DEVELOPMENT OF AN AMINO ACID RACEMIZATION DATABASE FOR COASTAL PLAIN SITES IN NORTH CAROLINA

by

Vincent Pellerito

A thesis submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Master of Science in Geology

Summer, 2004

DEVELOPMENT OF AN AMINO ACID RACEMIZATION

DATABASE FOR COASTAL PLAIN SITES

IN NORTH CAROLINA

by

Vincent Pellerito

Approved: _____

John F. Wehmiller, Ph.D.
Professor in charge of thesis on behalf of the Advisory Committee

Approved: _____

James Pizzuto, Ph.D.
Chair of the Department of Geology

Approved: _____

Mark Huddleston, Ph.D.
Dean of the College of Arts and Sciences

Approved: _____

Conrado M. Gempesaw II, Ph.D.
Vice Provost for Academic and International Programs

ACKNOWLEDGMENTS

Financial support for this thesis was provided through USGS Cooperative Agreement 02ERAG0050 with the University of Delaware. Additional support from the National Science Foundation (EAR9315052) is also gratefully acknowledged.

Much appreciation goes to my thesis advisor, Dr. Wehmiller, for his support and patience while developing this thesis. His willingness to trust my choices for this work is greatly appreciated and his encouragement, while under his wing, is greatly valued.

I would also like to acknowledge the many researchers I worked with from the North Carolina cooperative project. The experience I received at meetings interacting with researchers and simply listening to fascinating geological discussions was invaluable. In particular, I would like to thank Rob Thieler of the USGS in Woods Hole, MA and Bill Hoffman of the North Carolina Geological Survey in Raleigh, NC for their support of my work.

Finally, I would like to recognize the support of my family and friends for their love, humor, constructive criticism and encouragement throughout this time in my life. If I have learned anything these last few years it is that the most important part of living is the relationships you make.

I thank God for molding my heart and mind through these wonderful people.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

An extensive collection of unpublished and published AAR, radiometric and taphonomic characteristic data for mollusk samples from coastal North and South Carolina have been arranged into a relational database. Organizing over two decades of regional AAR data is particularly important for ongoing chronostratigraphic studies of coastal North Carolina, where active study of an extensive Quaternary sequence is underway as part of the North Carolina Coastal Geology cooperative.

A relational database design allows for integrated querying of multiple parameter datasets and ensures the database remains adaptable by removing any dependency on software. We also make use of current data sharing standards for the Microsoft® Windows® platform, employing data analysis software and GIS. Examination of this integrated dataset using advanced visualization techniques should improve understanding of the North Carolina coastal plain stratigraphy and help refine current chronostratigraphic estimates for the region. Furthermore, it builds on efforts to hone the accuracy and applicability of the AAR method as a chronological tool by incorporating numerous analyses over a thoroughly studied region such as the North Carolina coastal plain.

Future endeavors such as web accessibility of the database and possible incorporation into a larger data repository is assisted with proper design early on. In addition, a user-friendly database interface has been developed for continued chromatographic data collection for an active AAR laboratory.

CHAPTER 1

INTRODUCTION

1.1 Statement of Purpose and Objectives

The intention of this work is to support a current investigation to define the Quaternary geologic framework of the northeast portion of the Outer Banks, North Carolina by organizing a regional database of amino acid racemization (AAR) analyses going back since the early 1980's. Since the spring of 2001, the USGS Coastal and Marine Geology Program, regional academic institutions including the University of Delaware Aminostratigraphy Laboratory (UDAL) and public agencies are currently undertaking an effort to map and characterize a thick Quaternary depositional sequence underlying the Outer Banks barrier island-estuarine system. Resolving the complex stratigraphy underlying the Outer Banks would enhance current understanding of regional dynamic shore processes as well as aid in the development of a regional sea-level/climate history along the mid-Atlantic United States. The work from this thesis would not only be useful for the current USGS cooperative study (Coastal Carolina Project) but also for future aminostratigraphy studies of Atlantic Coastal Plain sites.

The principal objective of this work is to organize extensive unpublished and published AAR, radiometric, taphonomic characteristic and other data existing for mollusk samples from coastal North and South Carolina into a database that would facilitate efficient data handling. This work restructures a dataset that includes geochronological data, in particular AAR analyses, for coastal plain sites all along the

Eastern United States (see Wehmiller et al., 1988 for an early form of this database). Restructuring of the database to a relational format became increasingly important as the database grew in size and complexity.

While most of these data are currently available in spreadsheet form, development of a relational database structure allows for integrated querying of a multiple parameter dataset. In addition, utilizing a relational structure allows UDAL to take advantage of industry standards for data sharing and therefore seamlessly incorporate the database with other useful software applications such as a geographic information system (GIS). By reducing the time to collect and manipulate data, informative data exploration efforts are enhanced.

As a result of the new design, the database could also be made available for querying by other interested agencies outside of the Delaware laboratory, such as the Coastal Carolina Project. Efforts to distinguish stratigraphic relationships of erosional/depositional sequences of the Quaternary section of coastal North Carolina are aided by the geochronological data organized within AARDB. Currently, only data for North Carolina, and some of South Carolina and southern Virginia are included in the database (AARDB).

Along with the database product, a development plan is also presented here to aid others with intentions of improving their database design. Numerous texts on designing relational databases are available and some examples of these are listed in the Reference section. This work describes basic relational database principles (Chapter 4) and includes an in depth description of the development processes undertaken for designing AARDB (Appendix I).

1.2 Study Area

At present, AARDB contains over 500 sampling sites, the majority of which are from coastal North Carolina. Figure 1.1 displays all sampling locations currently stored in the database. Members of UDAL have amassed a collection of mollusk samples for this region since the early 1980's. Numerous AAR analyses currently exist and sample analyses for this region continue to accrue. The map exhibited in Figure 1.1 is merely shown to display existing sample locations and the distribution of sampling types such as collections from beaches, cores, excavations and exposures, as well as underwater samples (grab and core samples).

The selection box in Figure 1.1 represents sites of interest for the Coastal Carolina Project and is displayed at a greater scale in Figure 1.2. These exhibited sites are useful for determining the aminostratigraphy of the Outer Banks region (Figure 1.2). To demonstrate the capabilities of the database, data retrieval and application scenarios are presented for this region.

As of the spring of 2003, fourteen Rotosonic drill cores were collected along the barrier island estuarine system, thirteen along the barrier island and one within the mainland region of Dare County, North Carolina (Figure 1.2, OBX and MLD-01 cores). A record of middle to late Pleistocene valley incision and deposition sequences is represented within the Albemarle Embayment of northeast North Carolina (Riggs et al., 1992).

Figure 1.1 General site map showing all sampling locations currently stored in AARDB. Sites are identified based on the sampling site (Inland Core, Offshore Core, Underwater Grab, Excavation/Exposure, Surface or Spoil Pile). Currently all sites in NC studied by UDAL are included.

