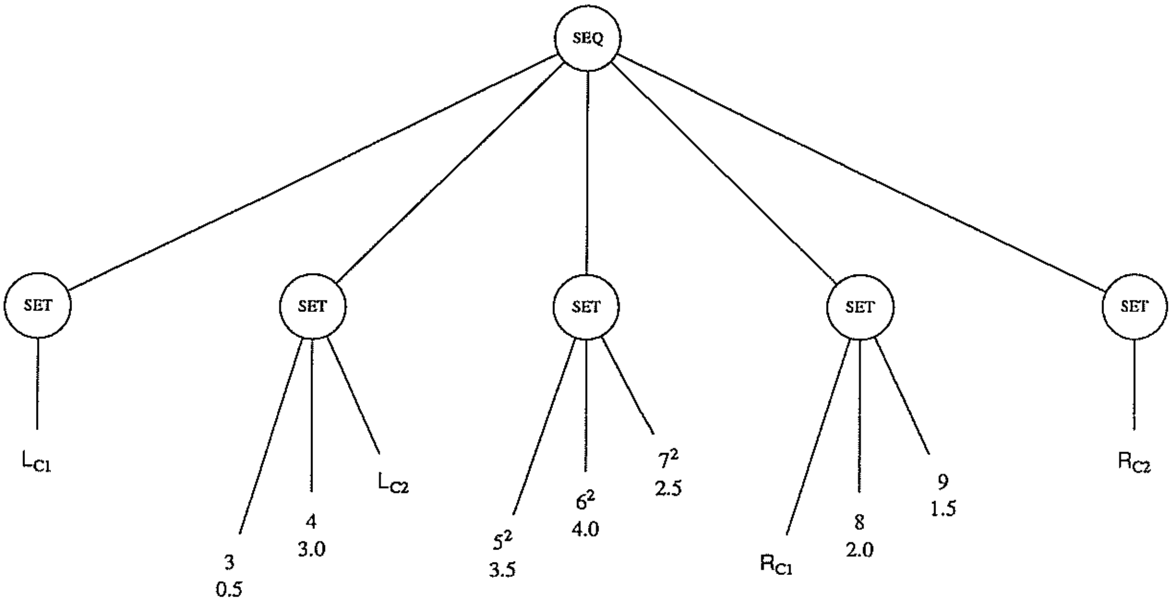
Now consider the incorporation of C5′ into the IT (${}_{\alpha}$map1234,${}_{\beta}$map1234). Analysis in the α domain produces essentially the same results as those obtained for C5; these results are shown in Figures 3.13 and 3.14. When the companion window corresponding to ${}_{\alpha}$map12345′$_1$ is found in ${}_{\beta}$map1234, no



(a) graphical form of the PORFM

$[\{L_{C1}\},\{3,4,L_{C2}\},\{5^2,6^2,7^2\},\{R_{C1},8,9\},\{R_{C2}\}]$

(b) sequence-set form of the POCM

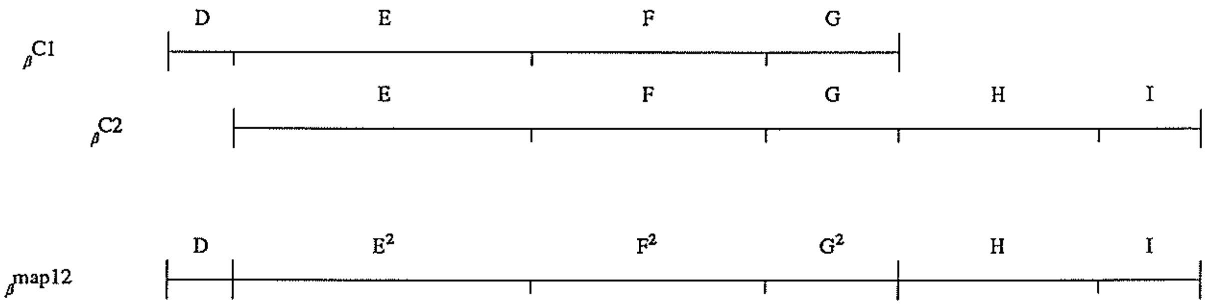


(c) sequence-set-tree form of the POCM

$[\{L_{C1}\},\{L_{C2}\},\{R_{C1}\},\{R_{C2}\}]$

(d) sequence-set form of the POCEM

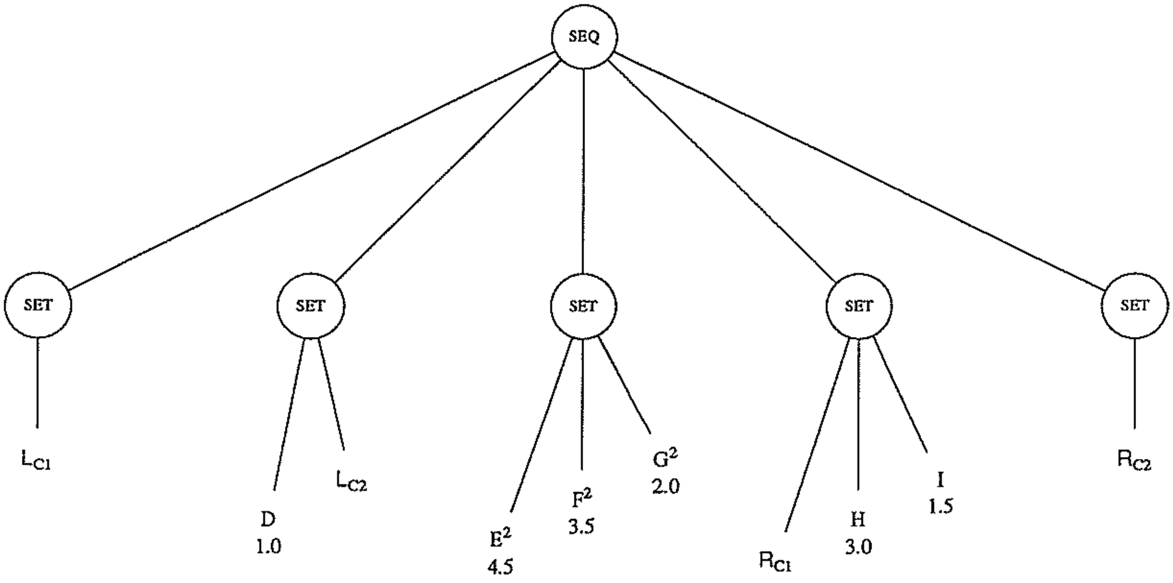
Figure 3.3: $_{\alpha}$map12



(a) graphical form of the PORFM

$[\{L_{C1}\},\{D,L_{C2}\},\{E^2,F^2,G^2\},\{R_{C1},H,I\},\{R_{C2}\}]$

(b) sequence-set form of the POCM

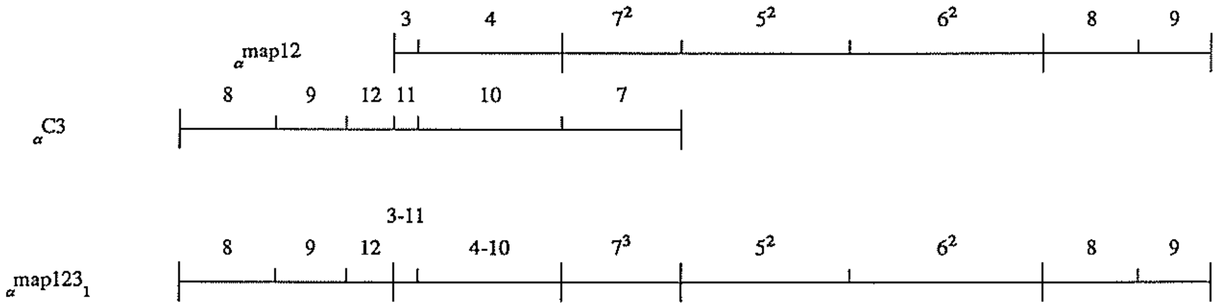


(c) sequence-set-tree form of the POCM

$[\{L_{C1}\},\{L_{C2}\},\{R_{C1}\},\{R_{C2}\}]$

(d) sequence-set form of the POCEM

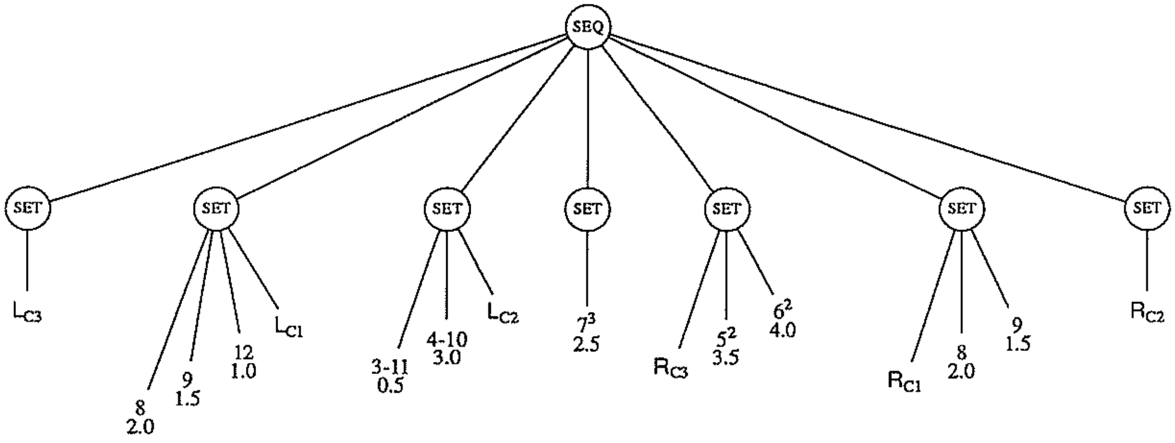
Figure 3.4: $_{\beta}$map12



(a) graphical form of the PORFM

$[\{L_{C3}\},\{8,9,12,L_{C1}\},\{3\text{-}11,4\text{-}10,L_{C2}\},\{7^3\},\{R_{C3},5^2,6^2\},\{R_{C1},8,9\},\{R_{C2}\}]$

(b) sequence-set form of the POCM

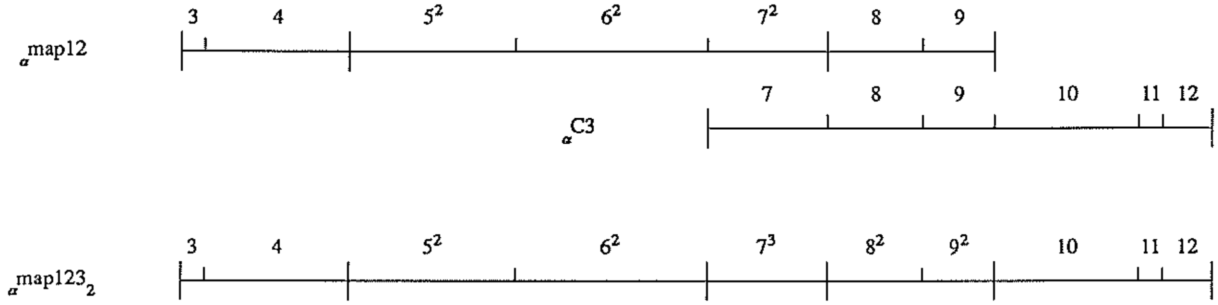


(c) sequence-set-tree form of the POCM

$[\{L_{C3}\},\{L_{C1}\},\{L_{C2}\},\{R_{C3}\},\{R_{C1}\},\{R_{C2}\}]$

(d) sequence-set form of the POCEM

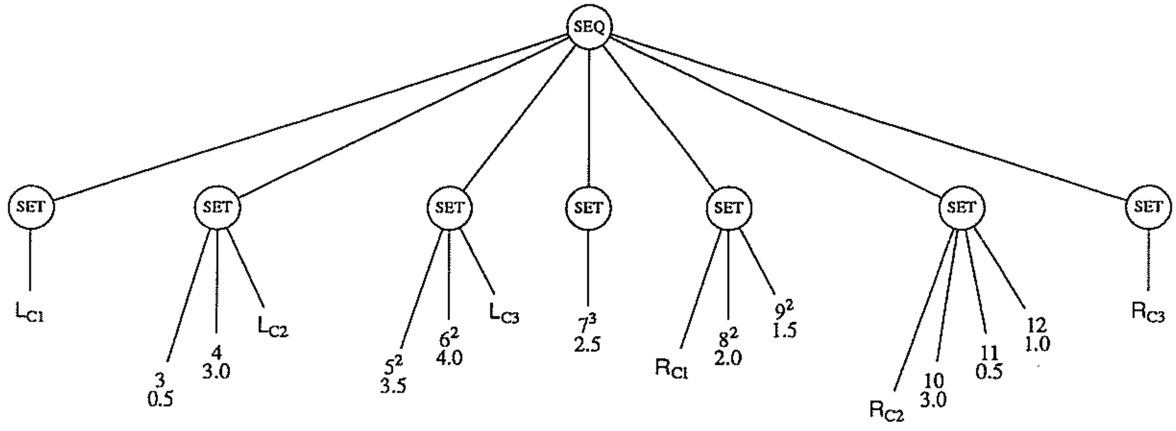
Figure 3.5: $_\alpha$map123$_1$



(a) graphical form of the PORFM

$[\{L_{C1}\},\{3,4,L_{C2}\},\{5^2,6^2,L_{C3}\},\{7^3\},\{R_{C1},8^2,9^2\},\{R_{C2},10,11,12\},\{R_{C3}\}]$

(b) sequence-set form of the POCM

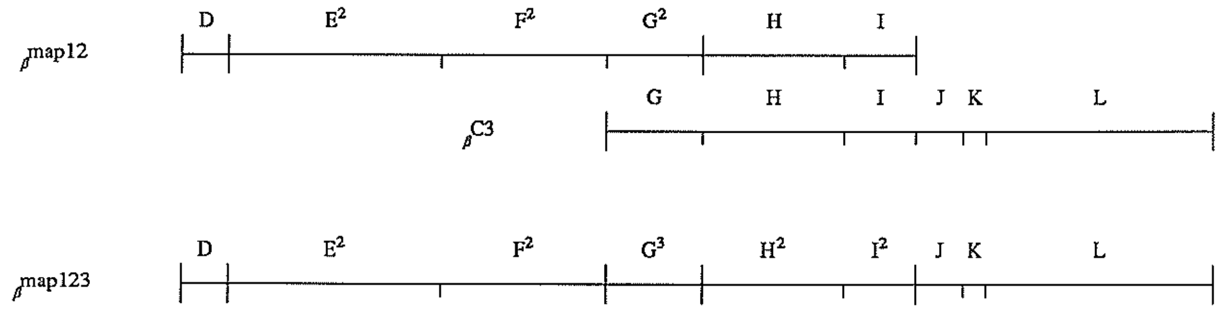


(c) sequence-set-tree form of the POCM

$[\{L_{C1}\},\{L_{C2}\},\{L_{C3}\},\{R_{C1}\},\{R_{C2}\},\{R_{C3}\}]$

(d) sequence-set form of the POCEM

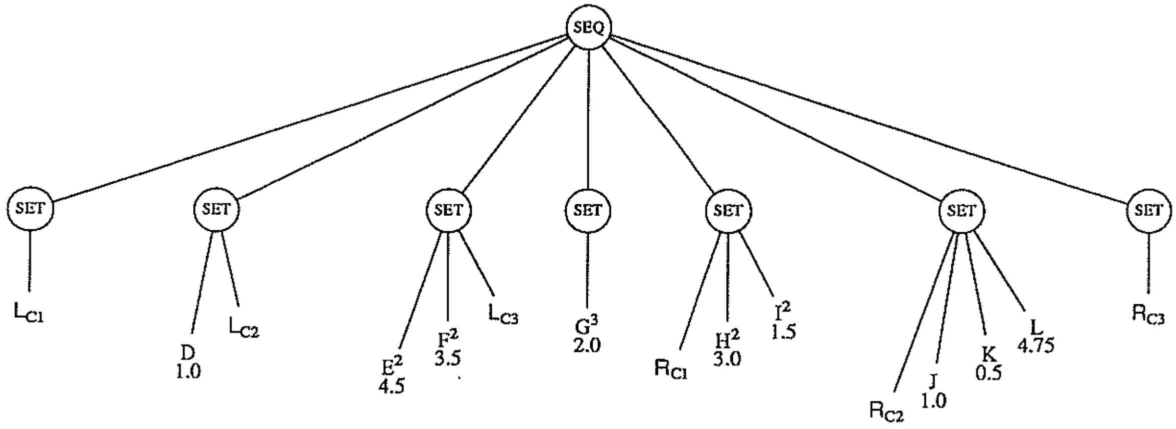
Figure 3.6: ${}_{\alpha}\text{map123}_2$



(a) graphical form of the PORFM

$[\{L_{C1}\},\{D,L_{C2}\},\{E^2,F^2,L_{C3}\},\{G^3\},\{R_{C1},H^2,I^2\},\{R_{C2},J,K,L\},\{R_{C3}\}]$

(b) sequence-set form of the POCM

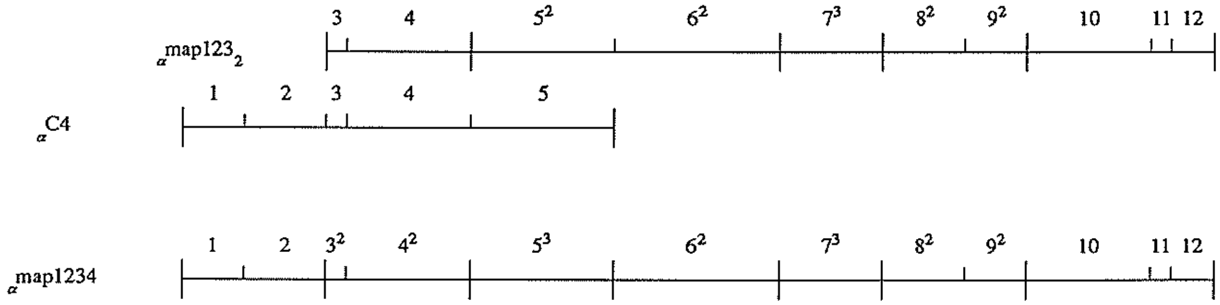


(c) sequence-set-tree form of the POCM

$[\{L_{C1}\},\{L_{C2}\},\{L_{C3}\},\{R_{C1}\},\{R_{C2}\},\{R_{C3}\}]$

(d) sequence-set form of the POCEM

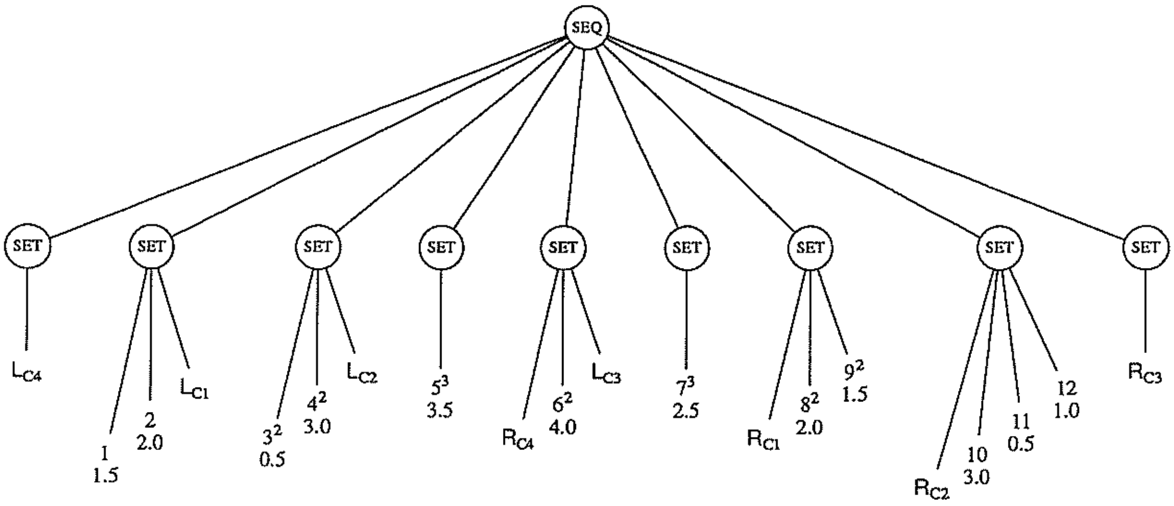
Figure 3.7: ${}_{\beta}$map123



(a) graphical form of the PORFM

$[\{L_{C4}\},\{1,2,L_{C1}\},\{3^2,4^2,L_{C2}\},\{5^3\},\{R_{C4},6^2,L_{C3}\},\{7^3\},\{R_{C1},8^2,9^2\},\{R_{C2},10,11,12\},\{R_{C3}\}]$

(b) sequence-set form of the POCM

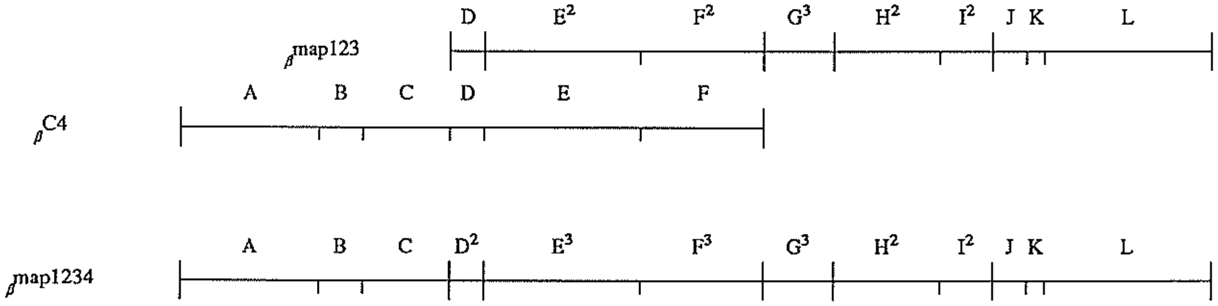


(c) sequence-set-tree form of the POCM

$[\{L_{C4}\},\{L_{C1}\},\{L_{C2}\},\{R_{C4},L_{C3}\},\{R_{C1}\},\{R_{C2}\},\{R_{C3}\}]$

(d) sequence-set form of the POCEM

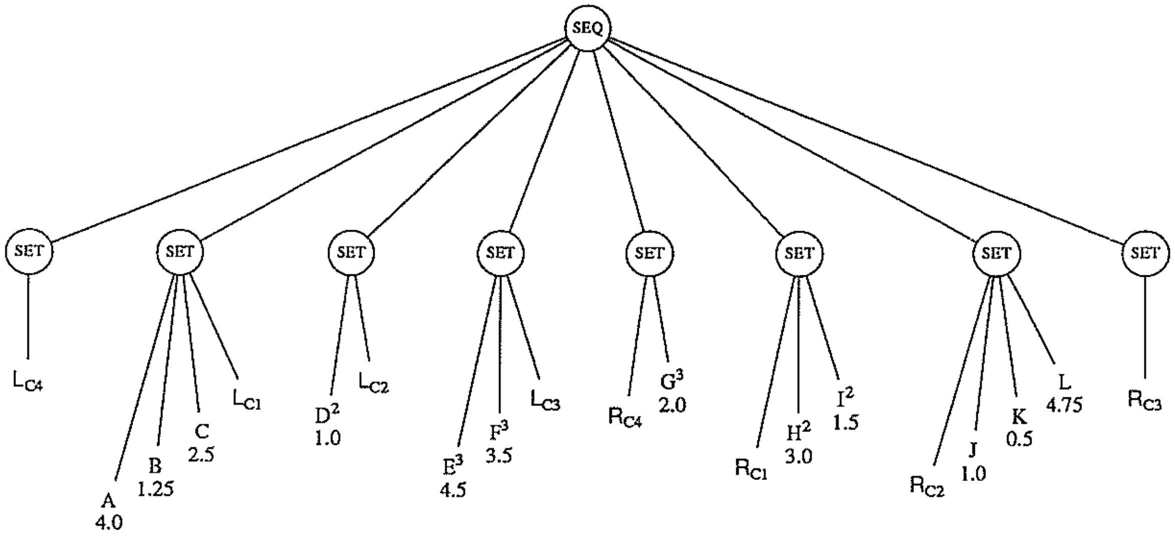
Figure 3.8: $_\alpha$map1234



(a) graphical form of the PORFM

$[\{L_{C4}\},\{A,B,C,L_{C1}\},\{D^2,L_{C2}\},\{E^3,F^3,L_{C3}\},\{R_{C4},G^3\},\{R_{C1},H^2,I^2\},\{R_{C2},J,K,L\},\{R_{C3}\}]$

(b) sequence-set form of the POCM

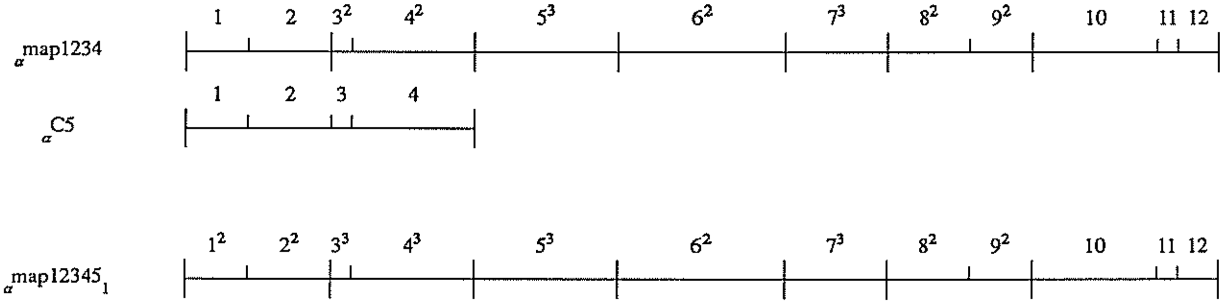


(c) sequence-set-tree form of the POCM

$[\{L_{C4}\},\{L_{C1}\},\{L_{C2}\},\{L_{C3}\},\{R_{C4}\},\{R_{C1}\},\{R_{C2}\},\{R_{C3}\}]$

(d) sequence-set form of the POCEM

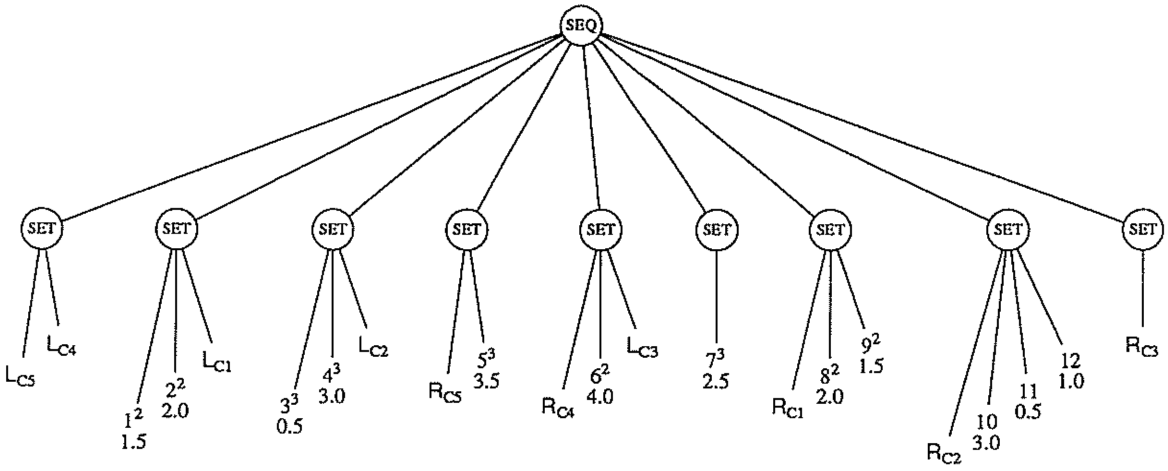
Figure 3.9: ${}_{\beta}$map1234



(a) graphical form of the PORFM

$[\{L_{C5},L_{C4}\},\{1^2,2^2,L_{C1}\},\{3^3,4^3,L_{C2}\},\{R_{C5},5^3\},\{R_{C4},6^2,L_{C3}\},\{7^3\},\{R_{C1},8^2,9^2\},\{R_{C2},10,11,12\},\{R_{C3}\}]$