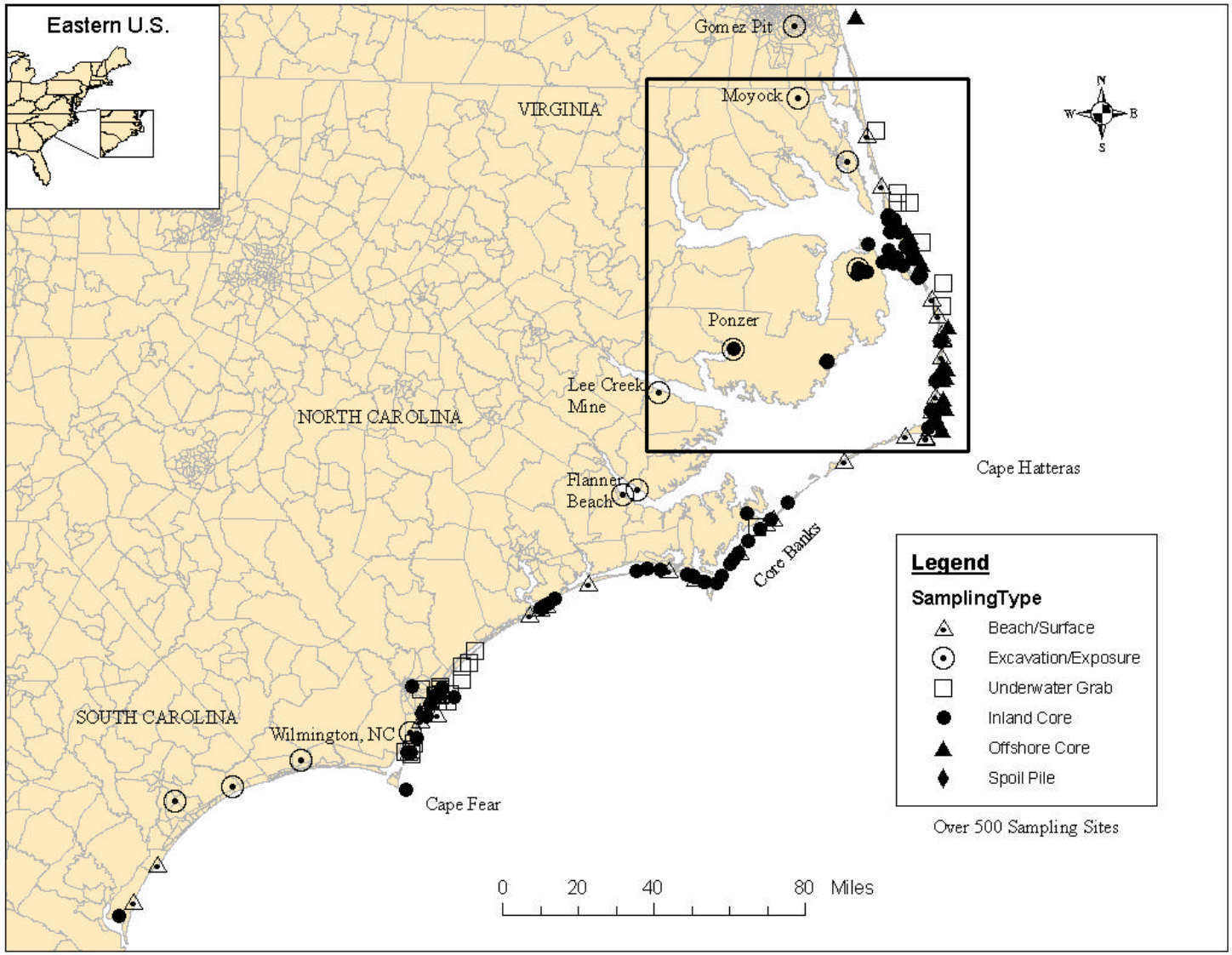
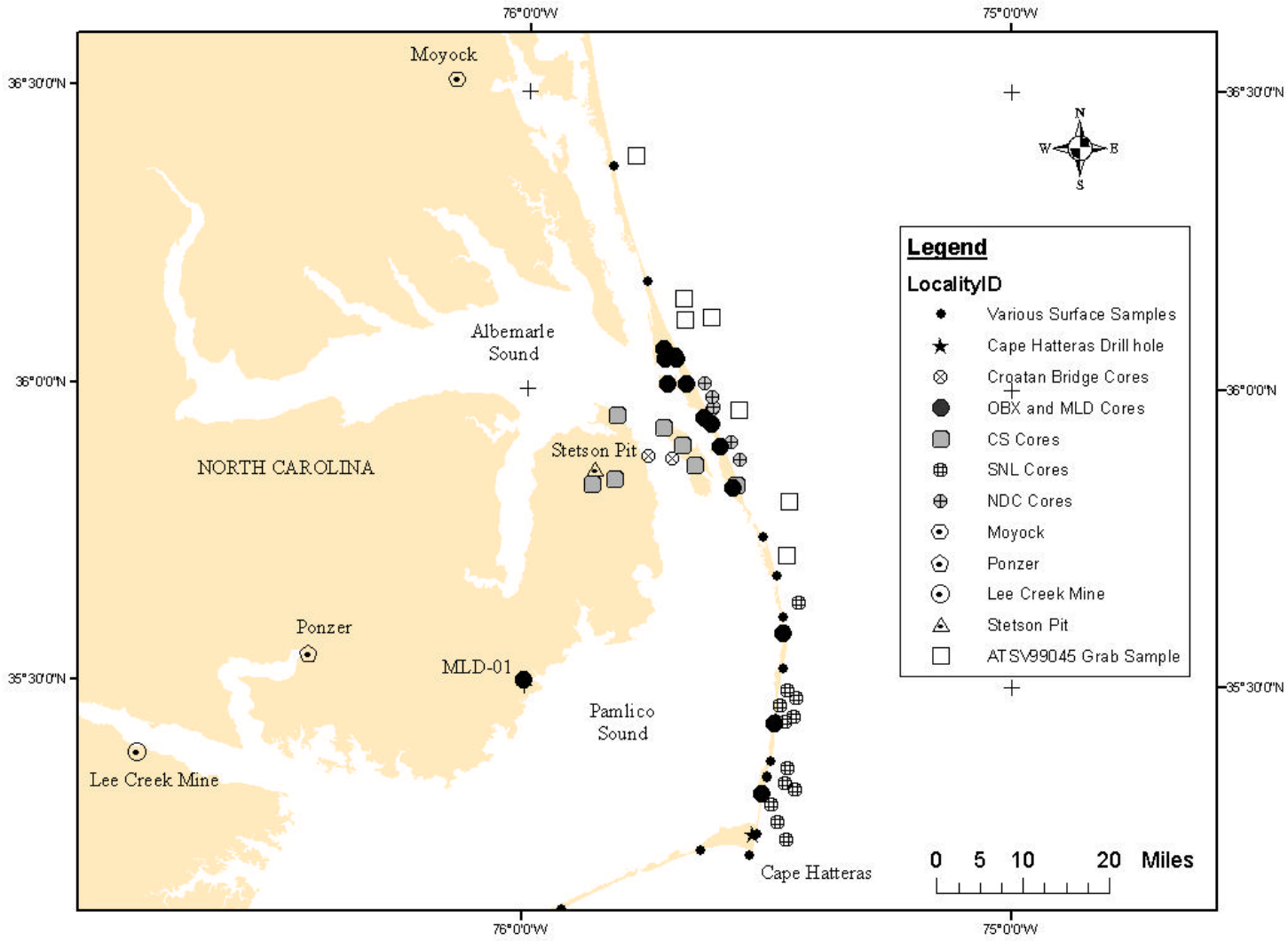

Eastern U.S.
Gomez Pit
VIRGINIA
Moyock
N
W
E
S
Ponzer
Lee Creek
Mine
NORTH CAROLINA
Cape Hatteras
Flanner
Beach
Core Banks
Legend
SamplingType
Beach/Surface
Excavation/Exposure
Underwater Grab
Inland Core
Offshore Core
Spoil Pile
Over 500 Sampling Sites
SOUTH CAROLINA
Wilmington, NC
Cape Fear
0
20
40
80
Miles

Figure 1.2 Map of northeastern North Carolina. Sampling locations specified from the selection box of Figure 1.1. These sites represent all the sites used in statistical and spatial analysis for this work.