(b) sequence-set form of the POCM

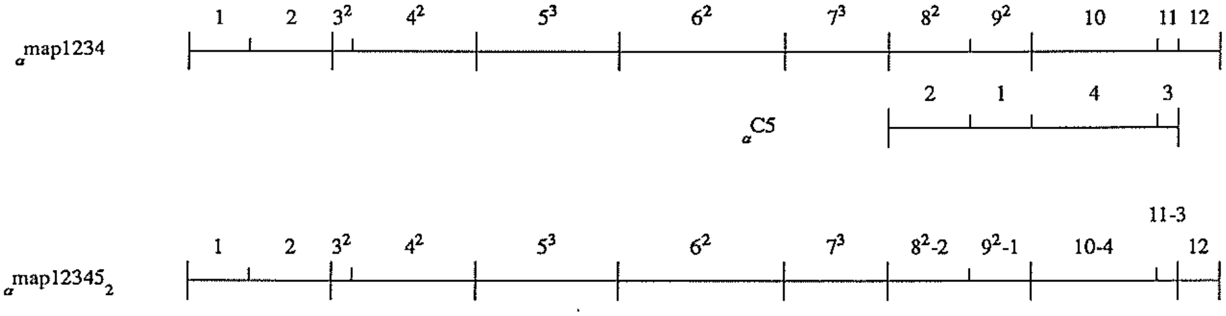


(c) sequence-set-tree form of the POCM

$[\{L_{C5},L_{C4}\},\{L_{C1}\},\{L_{C2}\},\{R_{C5}\},\{R_{C4},L_{C3}\},\{R_{C1}\},\{R_{C2}\},\{R_{C3}\}]$

(d) sequence-set form of the POCEM

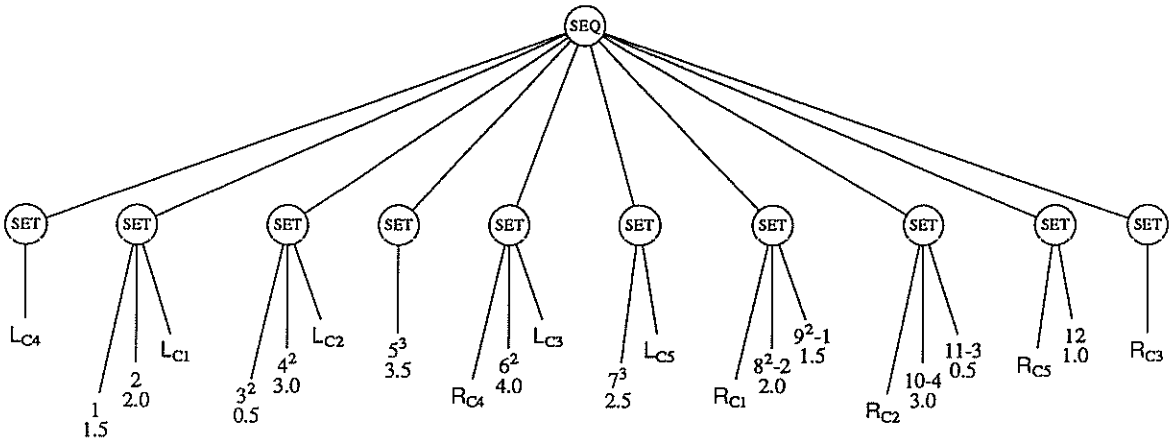
Figure 3.10: $_{\alpha}$map12345$_1$



(a) graphical form of the PORFM

$[\{L_{C4}\},\{1,2,L_{C1}\},\{3^2,4^2,L_{C2}\},\{5^3\},\{R_{C4},6^2,L_{C3}\},\{7^3,L_{C5}\},\{R_{C1},8^2\text{-}2,9^2\text{-}1\},\{R_{C2},10\text{-}4,11\text{-}3\},\{R_{C5},12\},\{R_{C3}\}]$

(b) sequence-set form of the POCM

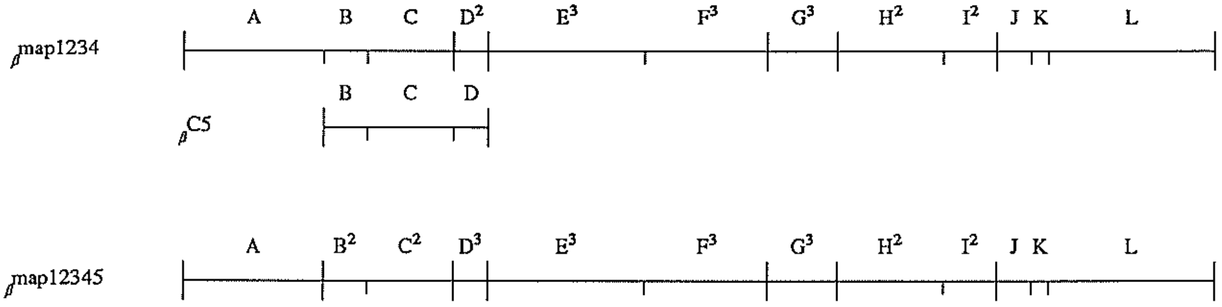


(c) sequence-set-tree form of the POCM

$[\{L_{C4}\},\{L_{C1}\},\{L_{C2}\},\{R_{C4},L_{C3}\},\{L_{C5}\},\{R_{C1}\},\{R_{C2}\},\{R_{C5}\},\{R_{C3}\}]$

(d) sequence-set form of the POCEM

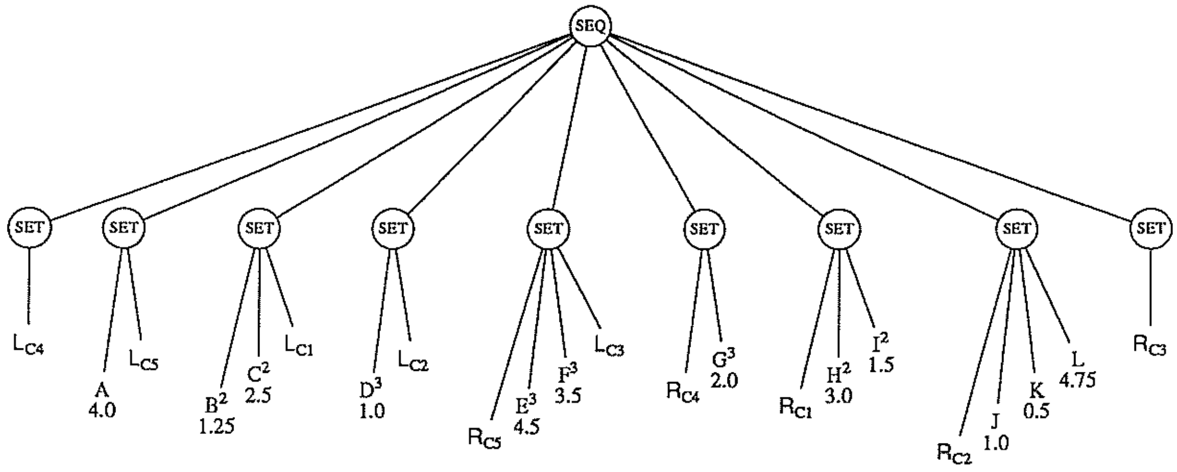
Figure 3.11: $_\alpha$map12345$_2$


$_{\beta}$map1234
A B C D^2 E^3 F^3 G^3 H^2 I^2 J K L
$_{\beta}$C5
B C D
$_{\beta}$map12345
A B^2 C^2 D^3 E^3 F^3 G^3 H^2 I^2 J K L

(a) graphical form of the PORFM

$[\{L_{C4}\},\{A,L_{C5}\},\{B^2,C^2,L_{C1}\},\{D^3,L_{C2}\},\{R_{C5},E^3,F^3,L_{C3}\},\{R_{C4},G^3\},\{R_{C1},H^2,I^2\},\{R_{C2},J,K,L\},\{R_{C3}\}]$

(b) sequence-set form of the POCM



(c) sequence-set-tree form of the POCM

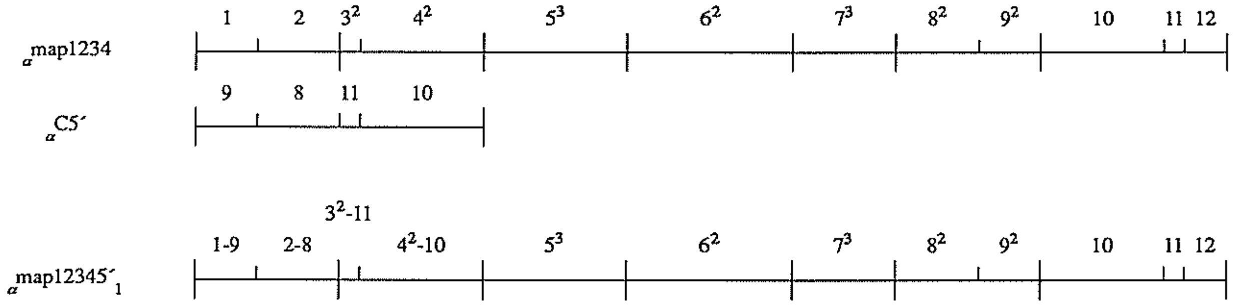
$[\{L_{C4}\},\{L_{C5}\},\{L_{C1}\},\{L_{C2}\},\{R_{C5},L_{C3}\},\{R_{C4}\},\{R_{C1}\},\{R_{C2}\},\{R_{C3}\}]$

(d) sequence-set form of the POCEM

Figure 3.12: $_{\beta}$map12345

incorporation of the β digestion of C5$'$ is possible. Thus this compound window is discarded. When the companion window corresponding to $_{\alpha}$map12345$'_2$ is found in $_{\beta}$map1234, there is exactly one way to incorporate the β digestion into the window; this is shown in Figure 3.15. The POCEMs of $_{\alpha}$map12345$'_2$ and $_{\beta}$map12345$'$ are compatible, and thus a unique pair of companion POCMs has been found. These maps represent the correct positioning of C5$'$. Here, the global ambiguity present in the α domain for both

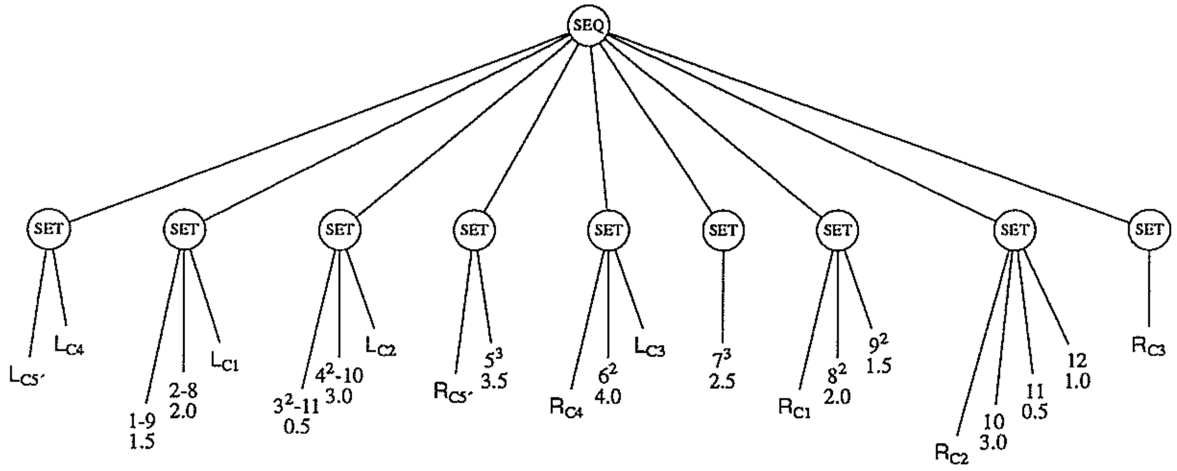
C5 and C5′ was correctly resolved by looking for confirmation in the β domain.



(a) graphical form of the PORFM

$[\{L_{C5'},L_{C4}\},\{1\text{-}9,2\text{-}8,L_{C1}\},\{3^2\text{-}11,4^2\text{-}10,L_{C2}\},\{R_{C5'},5^3\},\{R_{C4},6^2,L_{C3}\},\{7^3\},\{R_{C1},8^2,9^2\},\{R_{C2},10,11,12\},\{R_{C3}\}]$

(b) sequence-set form of the POCM

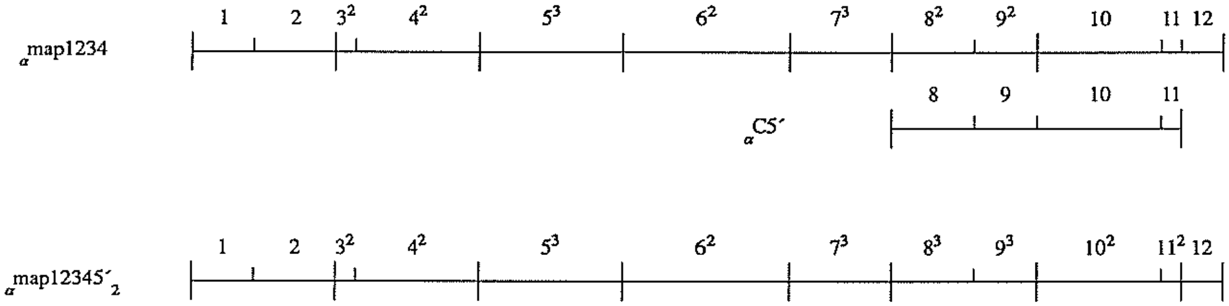


(c) sequence-set-tree form of the POCM

$[\{L_{C5'},L_{C4}\},\{L_{C1}\},\{L_{C2}\},\{R_{C5'}\},\{R_{C4},L_{C3}\},\{R_{C1}\},\{R_{C2}\},\{R_{C3}\}]$

(d) sequence-set form of the POCEM

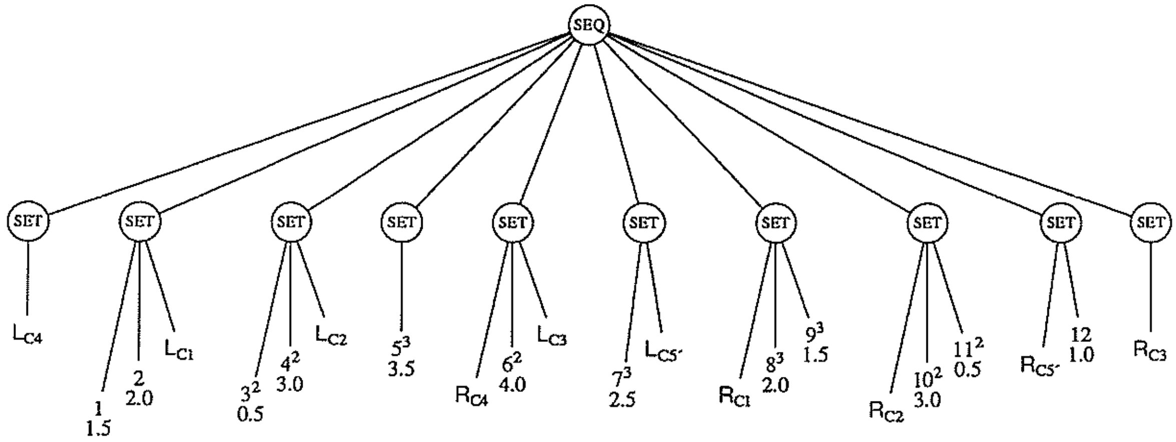
Figure 3.13: ${}_{\alpha}\text{map}12345'_1$