76°0'0"W
75°0'0"W
36°30'0"N
36°0'0"N
35°30'0"N
Moyock
Albemarle Sound
NORTH CAROLINA
Stetson Pit
Ponzer
MLD-01
Pamlico Sound
Lee Creek Mine
Cape Hatteras
N
W
E
S
Legend
LocalityID
Various Surface Samples
Cape Hatteras Drill hole
Croatan Bridge Cores
OBX and MLD Cores
CS Cores
SNL Cores
NDC Cores
Moyock
Ponzer
Lee Creek Mine
Stetson Pit
ATSV99045 Grab Sample
0 5 10 20 Miles

The geology of the barrier island region consists of Holocene beach sands perched on a thick sequence of pre-Holocene age basin deposits of the Albemarle Embayment (Riggs et al., 1995). The barrier island complex that makes up the Outer Banks is thin compared to the underlying pre-modern sequence of sediments and constitutes the extent of the sand source for this coastal plain region (Riggs et al., 1995). The underlying geology influences the dynamic morphology of the current barrier island-estuarine system by affecting the shore profile and acting as offshore bathymetric features that dampen incoming wave energy (Riggs et al., 1995).

Riggs et al. (1992) collected a seismic reflection section (Figure 1.3) offshore of Englehard, North Carolina, in Pamlico Sound (offshore of MLD-01 in Figure 1.2), in which eighteen distinct depositional sequences including numerous fluvial paleochannels were recognized (Riggs et al., 1992). The complex Quaternary record represented in the Albemarle Embayment is mostly a product of high frequency eustatic sea level changes caused by global climate variations (Riggs et al., 1992). Age assignments for these depositional sequences were broadly deciphered by such methods as AAR in Riggs et al. (1992).

Current drilling efforts by the Coastal Carolina Project seek to more thoroughly characterize these depositional sequences. Data collected from the OBX, MLD-01 and future cores (see Figure 1.2), along with geophysical imaging of the subsurface, will be used to differentiate mid to late-Pleistocene valley incision and deposition sequences over the course of several paleo-sea level transgressive/regressive events (Thieler et al., 2002).

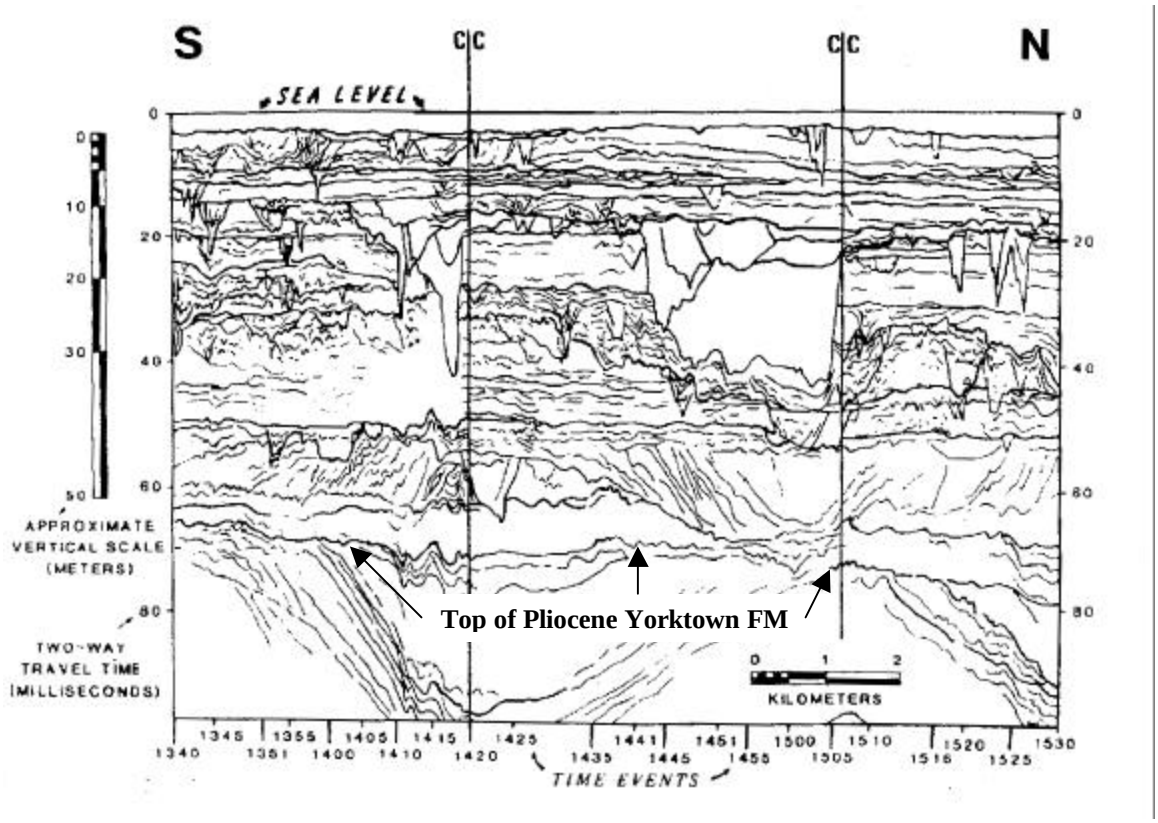


Figure 1.3 Figure 7 of Riggs et al. (1992) showing an interpreted illustration of a seismic line, line S2 in their Figure 1, collected off of the western shoreline of Pamlico Sound. Several stacked depositional sequences are interpreted from this section in Riggs et al. (1992). The top of the Yorktown Formation marks the base of the Quaternary. Modified from Riggs et al. (1992).