(a) graphical form of the PORFM

$[\{L_{C4}\},\{1,2,L_{C1}\},\{3^2,4^2,L_{C2}\},\{5^3\},\{R_{C4},6^2,L_{C3}\},\{7^3,L_{C5'}\},\{R_{C1},8^3,9^3\},\{R_{C2},10^2,11^2\},\{R_{C5'},12\},\{R_{C3}\}]$

(b) sequence-set form of the POCM

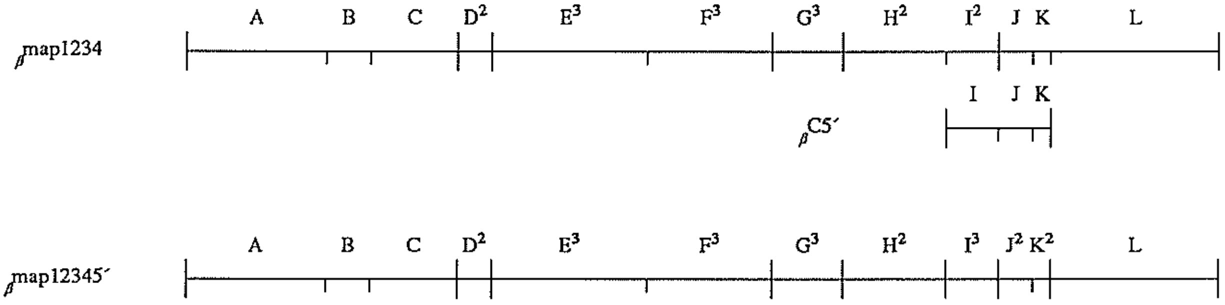


(c) sequence-set-tree form of the POCM

$[\{L_{C4}\},\{L_{C1}\},\{L_{C2}\},\{R_{C4},L_{C3}\},\{L_{C5'}\},\{R_{C1}\},\{R_{C2}\},\{R_{C5'}\},\{R_{C3}\}]$

(d) sequence-set form of the POCEM

Figure 3.14: ${}_{\alpha}$map12345$'_2$

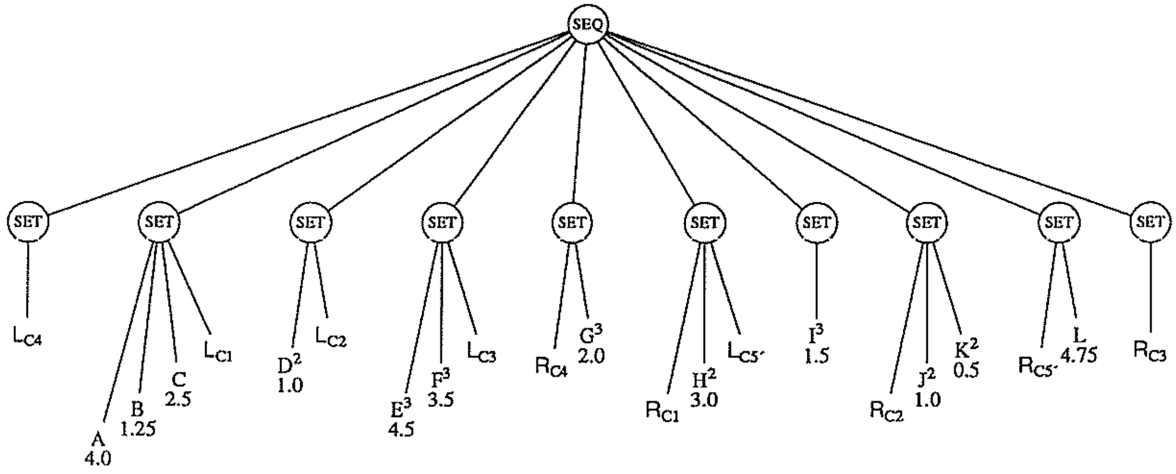


${}_{\beta}$map1234
A B C D^2 E^3 F^3 G^3 H^2 I^2 J K L
I J K
${}_{\beta}$C5′
${}_{\beta}$map12345′
A B C D^2 E^3 F^3 G^3 H^2 I^3 J^2 K^2 L

(a) graphical form of the PORFM

$[\{L_{C4}\},\{A,B,C,L_{C1}\},\{D^2,L_{C2}\},\{E^3,F^3,L_{C3}\},\{R_{C4},G^3\},\{R_{C1},H^2,L_{C5}'\},\{I^3\},\{R_{C2},J^2,K^2\},\{R_{C5}',L\},\{R_{C3}\}]$

(b) sequence-set form of the POCM



(c) sequence-set-tree form of the POCM

$[\{L_{C4}\},\{L_{C1}\},\{L_{C2}\},\{L_{C3}\},\{R_{C4}\},\{R_{C1},L_{C5'}\},\{R_{C2}\},\{R_{C5'}\},\{R_{C3}\}]$

(d) sequence-set form of the POCEM

Figure 3.15: ${}_{\beta}$map12345′

These two situations, in which MRE mapping has been shown to resolve global ambiguity, are not anomalous or contrived. The kind of resolution shown here occurs in almost all global ambiguity cases in which the companion ITs and ICs are "correct" in the first place. Clearly, MRE mapping supplies significant help in resolving global ambiguity. This should not be surprising, since the motivation for constructing this kind of algorithm comes from the two-sided puzzle analogy, in which the ambiguous placement of the puzzle pieces corresponds to the global ambiguity of clones. Perhaps a more surprising result is that

MRE mapping can sometimes help in resolving very subtle ambiguity in the placement of the clones, i.e., local ambiguity.

In this last phase of the running example, clones $C6$ through $C6'''''$ will be considered for incorporation into the IT $({}_{\alpha}\mathrm{map1234}, {}_{\beta}\mathrm{map1234})$. To preview the results here, ambiguity can be resolved for $C6'$ and $C6''$ and cannot be resolved for the other four.

Note the relationship between these six clones. All have the *identical* fragment-length data in their β digestions. They all have *similar* α fragment-length data, with fragments being either present or absent on each end. All contain fragments 5 through 8 in their α fragment-length data. Clones $C6$ through $C6''$ do not have fragment 9 on the right end, whereas clones $C6'''$ through $C6'''''$ do have fragment 9 on the right end. Clones $C6$ and $C6'''$ have fragments 3 and 4 present on their left ends; clones $C6'$ and $C6''''$ have only fragment 4 present on their left ends; clones $C6''$ and $C6'''''$ have neither fragment 3 nor fragment 4 present. These six clones represent all possible configurations which have *identical* β fragment-length data in this region of the genome.

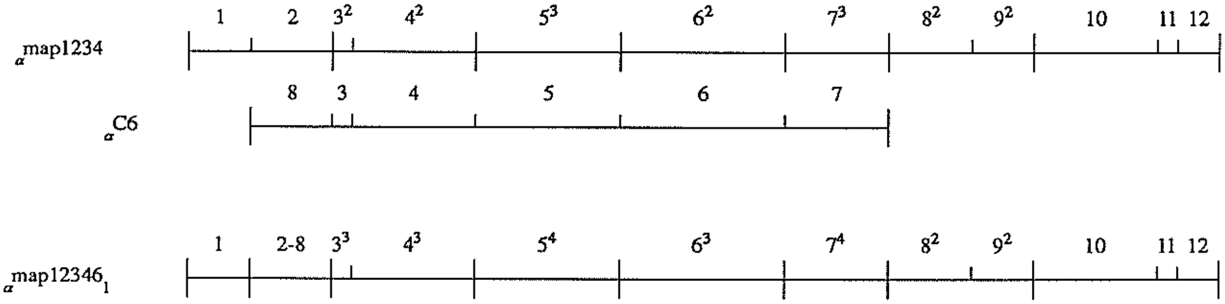
Assume that $C6$ is the IC selected for incorporation into the IT $({}_{\alpha}\mathrm{map1234}, {}_{\beta}\mathrm{map1234})$. There is only one window into which the α digestion of $C6$ will incorporate. However, two incorporations are possible; these are shown in Figures 3.16 and 3.17. In Figure 3.16, fragment 2 of the IT is confused with fragment 8 of $C6$; Figure 3.17 corresponds to the underlying reality. The corresponding companion window is found in the β domain, and there are two possible incorporations of the β digestion of $C6$; these are shown in Figures 3.18 and 3.19. Figure 3.18 corresponds to the underlying reality; in Figure 3.19, fragment J of the IT has been confused with fragment D of $C6$. The POCEMs of both ${}_{\alpha}\mathrm{map12346}_1$ and ${}_{\alpha}\mathrm{map12346}_2$ are compatible with that of ${}_{\beta}\mathrm{map12346}_1$; neither are compatible with that of ${}_{\beta}\mathrm{map12346}_2$. Even if this analysis eliminates ${}_{\beta}\mathrm{map12346}_2$ as a possible result (leaving only one possible map in the β domain, ${}_{\beta}\mathrm{map12346}_1$), a unique pair of solutions still has not been found, since there are still two α POCMs possible. Thus, local ambiguity in this case has not been resolved by MRE mapping.

As clones $C6'$ through $C6'''''$ are considered for incorporation into the IT, the possible outcomes in the β domain remain ${}_{\beta}\mathrm{map12346}_1$ and ${}_{\beta}\mathrm{map12346}_2$, since the β fragment-length data for all of $C6$ through $C6'''''$ are identical.

Assume that $C6'$ is now chosen as the IC for incorporation into the IT. There is only one window in which the α digestion of $C6'$ will incorporate, and only one incorporation is possible; this is shown in Figure 3.20. The exclusion of fragment 3 from $C6'$ eliminates the possibility of confusing fragment 2 of the IT with fragment 8 of $C6'$, because such a pairing of fragments would be topologically infeasible. The POCEM of ${}_{\alpha}\mathrm{map12346}'$ is compatible with that of ${}_{\beta}\mathrm{map12346}_1$ but not that of ${}_{\beta}\mathrm{map12346}_2$. Thus, a unique pair of POCMS, $({}_{\alpha}\mathrm{map12346}', {}_{\beta}\mathrm{map12346}_1)$, has been found, and MRE mapping has been able to resolve the local ambiguity in this case.

Moving to $C6''$, we find a similar situation. There is only one window in which the α digestion of $C6''$ will incorporate, and only one incorporation is possible; this is shown in Figure 3.21. The POCEM of ${}_{\alpha}\mathrm{map12346}''$ is compatible with that of ${}_{\beta}\mathrm{map12346}_1$ but not that of ${}_{\beta}\mathrm{map12346}_2$. Thus, a unique pair, $({}_{\alpha}\mathrm{map12346}'', {}_{\beta}\mathrm{map12346}_1)$, has been found, and MRE mapping has been able to resolve the local ambiguity in this case.