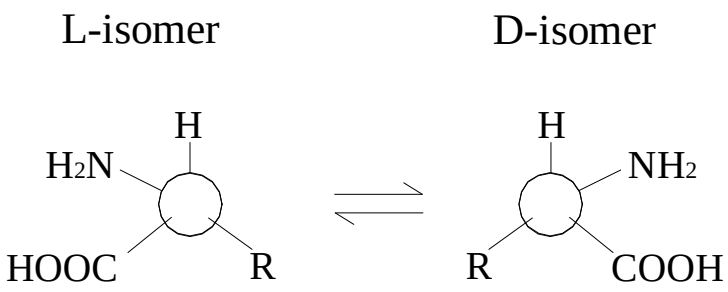
CHAPTER 2

AMINO ACID RACEMIZATION GEOCHRONOLOGY

2.1 Principles of the AAR Method

P.E. Hare (1962) originally proposed the technique of using racemization of amino acids in fossil specimens as a chemical dating method (Wehmiller and Miller, 2000). The method takes advantage of the fact that living organisms contain amino acids that exhibit chirality, that is, isomeric molecules whose mirror images are not superimposible. Chirality occurs for most amino acids that exhibit asymmetry about a central carbon atom with four different side-chains (Wehmiller, 1986) (see Figure 2.1). All living organisms are made up of the "left-handed" (Levo or L-) variety of amino acids. When an organism dies, its amino acids diagenetically alter to their respective enantiomers or the "right-handed" (Dextro or D-) variety in a process called racemization. The ratio of the D- amino acid to its "left-handed" enantiomer within a fossil will increase with time until a point of equilibrium (termed racemic equilibrium) is reached (usually 1.0), within as much as a few million years (Wehmiller, 1993).

For amino acids with more than one central carbon atom more than two molecular forms can exist called diastereomers. For example, L-isoleucine undergoes epimerization (as opposed to racemization) about one of its two central carbon atoms (termed the alpha carbon) to form its diastereomer D-alloisoleucine (Miller and Brigham-Grette, 1989). Racemic equilibrium for D-alloisoleucine/L-isoleucine is typically about 1.3.



-H	Glycine*	-CH_2 CH (CH_3)$_2$	Leucine
-CH_3	Alanine	-CH (CH_3) (C_2H_5)	Isoleucine
-CH (CH_3)	Valine	-CH_2 CH_2 COOH	Glutamic Acid
-CH_2 COOH	Aspartic Acid	-CH_2 (C_6H_5)	Phenylalanine

*Glycine does not exhibit chirality.

Figure 2.1 General structure of D and L enantiomers. Common replacement groups (side chains) and their amino acid names are also listed. Modified from Wehmiller (1984).

Several aspects of the AAR method need to be understood when using it as a geochronologic tool. A basic assumption of the method is that amino acids diagenetically alter in a predictable manner (Wehmiller and Miller, 2000); therefore outside influences on a specimen are considered to by and large alter racemization rates. For example, temperature changes over the course of a sample's taphonomic history will influence racemization rates. Therefore, regional temperature differences are an important feature to consider when applying the method (Wehmiller and Miller, 2000).

In addition, enantiomeric ratios will vary between samples of similar ages but differing genera (i.e., intergeneric differences). This variance means that regional AAR studies should utilize a specific genus or establish before hand the intergeneric relationships of the genera used. Several investigators have worked to establish intergeneric relationships for regional comparisons among several genera (e.g., Lajoie et

al., 1980; Wehmiller, 1980; York, 1990). Wehmiller (1980) concluded that an inverse relationship exists between the amount of Aspartic Acid found in a particular genus of mollusk or foraminfera and the relative racemization rate of a particular genus (Wehmiller, 1980). Consequently, slow racemizing genera tend to have a greater relative abundance of Aspartic Acid (Wehmiller, 1980).

Differences in the apparent racemization rates between amino acids (intrageneric differences) are another complexity of the AAR method. Intrageneric studies on several genera identify relatively "slow" and "fast" racemizing amino acids. Lajoie et al. (1980) determined the order of apparent racemization rates of five amino acids for several genera, concluding that Proline racemizes relatively "fast" and Valine racemizes considerably slower, with Leucine falling between the two with a more moderate racemization rate (Lajoie et al., 1980). Furthermore, through regression analysis, similar to those performed for intergeneric studies, equivalences have been determined for amino acids for several genera (e.g., Lajoie et al., 1980; Wehmiller et al., 1988; York, 1990).

2.2 Presentation of AAR Data

Despite inherent complexities, acceptance of the AAR method has been enhanced by extensive utilization of the technique for a variety of problems (Wehmiller and Miller, 2000). AAR has been successfully applied to problems with calculating rates of tectonic and geomorphic processes along the Pacific coast of the western United States, as well as with framework geological studies deciphering the chronostratigraphy of Atlantic Coastal Plain sediments (Wehmiller and Miller, 2000) (see Wehmiller, 1993 for examples of such applications).

D/L ratios have been presented in several ways, although the most common are formats in which D/L values are plotted with temperature (or latitude) or sample age if AAR results have been calibrated by an independent dating method (Wehmiller and Miller, 2000). Figures 2.2 and 2.3 are taken from Wehmiller and Miller (2000) and represent an internally consistent relationship between D/L values, temperature and time (Wehmiller and Miller, 2000).

Since AAR reactions are dependent on temperature, efforts are made to kinetically model pathways of racemization for differing effective or average temperatures experienced by a sample over its taphonomic history (Wehmiller and Miller, 2000). The simplest way to model the relationship between temperature and racemization rates is through the equation $D/L = k(t)^{1/2}$, where k is the forward rate constant and t is time (Wehmiller and Miller, 2000).

Figure 2.2 graphically displays the exponential relationship between D/L values and time. The trends in Figure 2.2, as well as for Figure 2.3, are kinetic model estimates from laboratory experiments (Wehmiller and Miller, 2000). Isochrons of temperature, as shown in Figure 2.2, display how samples of similar ages would have differing D/L ratios, and therefore exhibit different kinetic trend, when exposed to different temperature histories. Alternatively, isochrons of D/L ratios (i.e., of the same relative age) could also be calculated from the kinetic pathways like those exhibited in Figure 2.3. (Wehmiller and Miller, 2000).

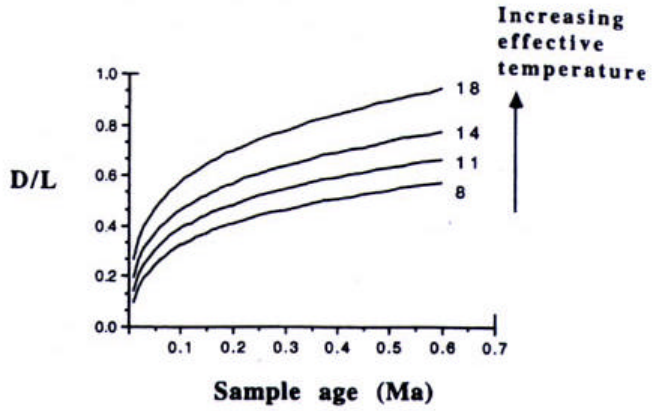


Figure 2.2 Plot of D/L versus sample age. There is a direct relationship between increasing D/L and increasing age of a sample. From Wehmiller and Miller, 2000.