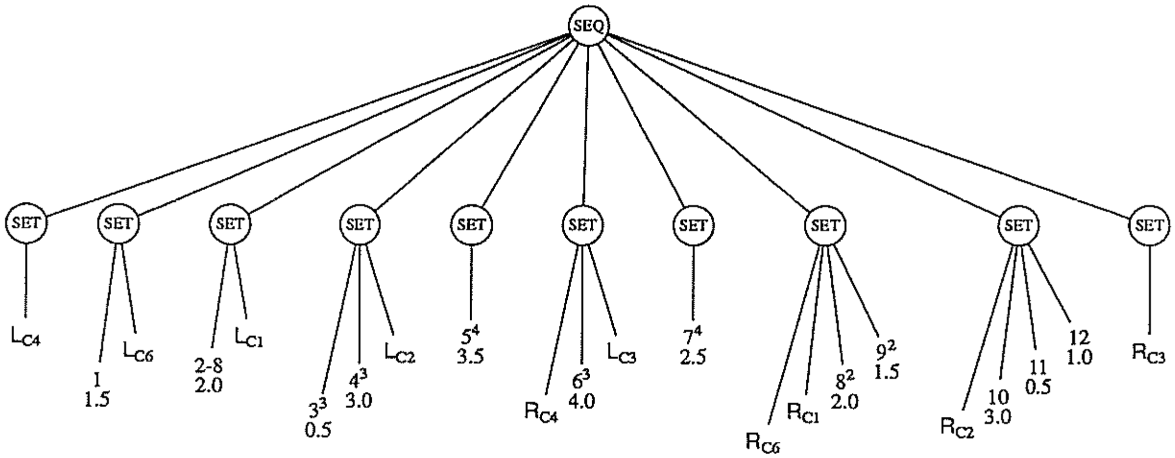
Now assume that $C6'''$ is chosen as the IC; this introduces fragment 9 to the α digestion. There are seven possible incorporations of the α digestion of $C6'''$ into ${}_{\alpha}\mathrm{map1234}$; these are shown in Figures 3.22 through 3.28. Figure 3.22 represents the correct underlying reality. In the other POCMs, fragment 2 can be confused with fragment 8, as was true with $C6$. However, the introduction of fragment 9 allows confusion of fragment 9 with fragment 1, and introduces a secondary confusion pattern involving the confusion of fragment 10 with fragment 4 and fragment 11 with fragment 3. The POCEMs of all of these POCMs are



(a) graphical form of the PORFM

$[\{L_{C4}\},\{1,L_{C6}\},\{2\text{-}8,L_{C1}\},\{3^3,4^3,L_{C2}\},\{5^4\},\{R_{C4},6^3,L_{C3}\},\{7^4\},\{R_{C6},R_{C1},8^2,9^2\},\{R_{C2},10,11,12\},\{R_{C3}\}]$

(b) sequence-set form of the POCM

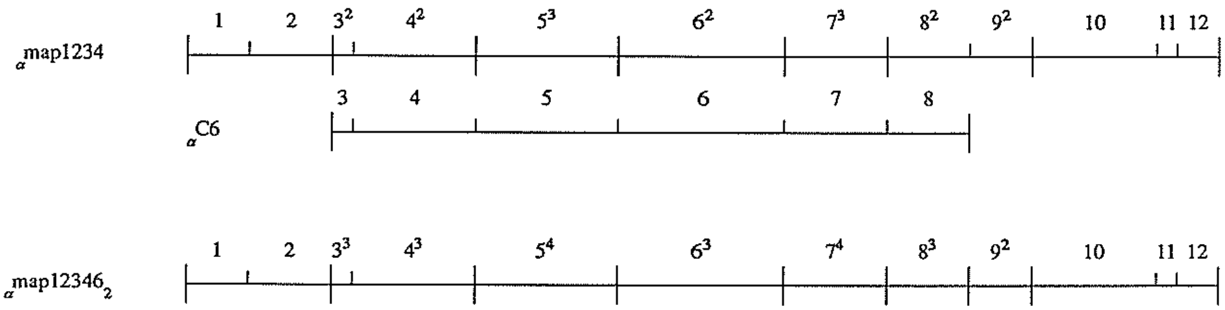


(c) sequence-set-tree form of the POCM

$[\{L_{C4}\},\{L_{C6}\},\{L_{C1}\},\{L_{C2}\},\{R_{C4},L_{C3}\},\{R_{C6},R_{C1}\},\{R_{C2}\},\{R_{C3}\}]$

(d) sequence-set form of the POCEM

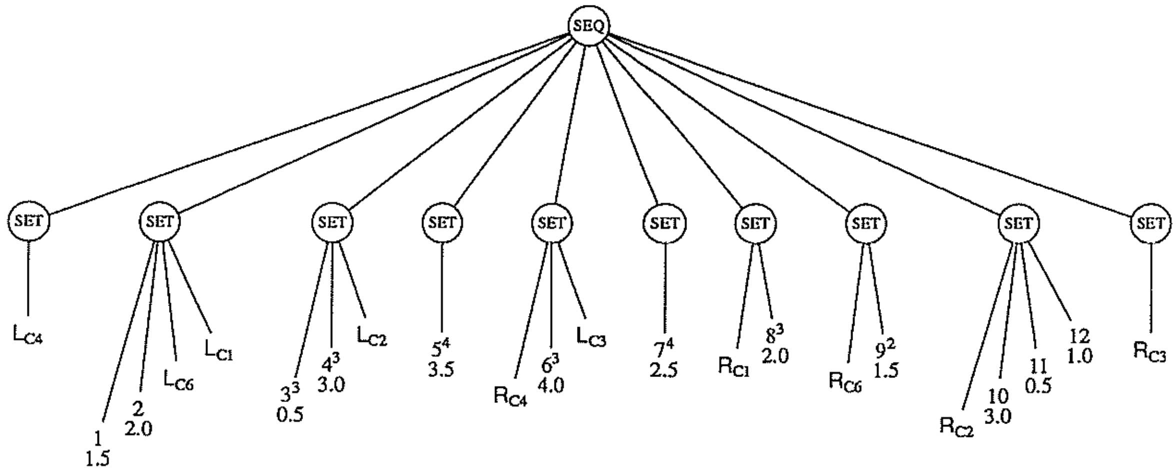
Figure 3.16: $_\alpha$map12346$_1$



(a) graphical form of the PORFM

$[\{L_{C4}\},\{1,2,L_{C6},L_{C1}\},\{3^3,4^3,L_{C2}\},\{5^4\},\{R_{C4},6^3,L_{C3}\},\{7^4\},\{R_{C1},8^3\},\{R_{C6},9^2\},\{R_{C2},10,11,12\},\{R_{C3}\}]$

(b) sequence-set form of the POCM

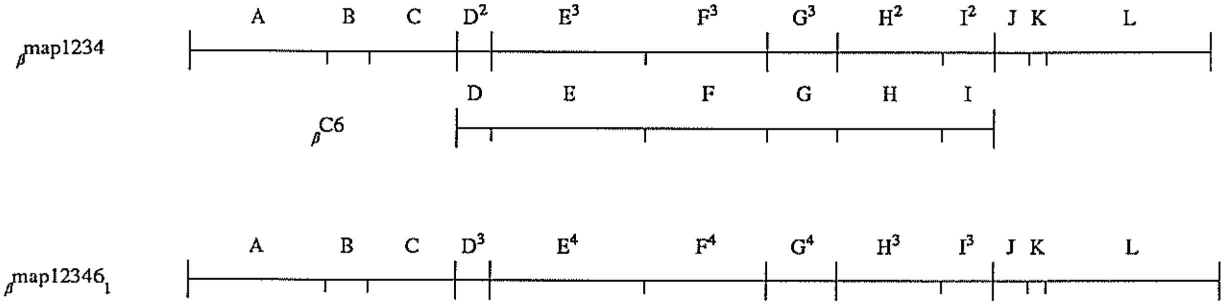


(c) sequence-set-tree form of the POCM

$[\{L_{C4}\},\{L_{C6},L_{C1}\},\{L_{C2}\},\{R_{C4},L_{C3}\},\{R_{C1}\},\{R_{C6}\},\{R_{C2}\},\{R_{C3}\}]$

(d) sequence-set form of the POCEM

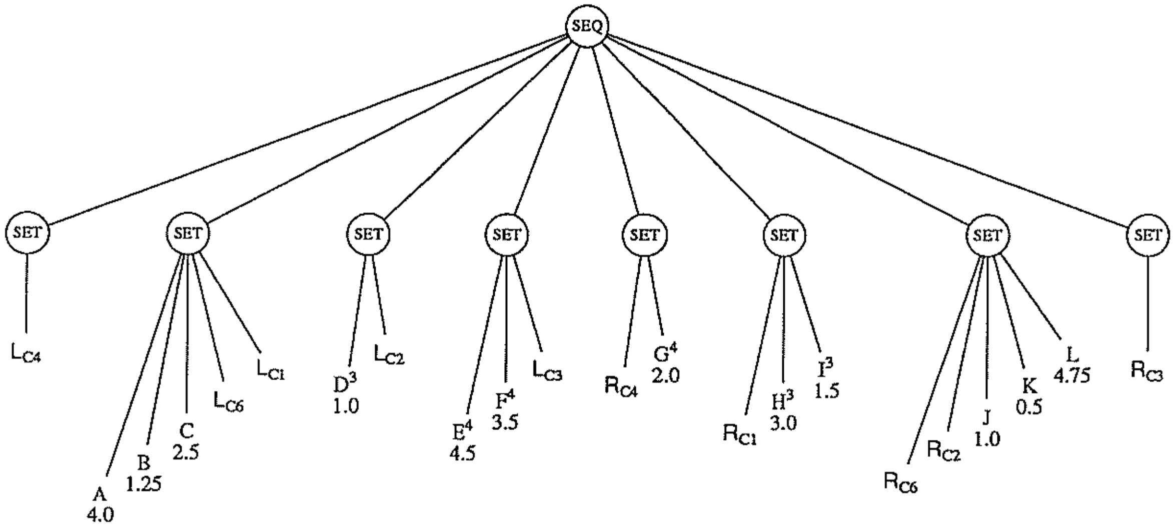
Figure 3.17: ${}_{\alpha}\text{map12346}_2$



(a) graphical form of the PORFM

$[\{L_{C4}\},\{A,B,C,L_{C6},L_{C1}\},\{D^3,L_{C2}\},\{E^4,F^4,L_{C3}\},\{R_{C4},G^4\},\{R_{C1},H^3,I^3\},\{R_{C6},R_{C2},J,K,L\},\{R_{C3}\}]$

(b) sequence-set form of the POCM

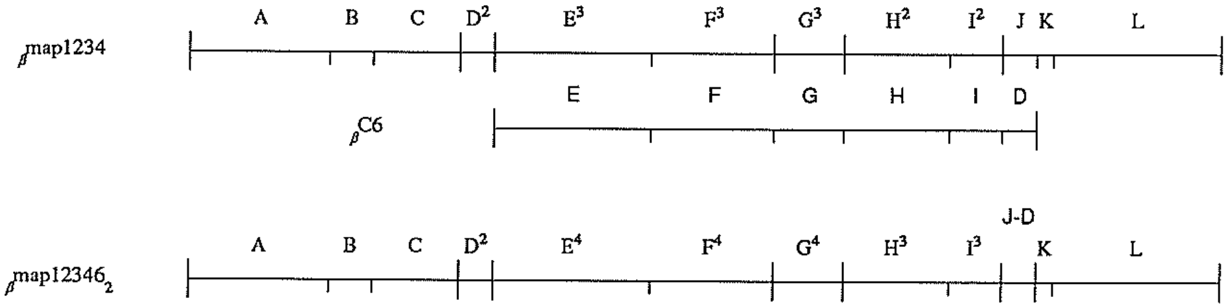


(c) sequence-set-tree form of the POCM

$[\{L_{C4}\},\{L_{C6},L_{C1}\},\{L_{C2}\},\{L_{C3}\},\{R_{C4}\},\{R_{C1}\},\{R_{C6},R_{C2}\},\{R_{C3}\}]$

(d) sequence-set form of the POCEM

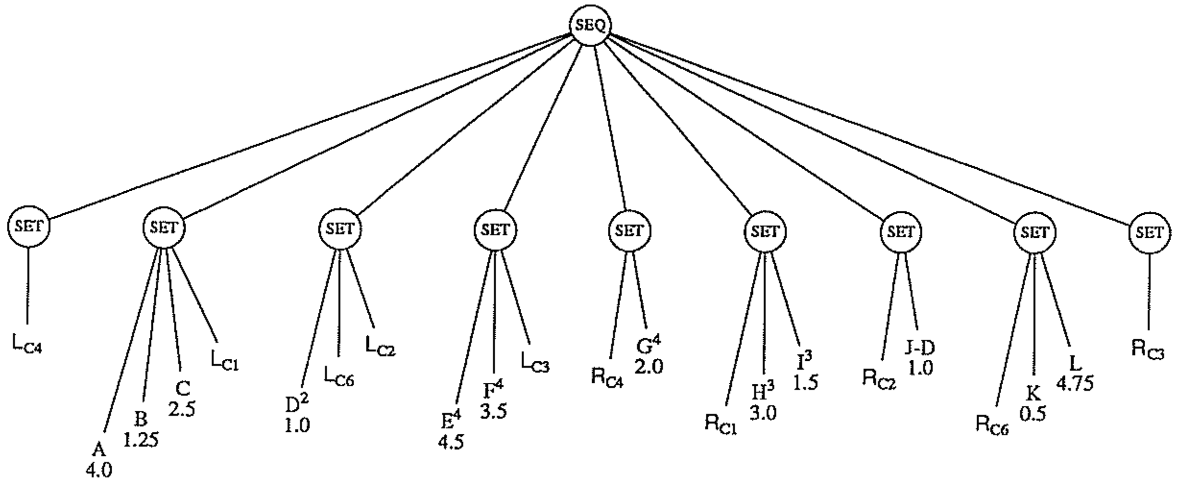
Figure 3.18: $_{\beta}$map12346$_1$



(a) graphical form of the PORFM

$[\{L_{C4}\},\{A,B,C,L_{C1}\},\{D^2,L_{C6},L_{C2}\},\{E^4,F^4,L_{C3}\},\{R_{C4},G^4\},\{R_{C1},H^3,I^3\},\{R_{C2},J\text{-}D\},\{R_{C6},K,L\},\{R_{C3}\}]$