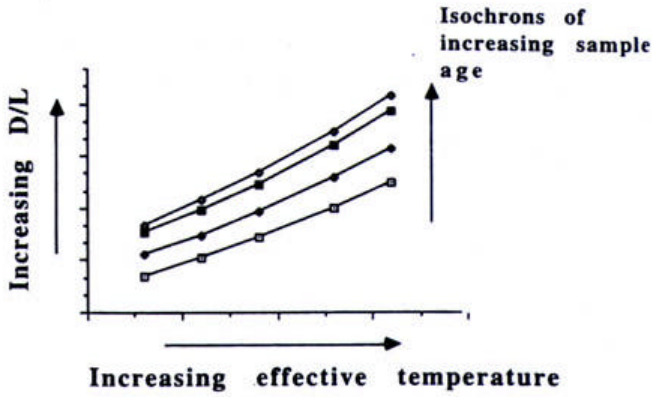


Figure 2.3 D/L Values vs. Effective Temperature. Increase in D/L is directly related to an increase in effective temperature. From Wehmiller and Miller, 2000.

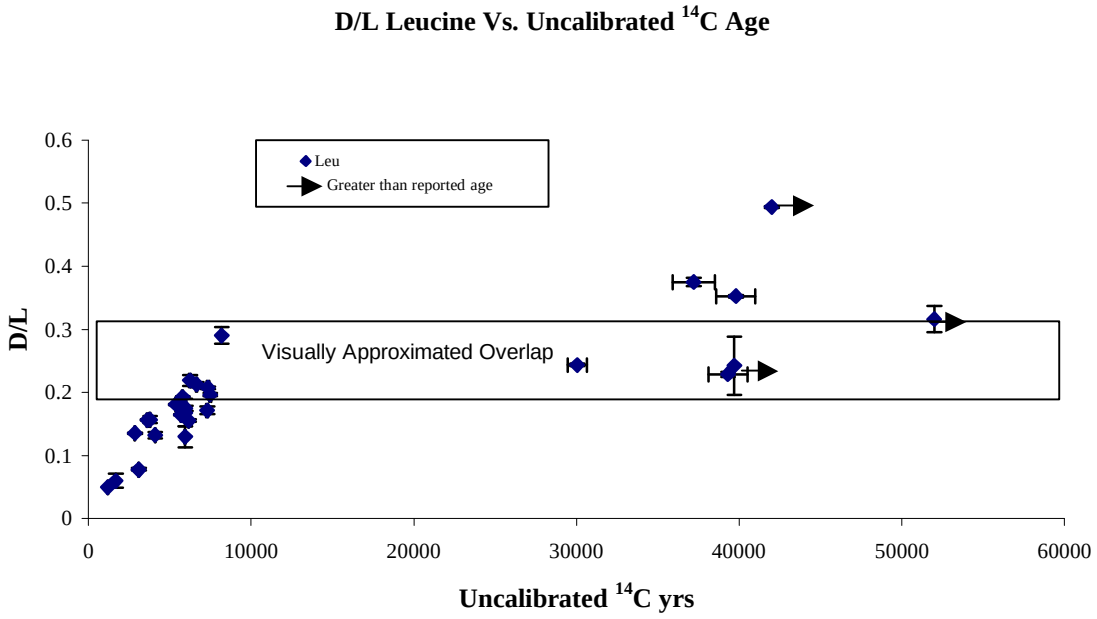


Figure 2.4 Comparing radiocarbon ages with D/L Leucine for northeast NC sites. D/L values exhibit an increasing trend with age (i.e., increased racemization with age). However, an overlap is apparent within the early Holocene/late Pleistocene time frame.

However, considerable spread between data points is common and a consequence of various sources of analytical uncertainty (see Wehmiller and Miller, 2000 for more discussion on assessing the confidence of AAR data). Figure 2.4 is a plot of radiocarbon dated *Mercenaria* samples from northeastern North Carolina. These samples exhibit a good trend of increasing D/L value with increasing age and could be used to calibrate a kinetic model for *Mercenaria* racemization from the region. Nonetheless, the age resolution of such a model would be limited because of the uncertainty exhibited by the AAR method. For example, considerable overlap in D/L values exists for samples of early Holocene to late Pleistocene in age (Figure 2.4). Dealing with the uncertainties of modeling racemization for determining ages is the focus of continuing research (Wehmiller and Miller, 2000).

2.3 Aminostratigraphy

Possibly the most common means to interpret AAR data is by using the D/L ratios of samples retrieved *in situ* as a relative-age tool for stratigraphic correlation. The method, first termed aminostratigraphy by Miller and Hare (1980), is qualitative in contrast to determining geochronology from kinetic pathways of a particular genus. Aminostratigraphy relies on a simple premise; samples collected from the same region would be expected to have similar temperature histories. Therefore, differences in D/L ratios between samples of the same genus could be attributed to sample age. When these D/L values are further constrained by discrete lithostratigraphic units then the expectation would be to find successively older samples with depth.

Use of AAR analyses as a chrononstratigraphic tool can establish the relative age relationships of geologic units where samples are collected. Units are traditionally distinguished based on mean D/L values from an exhibited cluster of ratios. Confidence on whether a cluster constitutes a particular population or aminozone is normally determined by the coefficient of variance (CV) displayed by a particular mollusk genus (Miller and Brigham-Grette, 1989; Wehmiller et al., 1995). Occasionally, studies will also include more rigorous statistical techniques for determining the viability of an aminozone (e.g., see York, 1990).

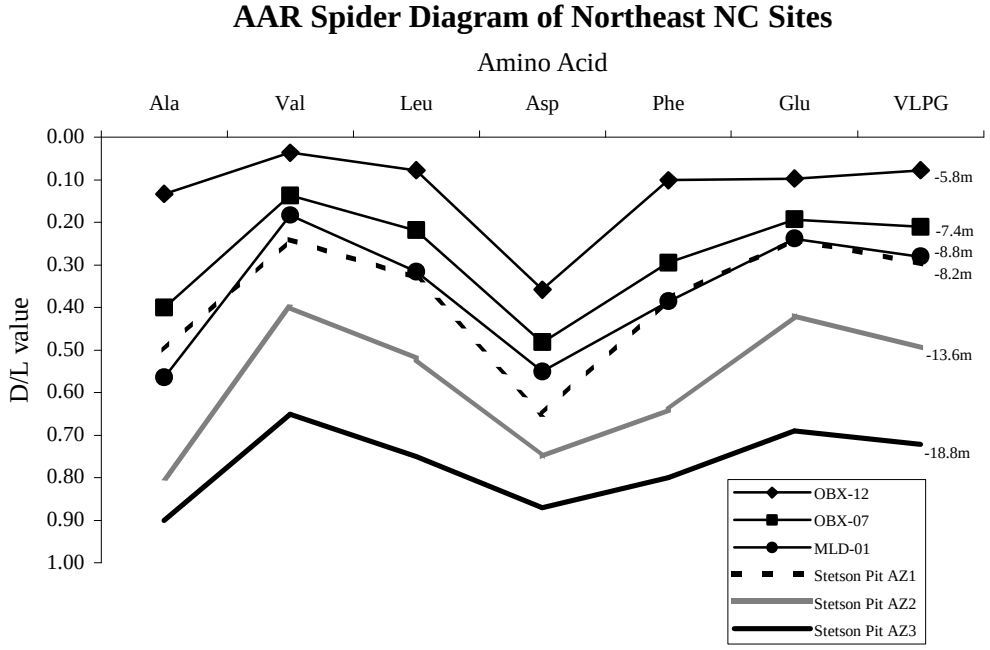


Figure 2.5 Spider diagram showing possible aminozones based on the values of several amino acids. Each data series is an analyzed sub-sample with the approximate sample elevation (MSL) exhibited on the right. The sample from core MLD-01 likely correlates with the upper Stetson Pit late last-interglacial aminozone (York et al., 1989), while AAR values from cores OBX-07 and OBX-12 likely represent later (Holocene, based on radiocarbon data) aminozones. For reference, the earlier Stetson Pit aminozones from York et al. (1989) are also shown.

Comparing several amino acid ratios has long been useful in AAR laboratories for interpreting aminostratigraphic results (Wehmiller, per comm., 2003), though normally just as general guidelines. For example, Figure 2.5 shows a "spider diagram" of AAR analyses and interprets aminozones in the context of several amino acids. Spider diagrams are a useful way to interpret multiple amino acid ratios for a particular sample; however, distinctions made between apparent aminozones are mostly qualitative.

Of course, a site's aminostratigraphy is often complicated by environmental factors. For instance, reworking of mollusk shells from older lithologic units is common along the dynamic barrier island systems of the continental margins (see Wehmiller et al., 1995; York, 1990; Bart, 2001). Continually submerged/emerged landscapes from fluctuations in sea level (local or eustatic) would tend to exhume fossil mollusks, deposit them on a beach and then possibly rebury them in a younger geologic unit, all of which would also obscure their thermal history (Wehmiller et al., 1995).

Nonetheless, where stratigraphically consistent D/L values are determined, the aminostratigraphy of the region can be a useful tool in deciphering the depositional history of the region (e.g., Toscano and York, 1992; Riggs et al., 1992; Harris, 2000). Geologic applications of AARDB are presented in this work to show the databases' usefulness for aminostratigraphic studies along the Atlantic Coastal Plain.

CHAPTER 3

METHODS

3.1 Database Development

Development of the AARDB was undertaken using relational database principles. These principles are outlined in Chapter 4 and are mostly derived from Elmasri and Navathe (2000), *Fundamentals of Database Systems*. The Reference section at the end of this report also lists other texts utilized here for designing the database.

Appendix I of this report is a description of the database development process as it was applied for this work. Data organization issues common for laboratory data are discussed in this section and Appendix I could be used as a guide for researchers with similar data management goals.

3.2 Statistical Treatment of Data

As part of this work, data selected from AARDB were subjected to traditional statistical methods. These methods include typical exploratory methods on sample populations but also a multivariate method used for age discrimination of *Mercenaria* samples. Procedures and assumptions for the statistical methods utilized here are described in Davis (1986). Statistical analyses are mainly presented in this study as examples of data exploration with the new AAR database. Discussion of the applicability of particular statistical methods is left for Chapter 7 (Discussion and Concluding Remarks) of this work.

In this study, discriminant analysis was investigated as a possible method to distinguish between aminozones (particularly Holocene versus Pleistocene age shells) of *Mercenaria* amino acid ratios. Discriminant analysis seeks to quantify the separation between two or more groups (Davis, 1986). Six amino acid D/L values (Alanine, Aspartic Acid, Glutamic Acid, Leucine, Phenylalanine and Valine) are used to determine a discriminant function for distinguishing between fossil shells of two age groups (early Holocene or late Pleistocene). A linear discriminant function is derived from a set of measurements (several variables) that calculates a discriminant score along a line that characterizes the discriminant function (Davis, 1986). Figure 3.1 displays how the discriminant function can be envisioned for two groups from distributions of two variables (i.e., bivariate distribution). The discriminant function can successfully discriminate between two groups even when plotting sample distributions show considerable overlap.

Mercenaria with a D/L Leucine range of approximately 0.15 to 0.36 were employed for the discriminant function calibration sample. This D/L Leucine range was chosen based on the spread of D/L Leucine for northeast North Carolina *Mercenaria* samples of radiocarbon dated late Pleistocene to early Holocene age, as shown in Figure 2.4. This dataset was chosen to limit intrageneric variance apparent between Holocene and all pre-Holocene age samples. For example, Leucine racemization exhibits various trends compared with other amino acids and these trends can change with age for some amino acids (see Lajoie et al., 1980; Kimber et al., 1986; Kimber and Griffin, 1987).