(b) sequence-set form of the POCM

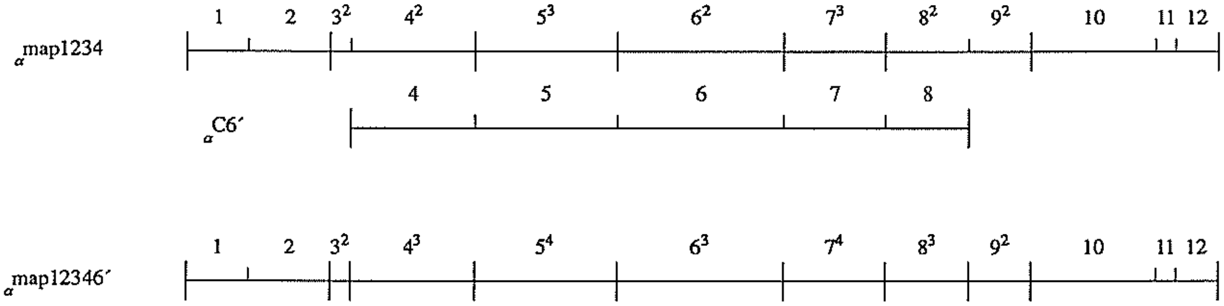


(c) sequence-set-tree form of the POCM

$[\{L_{C4}\},\{L_{C1}\},\{L_{C6},L_{C2}\},\{L_{C3}\},\{R_{C4}\},\{R_{C1}\},\{R_{C2}\},\{R_{C6}\},\{R_{C3}\}]$

(d) sequence-set form of the POCEM

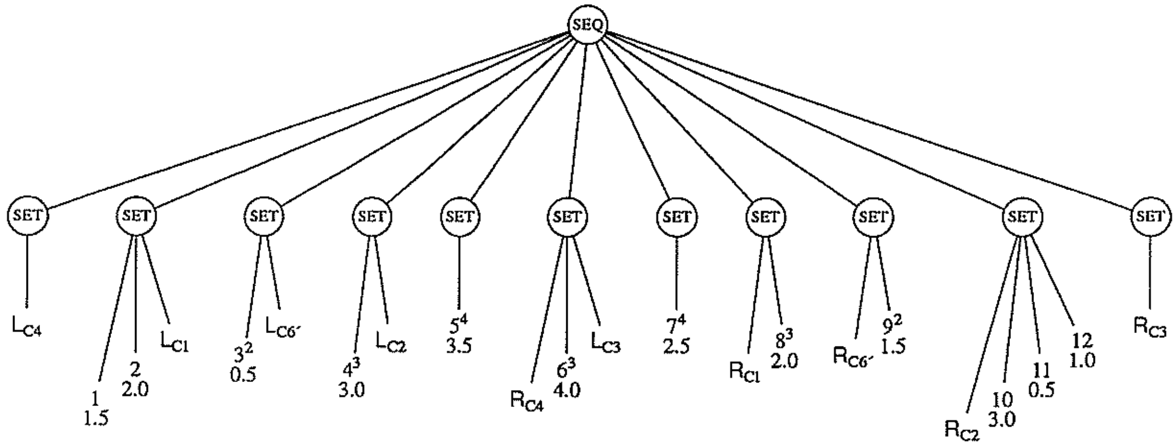
Figure 3.19: $_{\beta}$map12346$_2$



(a) graphical form of the PORFM

$[\{L_{C4}\},\{1,2,L_{C1}\},\{3^2,L_{C6'}\},\{4^3,L_{C2}\},\{5^4\},\{R_{C4},6^3,L_{C3}\},\{7^4\},\{R_{C1},8^3\},\{R_{C6'},9^2\},\{R_{C2},10,11,12\},\{R_{C3}\}]$

(b) sequence-set form of the POCM

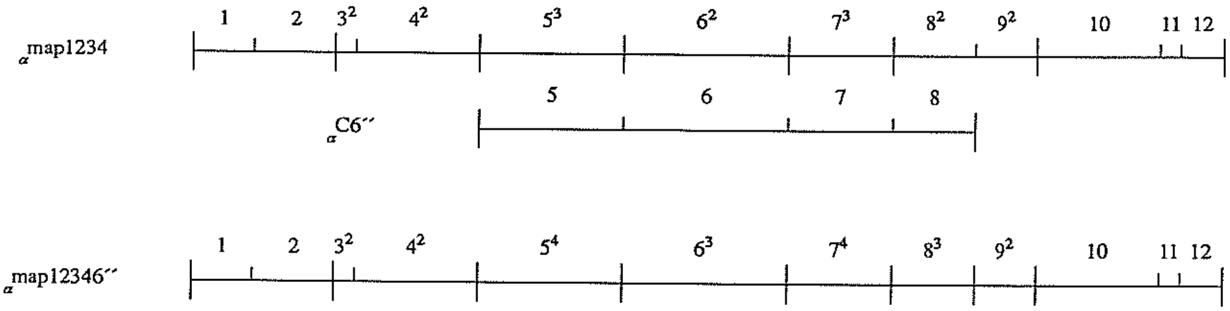


(c) sequence-set-tree form of the POCM

$[\{L_{C4}\},\{L_{C1}\},\{L_{C6'}\},\{L_{C2}\},\{R_{C4},L_{C3}\},\{R_{C1}\},\{R_{C6'}\},\{R_{C2}\},\{R_{C3}\}]$

(d) sequence-set form of the POCEM

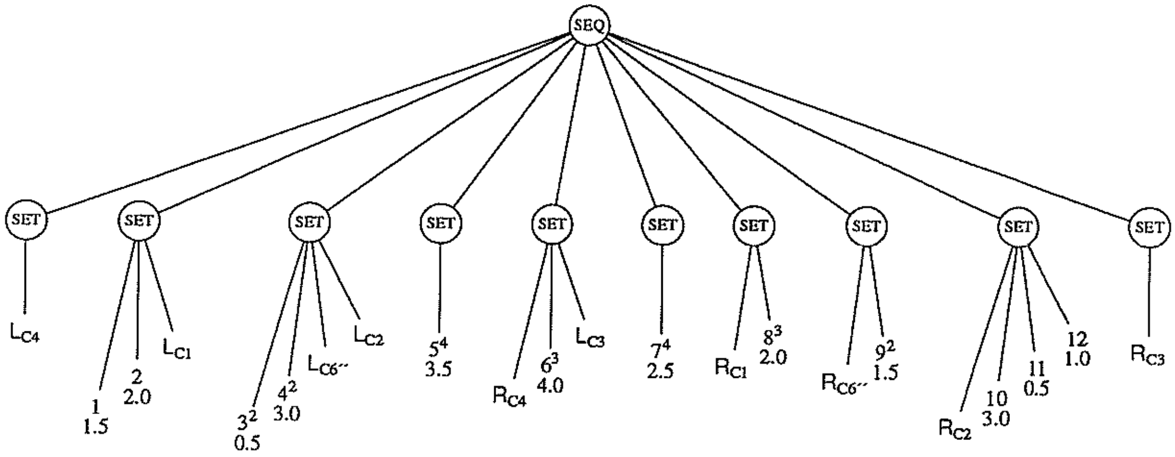
Figure 3.20: $_{\alpha}$map12346′



(a) graphical form of the PORFM

$[\{L_{C4}\},\{1,2,L_{C1}\},\{3^2,4^2,L_{C6''},L_{C2}\},\{5^4\},\{R_{C4},6^3,L_{C3}\},\{7^4\},\{R_{C1},8^3\},\{R_{C6''},9^2\},\{R_{C2},10,11,12\},\{R_{C3}\}]$

(b) sequence-set form of the POCM



(c) sequence-set-tree form of the POCM

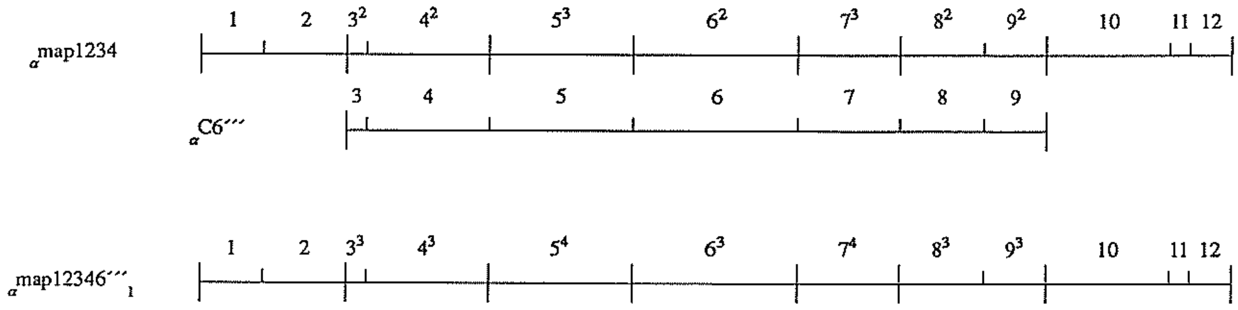
$[\{L_{C4}\},\{L_{C1}\},\{L_{C6''},L_{C2}\},\{R_{C4},L_{C3}\},\{R_{C1}\},\{R_{C6''}\},\{R_{C2}\},\{R_{C3}\}]$

(d) sequence-set form of the POCEM

Figure 3.21: $_\alpha$map12346''

compatible with that of $_\beta$map12346$_1$. The POCEMs of all but three of these POCMs ($_\alpha$map12346'''$_2$, $_\alpha$map12346'''$_3$, and $_\alpha$map12346'''$_4$) are compatible with that of $_\beta$map12346$_2$. Clearly, there is no unique solution.

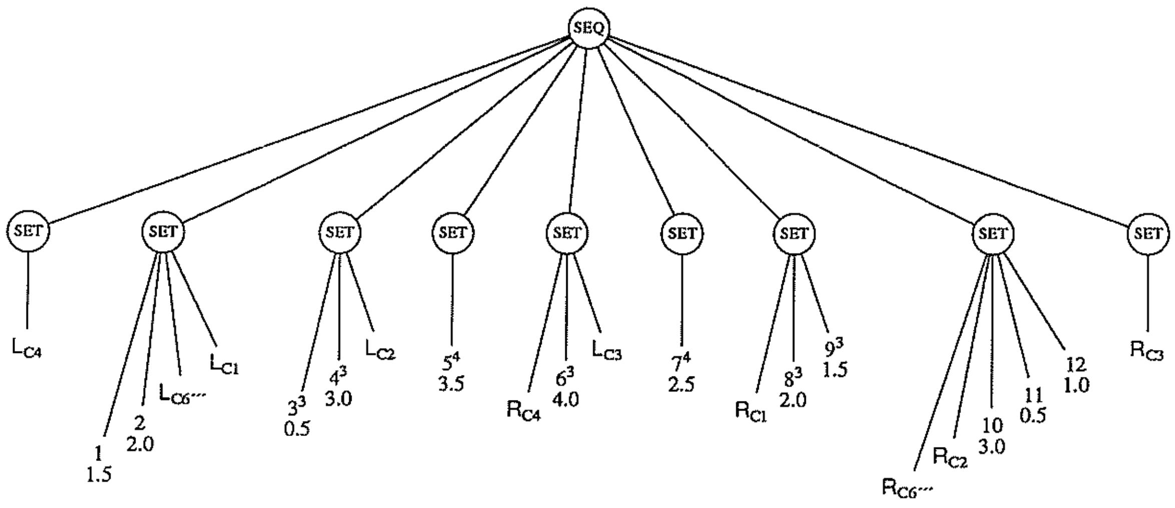
Note that different POCMs can have the identical PORFM. For instance, the POCMs for $_\alpha$map12346'''$_1$ and $_\alpha$map12346'''$_4$ have the identical group structure and identical virtual fragment membership within the groups; only the multiplicity of the real fragments associated with some of the virtual fragments differs between the two corresponding PORFMs. The clone end information present in the POCMs supplies significant information about the structure of the map being constructed.



(a) graphical form of the PORFM

$[\{L_{C4}\},\{1,2,L_{C6'''},L_{C1}\},\{3^3,4^3,L_{C2}\},\{5^4\},\{R_{C4},6^3,L_{C3}\},\{7^4\},\{R_{C1},8^3,9^3\},\{R_{C6'''},R_{C2},10,11,12\},\{R_{C3}\}]$

(b) sequence-set form of the POCM

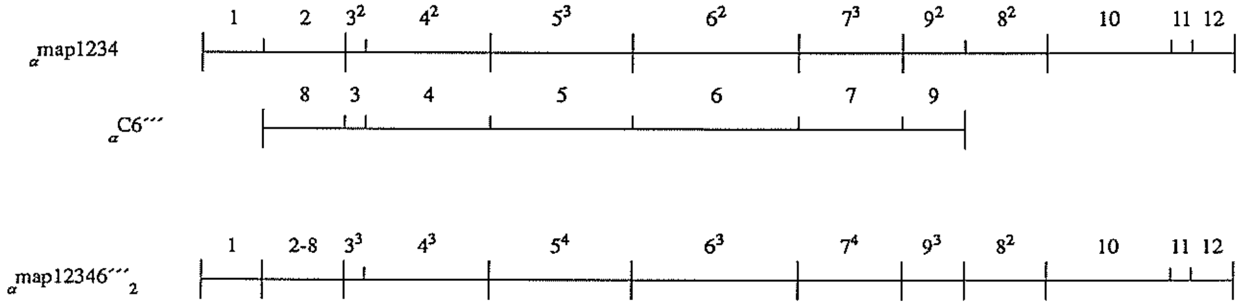


(c) sequence-set-tree form of the POCM

$[\{L_{C4}\},\{L_{C6'''},L_{C1}\},\{L_{C2}\},\{R_{C4},L_{C3}\},\{R_{C1}\},\{R_{C6'''},R_{C2}\},\{R_{C3}\}]$

(d) sequence-set form of the POCEM

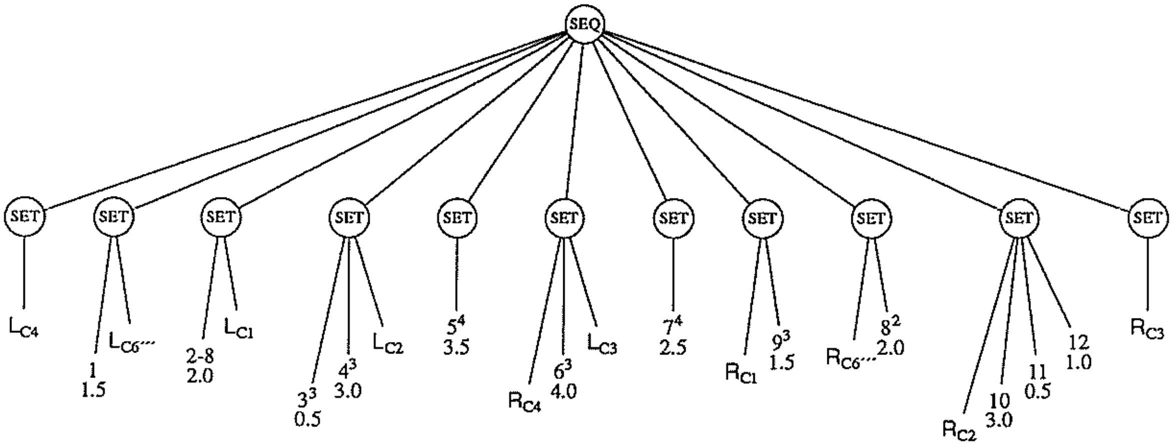
Figure 3.22: $_{\alpha}\text{map12346}'''_{1}$



(a) graphical form of the PORFM

$[\{L_{C4}\},\{1,L_{C6'''}\},\{2\text{-}8,L_{C1}\},\{3^3,4^3,L_{C2}\},\{5^4\},\{R_{C4},6^3,L_{C3}\},\{7^4\},\{R_{C1},9^3\},\{R_{C6'''},8^2\},\{R_{C2},10,11,12\},\{R_{C3}\}]$

(b) sequence-set form of the POCM

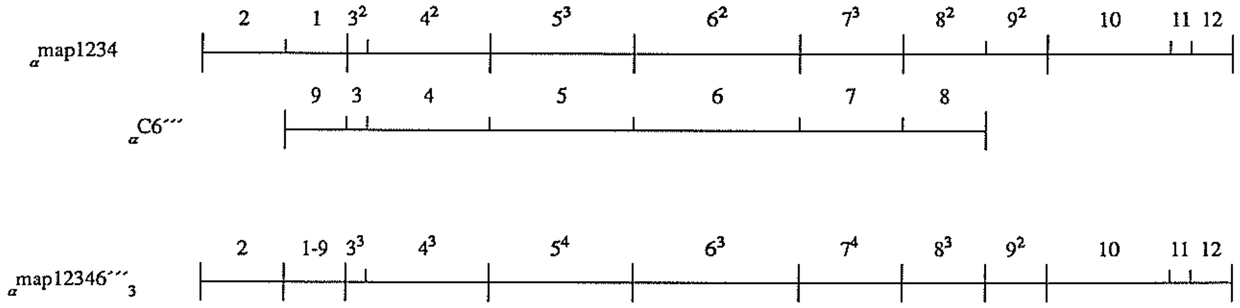


(c) sequence-set-tree form of the POCM

$[\{L_{C4}\},\{L_{C6'''}\},\{L_{C1}\},\{L_{C2}\},\{R_{C4},L_{C3}\},\{R_{C1}\},\{R_{C6'''}\},\{R_{C2}\},\{R_{C3}\}]$

(d) sequence-set form of the POCEM

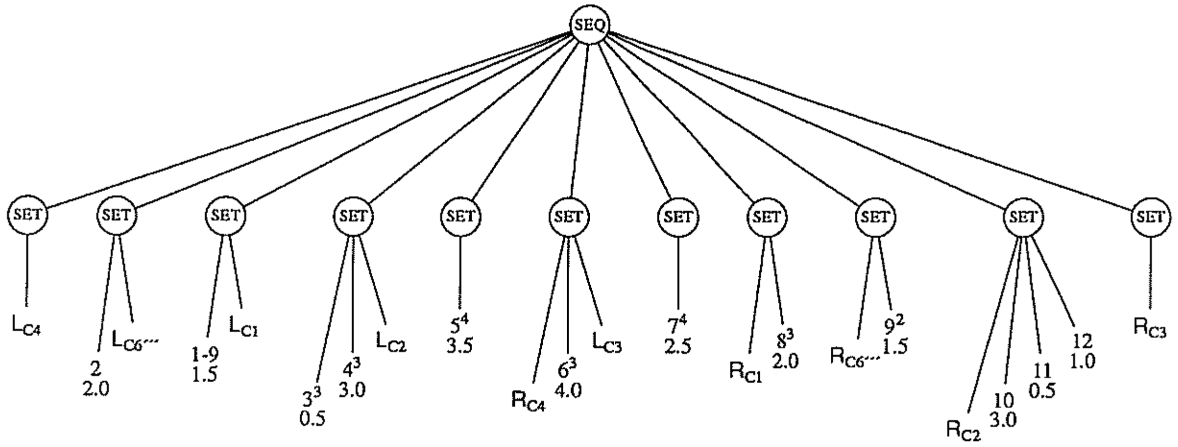
Figure 3.23: ${}_{\alpha}$map12346'''$_2$



(a) graphical form of the PORFM

$[\{L_{C4}\},\{2,L_{C6'''}\},\{1\text{-}9,L_{C1}\},\{3^3,4^3,L_{C2}\},\{5^4\},\{R_{C4},6^3,L_{C3}\},\{7^4\},\{R_{C1},8^3\},\{R_{C6'''},9^2\},\{R_{C2},10,11,12\},\{R_{C3}\}]$

(b) sequence-set form of the POCM

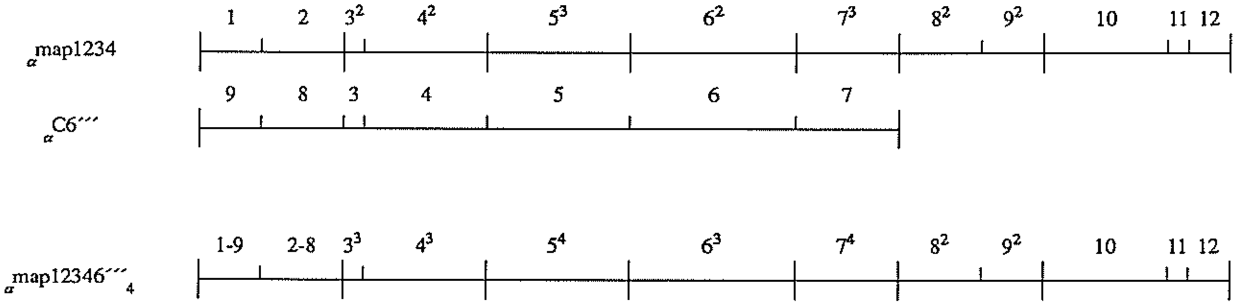


(c) sequence-set-tree form of the POCM

$[\{L_{C4}\},\{L_{C6'''}\},\{L_{C1}\},\{L_{C2}\},\{R_{C4},L_{C3}\},\{R_{C1}\},\{R_{C6'''}\},\{R_{C2}\},\{R_{C3}\}]$

(d) sequence-set form of the POCEM

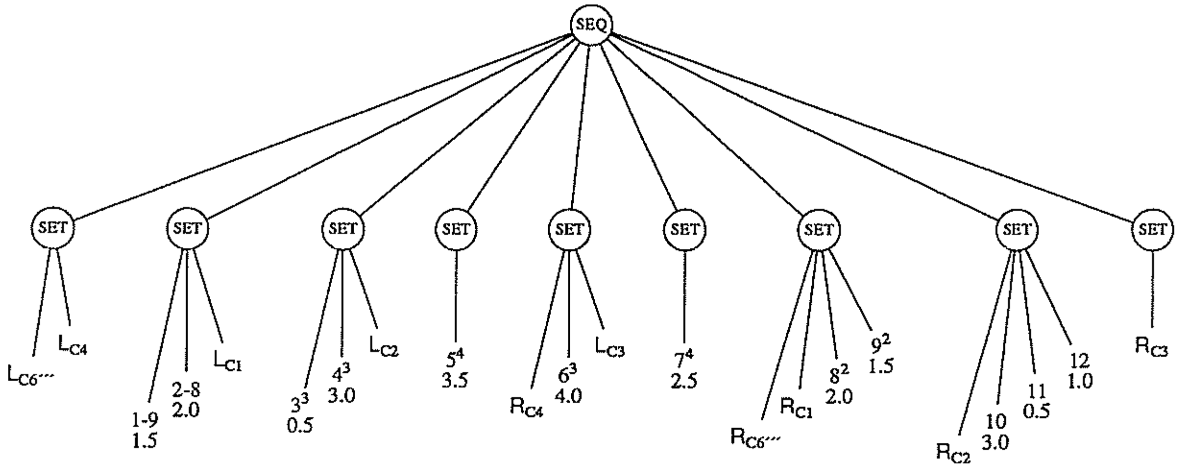
Figure 3.24: ${}_{\alpha}\text{map}12346'''_{3}$



(a) graphical form of the PORFM

$$[\{L_{C6'''},L_{C4}\},\{1\text{-}9,2\text{-}8,L_{C1}\},\{3^3,4^3,L_{C2}\},\{5^4\},\{R_{C4},6^3,L_{C3}\},\{7^4\},\{R_{C6'''},R_{C1},8^2,9^2\},\{R_{C2},10,11,12\},\{R_{C3}\}]$$

(b) sequence-set form of the POCM

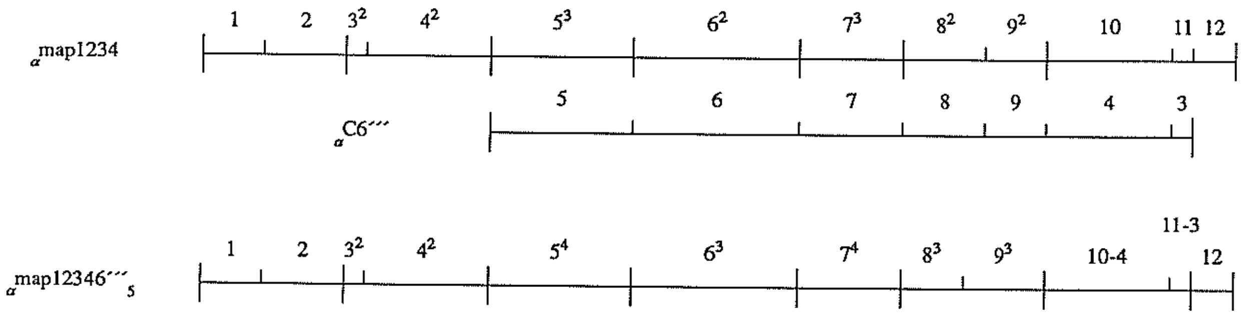


(c) sequence-set-tree form of the POCM

$$[\{L_{C6'''},L_{C4}\},\{L_{C1}\},\{L_{C2}\},\{R_{C4},L_{C3}\},\{R_{C6'''},R_{C1}\},\{R_{C2}\},\{R_{C3}\}]$$

(d) sequence-set form of the POCEM

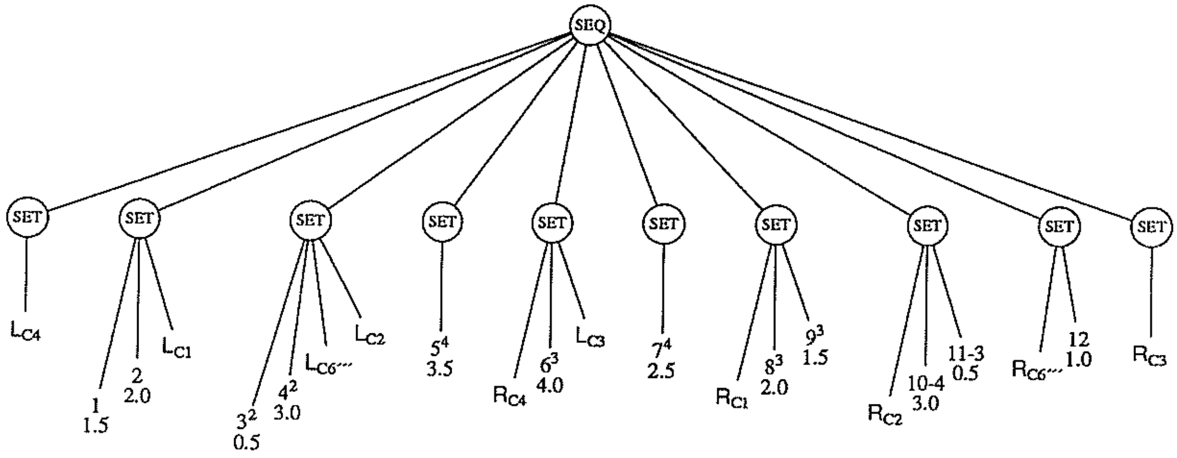
Figure 3.25: ${}_{\alpha}\text{map12346}'''_4$



(a) graphical form of the PORFM

$[\{L_{C4}\},\{1,2,L_{C1}\},\{3^2,4^2,L_{C6'''},L_{C2}\},\{5^4\},\{R_{C4},6^3,L_{C3}\},\{7^4\},\{R_{C1},8^3,9^3\},\{R_{C2},10\text{-}4,11\text{-}3\},\{R_{C6'''},12\},\{R_{C3}\}]$

(b) sequence-set form of the POCM

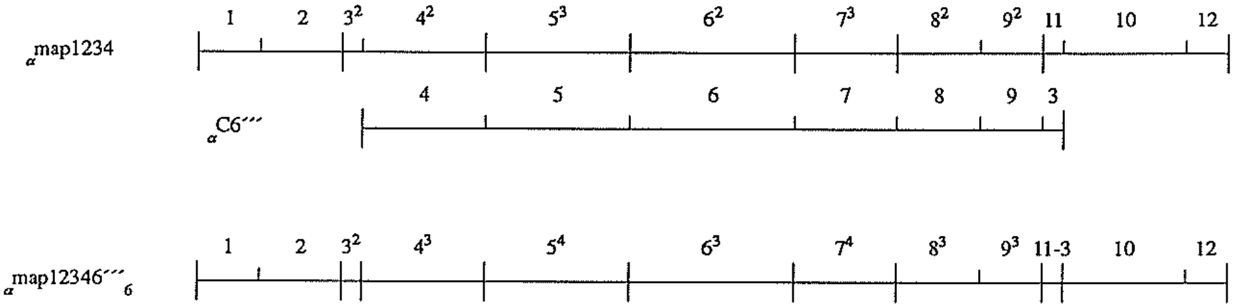


(c) sequence-set-tree form of the POCM

$[\{L_{C4}\},\{L_{C1}\},\{L_{C6'''},L_{C2}\},\{R_{C4},L_{C3}\},\{R_{C1}\},\{R_{C2}\},\{R_{C6'''}\},\{R_{C3}\}]$

(d) sequence-set form of the POCEM

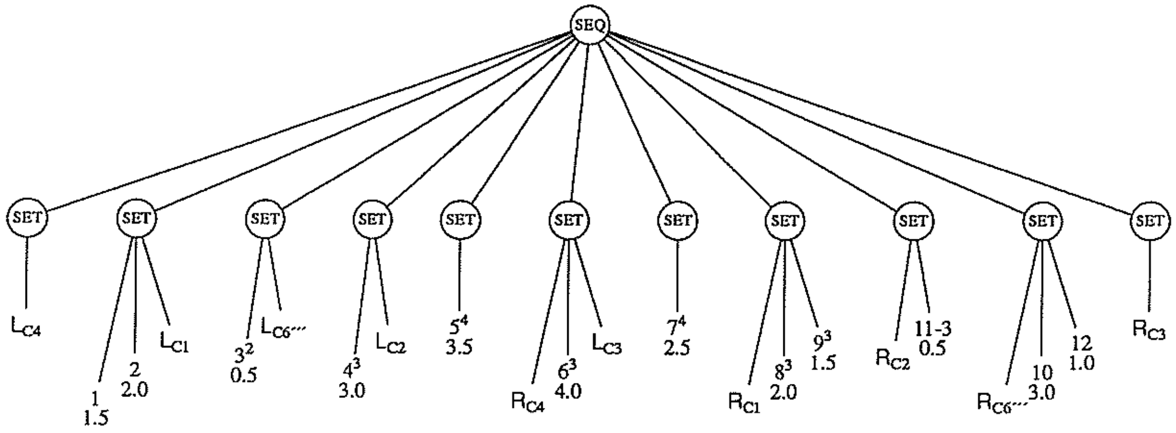
Figure 3.26: ${}_{\alpha}\mathrm{map}12346'''_{5}$