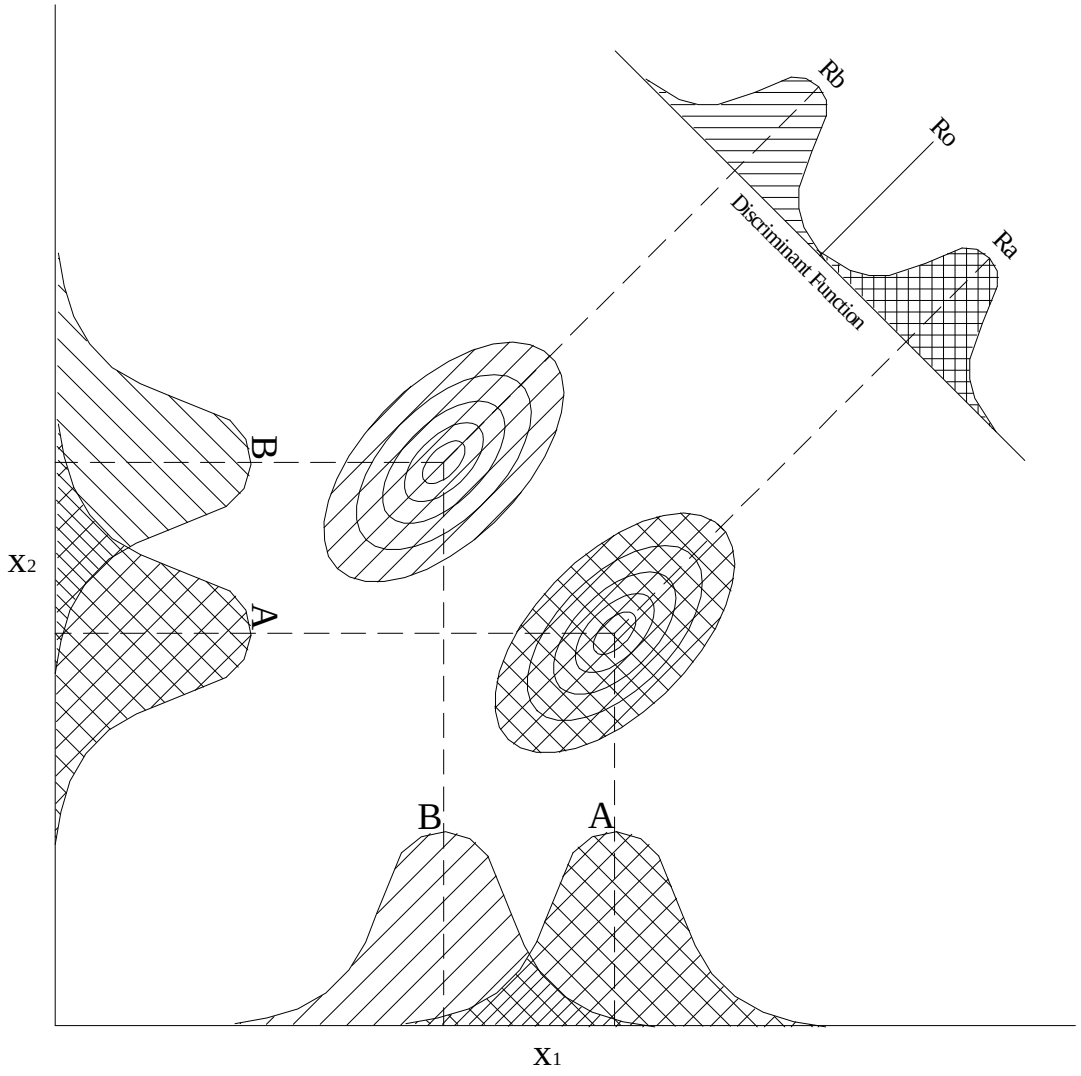


Figure 3.1 Bivariate distribution plot of two groups (A and B) showing overlap for both variables (x1 and x2). Groups can be distinguished by projecting the groups onto the discriminant function line. Ra represents the centroid of Group A, while Rb represents the centroid of Group B. Ro designates the discriminant index. After Davis, 1986.

The calibration sample was also chosen to include enough samples to perform the calculation, as few samples from the early Holocene to late Pleistocene period have been collected for the Atlantic Coastal Plain (Wehmiller, pers. comm., 2003). In addition, a test population of *Mercenaria* was used to assess the calculated discriminant function. Calculations for the discriminant analysis performed for this work, as well as statistical tests of significance and the entire discriminant analysis output as performed in MiniTab version 13, are contained in Appendix II.

Spatial analysis, in particular inverse distance weighting method of spatial interpolation (IDW), is also attempted with data extracted from the database and exhibited in a GIS. The inverse distance weighting method is a good, general purpose contouring method that is used here as preliminary exploration of spatial distributions of D/L ratios. Appendix III contains method properties, plots showing predicted versus actual values and plots of residuals. This information is useful for determining the accuracy of the contouring method.

3.3 Sample Preparation and Laboratory Technique

As part of the Coastal Carolina Project, fossil mollusks collected from cores and beach transect sites of coastal North Carolina were analyzed using the AAR method for aminostratigraphic interpretation of barrier island and estuarine sub-surface deposits of the Outer Banks. AAR Laboratory procedures undertaken as part of this thesis follow sample preparation and amino acid extraction techniques described in York (1984), York (1990) and Wehmiller and Miller (2000).

Shells were chosen for AAR analysis based on their preservation state and sometimes on shell color, based on the assumption that dark colored shells tend to have higher D/L ratios (i.e., tend to be relatively older) (see Wehmiller et al., 1995). These shells were then documented; assigning them each a Sample ID and recording them into the AAR database. Some shells analyzed came from an archived collection so UDAMS numbers (University of Delaware AMinoStratigraphy geographic location identifier) were previously allocated.

Sampling of these specimens entailed cutting a fragment through the umbo of the shell, near the valve hinge (Figure 3.2) with a rock saw. Samples (i.e., sub-samples) were collected from the umbo to achieve results consistent with other *Mercenaria* analyzed from the Atlantic Coastal Plain (e.g., see Belknap, 1979;York, 1984 and 1990; Bart, 2001). Each fragment cut was then assigned a sub-sample ID (AAR Lab Number) and recorded in the database. For further documentation, the shell and a 5x8 index card displaying identification numbers were photographed and stored in the database.

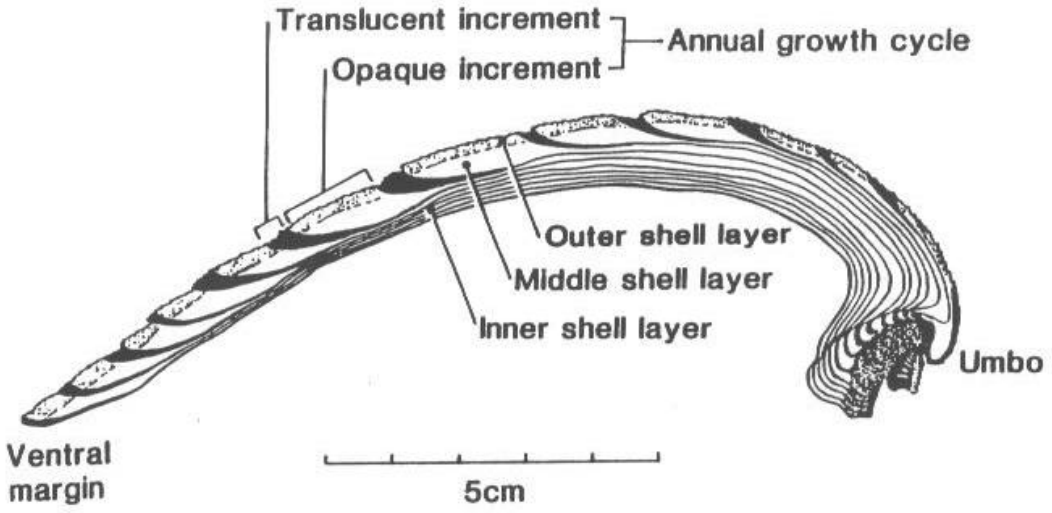


Figure 3.2 Cross-sectional view of a *Mercenaria* marking the umbo and the middle carbonate matrix layer.

Each fragment cut from a valve was trimmed and abraded (i.e., mechanically cleaned) to remove the surface carbonate layer that may have experienced diagenetic alteration. Sometimes a small fragment (ca.1.0 gm) needed to be abraded further by a more delicate tool. For this task a dental rotary drill with carbonundrum grinding disc was used. Throughout the mechanical cleaning stage, the shell fragment was continuously cooled with tap water to diminish any frictional heat generated from the saw or rotary drill. Additionally, latex rubber gloves were worn at all times to avoid direct contact with human skin. The final carbonate sample for each shell represented the middle shell layer (Figure 3.2) and, again, is consistent with other workers (e.g., Belknap, 1979; York, 1984 and 1990; Bart, 2001).

After mechanical cleaning, each cut fragment was transferred to a glass test tube for a delicate chemical cleaning. This process entailed submerging the fragment in the test tube with distilled water and slowly adding dilute hydrochloric acid (HCl). When a gentle fizzing was achieved the sample was allowed to sit in this solution for about one minute. Then the dilute HCl was decanted and distilled water was again added while swirling the test tube to ensure a thorough rinse. The process was repeated several times until about 10% to 30% of the fragment had been dissolved, ensuring that no outer carbonate material remained on the sample (Wehmiller and Miller, 2000). Subsequently, the test tubes along with shell fragments were covered with aluminum foil and placed in a vacuum oven on low heat for one hour to dry up any moisture.

Next, the weight of each shell fragment needed to be documented. From this stage on care was taken not to handle the fragment to avoid possible contamination. Samples were weighed to the nearest 0.001 grams and transferred (without handling) to screw cap vials.