(a) graphical form of the PORFM

$[\{L_{C4}\},\{1,2,L_{C1}\},\{3^2,L_{C6'''}\},\{4^3,L_{C2}\},\{5^4\},\{R_{C4},6^3,L_{C3}\},\{7^4\},\{R_{C1},8^3,9^3\},\{R_{C2},11\text{-}3\},\{R_{C6'''},10,12\},\{R_{C3}\}]$

(b) sequence-set form of the POCM

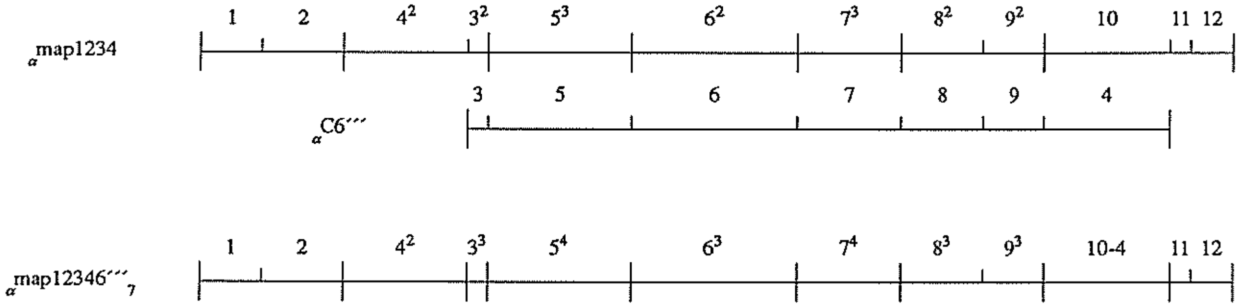


(c) sequence-set-tree form of the POCM

$[\{L_{C4}\},\{L_{C1}\},\{L_{C6'''}\},\{L_{C2}\},\{R_{C4},L_{C3}\},\{R_{C1}\},\{R_{C2}\},\{R_{C6'''}\},\{R_{C3}\}]$

(d) sequence-set form of the POCEM

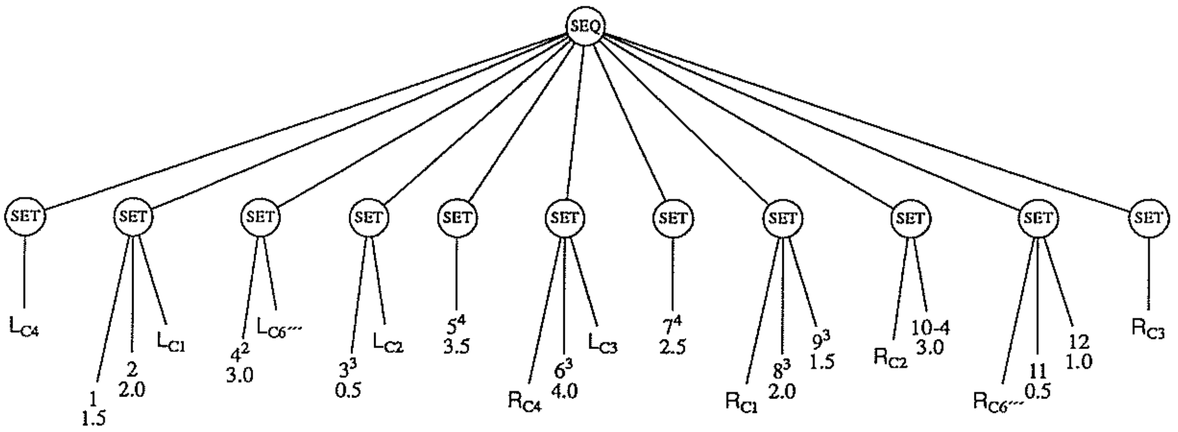
Figure 3.27: ${}_{\alpha}map12346'''_{6}$



(a) graphical form of the PORFM

$[\{L_{C4}\},\{1,2,L_{C1}\},\{4^2,L_{C6'''}\},\{3^3,L_{C2}\},\{5^4\},\{R_{C4},6^3,L_{C3}\},\{7^4\},\{R_{C1},8^3,9^3\},\{10\text{-}4,R_{C2}\},\{R_{C6'''},11,12\},\{R_{C3}\}]$

(b) sequence-set form of the POCM



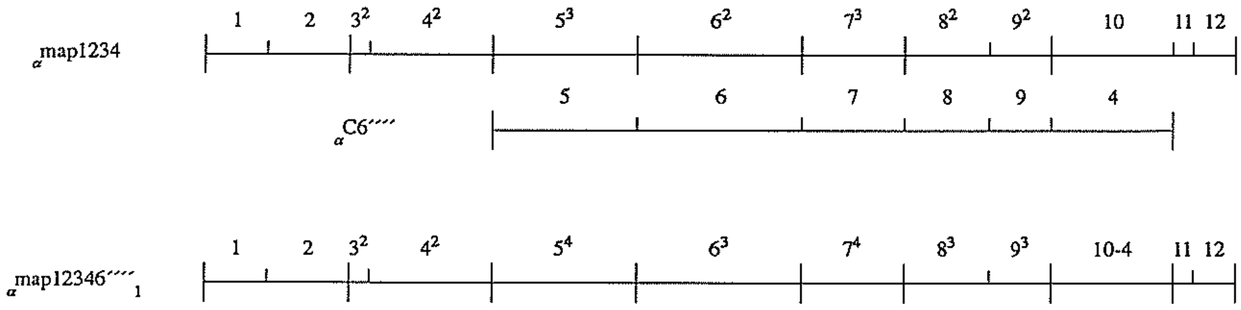
(c) sequence-set-tree form of the POCM

$[\{L_{C4}\},\{L_{C1}\},\{L_{C6'''}\},\{L_{C2}\},\{R_{C4},L_{C3}\},\{R_{C1}\},\{R_{C2}\},\{R_{C6'''}\},\{R_{C3}\}]$

(d) sequence-set form of the POCEM

Figure 3.28: $_{\alpha}$map12346'''$_7$

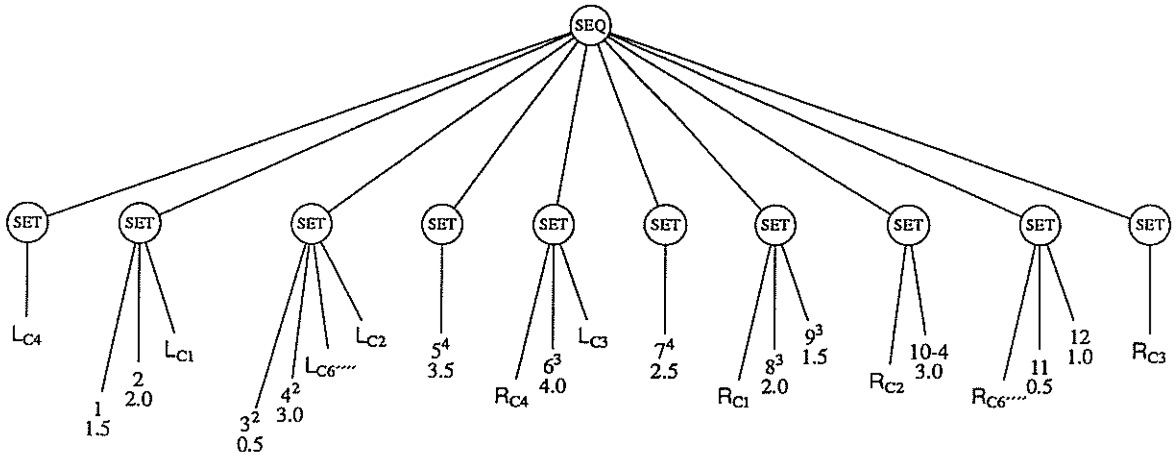
Moving to C6'''' removes fragment 3 from the α digestion. There are two possible incorporations of the α digestion of C6'''' into $_{\alpha}$map1234; these are shown in Figures 3.29 and 3.30. Here, all possible confusion involving fragments 1 and 2 have been excluded, because the absence of fragment 3 makes them topologically infeasible; all possible confusion involving fragment 3 itself is excluded, because it is not present. The POCEMs of both $_{\alpha}$map12346''''$_1$ and $_{\alpha}$map12346''''$_2$ are compatible with those of both $_{\beta}$map12346$_1$ and $_{\beta}$map12346$_2$. Again, there is no unique solution.



(a) graphical form of the PORFM

$[\{L_{C4}\},\{L_{C1},1,2\},\{3^2,4^2,L_{C6''''},L_{C2}\},\{5^4\},\{R_{C4},6^3,L_{C3}\},\{7^4\},\{R_{C1},8^3,9^3\},\{R_{C2},10\text{-}4\},\{R_{C6''''},11,12\},\{R_{C3}\}]$

(b) sequence-set form of the POCM

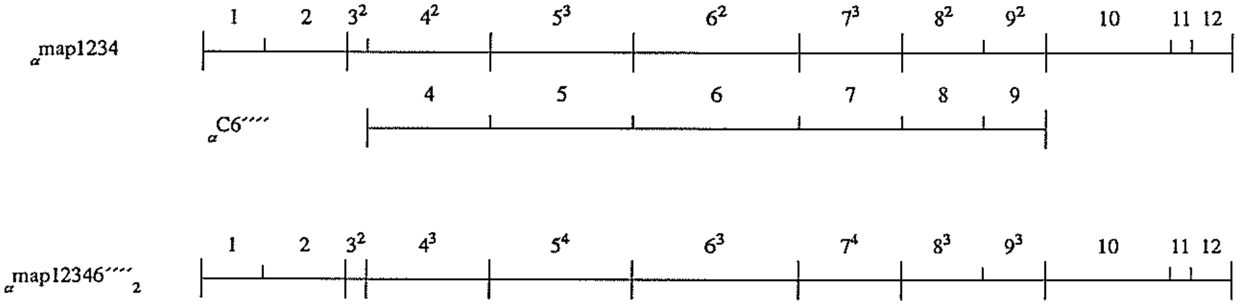


(c) sequence-set-tree form of the POCM

$[\{L_{C4}\},\{L_{C1}\},\{L_{C6''''},L_{C2}\},\{R_{C4},L_{C3}\},\{R_{C1}\},\{R_{C2}\},\{R_{C6''''}\},\{R_{C3}\}]$

(d) sequence-set form of the POCEM

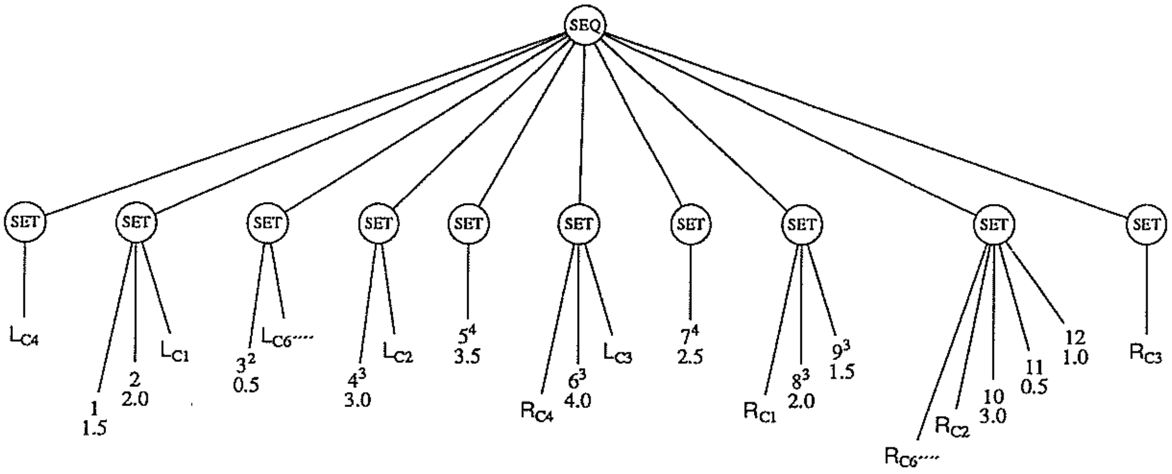
Figure 3.29: $_{\alpha}$map12346''''$_1$



(a) graphical form of the PORFM

$[\{L_{C4}\},\{1,2,L_{C1}\},\{3^2,L_{C6''''}\},\{4^3,L_{C2}\},\{5^4\},\{R_{C4},6^3,L_{C3}\},\{7^4\},\{R_{C1},8^3,9^3\},\{R_{C6''''},R_{C2},10,11,12\},\{R_{C3}\}]$

(b) sequence-set form of the POCM



(c) sequence-set-tree form of the POCM

$[\{L_{C4}\},\{L_{C1}\},\{L_{C6''''}\},\{L_{C2}\},\{R_{C4},L_{C3}\},\{R_{C1}\},\{R_{C6''''},R_{C2}\},\{R_{C3}\}]$

(d) sequence-set form of the POCEM

Figure 3.30: ${}_{\alpha}$map12346''''${}_2$

Moving to C6''''' removes fragment 4 from the α digestion. There is only one possible incorporations of the α digestion of C6'''' into ${}_{\alpha}$map1234; this is shown in Figure 3.31. The POCEM of ${}_{\alpha}$map12346''''' is compatible with those of both ${}_{\beta}$map12346${}_1$ and ${}_{\beta}$map12346${}_2$. Again, there is no unique solution.