At this point the samples need to be completely dissolved in preparation of being hydrolyzed. Concentrated HCl is added to the fragments in the screw cap vials at 0.1 ml increments so that the product does not boil over. The amount of HCl added is proportional to the weight of the dried carbonate material, approximately 3.5 ml/gm of carbonate for total amino acid analysis. Once the samples were dissolved, the vials were purged of air by forcing inert Nitrogen gas (N_2) into the vials. The vials were then capped, sealed tightly and placed in a heating block at 110°C for 22 hours.

Following hydrolysis, the samples were desalted using hydrofluoric acid (HF). The hydrolyzates were transferred by pipette to plastic centrifuge test tubes. Approximately 1.25µl/gm of carbonate was added to each test tube and swirled repeatedly. Then the test tubes were centrifuged for approximately 25 minutes to separate calcium fluoride (CaF_2). The supernatant liquid was transferred by pipette to new polycarbonate round bottom test tubes and dried under a stream of N_2. Once the liquid had been thoroughly dried, the residue in the polycarbonate test tubes were taken up with 1M HCl, transferred to a screw cap vial and dried again under a stream of N_2.

To make the derivative containing the amino acids, reagents are added for esterification and acylation of the amino acids. An anhydrous sample is required for this step so just prior to making the derivative the vials were placed in a vacuum oven on low for about one hour. Approximately 0.75 ml of isopropyl/HCl was added to the vials. The vials were then capped and placed in a heating block at 105°C for one hour. Next, the samples were dried down under a stream of N_2 for about five minutes and approximately 1.0 ml of dichloromethane (DCM) and 0.2 ml of trifluoroacetic acid (TFA) were added for the acylation step. The vials were then capped and allowed to sit in a plastic jar filled with desiccant for at least 2 hours to complete the reaction.

To complete the derivatives and prepare them for storage the reagents in the vials were dried down under N_2. The liquid was brought down to about 50% of its volume and then transferred to a smaller screw cap vial. Then the remaining liquid was dried down with N_2. Cyclohexane was then added to the vials at about 0.1ml /0.3 gm of carbonate. The vials are then placed in small jars filled with desiccant and stored in a dedicated

refrigerator to await GC analysis by injection. A sample chromatogram is included as Figure 3.3 to show the relative amino acid peak retention times.

Generally, several injections were executed to establish mean D/L values for a particular laboratory aliquot. Therefore, mean values are reported with standard deviations and the number of injections included in the calculation. Calculated D/L values along with simple statistics for samples exhibited in this thesis can be queried using the database file (aardb.mdb) on the included CD, Appendix IV.

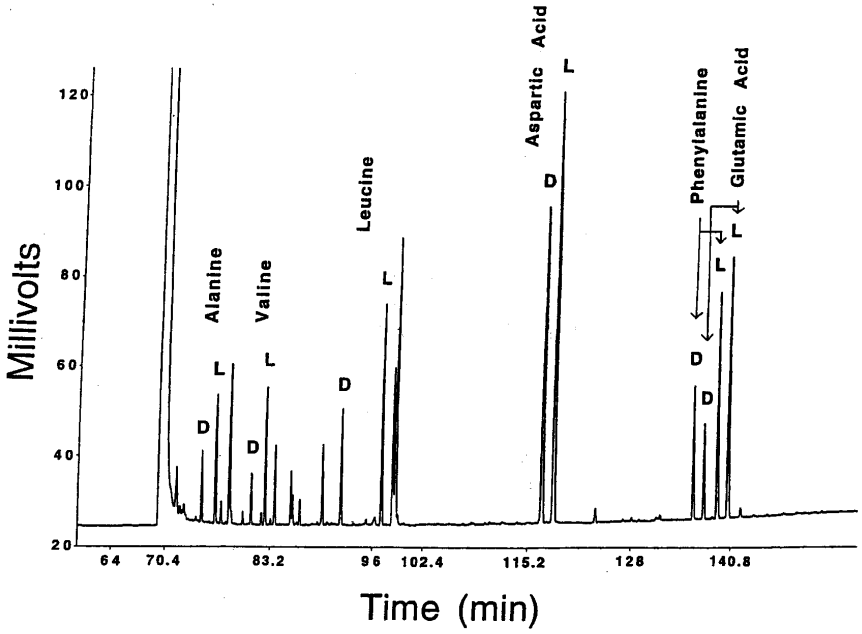


Figure 3.3 An example of a chromatogram from GC analysis from a Pleistocene shell. Amino acids enantiomers are labeled in the figure. AAR D/L values are determined by calculating the ratio of the areas from both enantiomers of an amino acid.

CHAPTER 4

PRINCIPLES OF DATABASE DEVELOPMENT

4.1 Overview of Data Management in the Geosciences

Within the geosciences there has been an historical commitment to long-term data archiving and data management such that today geoscience organizations, whether governmental or academic, have shown a willingness to embrace new technologies and work them into research. For instance, the International Council for Science (ICS) established the World Data Center (WDC) in 1957 (Allen, 1988). WDCs are data repositories of geophysical and solar data that promote efficient data sharing internationally between researchers (Allen, 1988). In addition, the ICS currently sponsors and facilitates short courses and degree programs for teaching data management issues in marine geoscience (Dittert, April 22, 2003). These programs, which are spread internationally among several universities, seek to educate students on data management issues within the geosciences (Dittert, April 22, 2003).

Likewise, government agencies like the USGS and the Federal Geographic Data Commission (FGDC) have taken lead roles in utilizing the latest advances in information technology and mainstreaming such technologies for public access of geological data. For example, the National Spatial Data Infrastructure (NSDI), under the auspices of the FGDC, seeks to improve the accessibility and use of geospatial data through programs

that share the responsibilities of geospatial data creation and maintenance (National Research Council, 2001).

During the 1980's much of the data management work in the geological sciences focused on developing database management systems within the structure of traditional geological data banks. Simply due to the multi-variable nature of geologic information, these data banks often consisted of a collection of records representing observations with multiple dimensions or columns characterizing the myriad of geological parameters. Therefore, data management schemas were specialized to meet the needs of a particular discipline in geology with data processing applications often a component of the system (see for example Young, 1982; Coffey et al., 1982; Fletcher, 1987). However, with changes in data presentation or advancements in software applications, problems of interoperability and obsolete data structures became apparent.

E.F. Codd's 1970 landmark paper on relational database design discussed these and other such issues for data management systems of the time. He presented a relational data model and its corresponding calculus for shared databases that eliminated their dependency with application programs. That is, data in a database was finally independent of the software used to manage and query the database. Changes within the business community proceeded from Codd (1970) such that today the relational model is the standard for all commercial database management systems (Elsmasri and Navathe, 2000).

However, changes were not apparent in the geoscience community until later. For example, at the first meeting of the International Geographical Union (IGU) in 1988, the USGS announced its intention to utilize "new" technologies such as relational database management systems (RDBMS) in its global database planning efforts (Hill and Walton, 1988). By the mid 1990s, relational databases in the geological sciences became more common for multi-user systems, though the relational model was not as prevalent for single user systems (see for example Gunderson, 1994, noting database vendors and their capabilities).

Today, with the development of database systems on the personal computer (PC) platform, such as Microsoft® Access, single-user relational systems have become more prevalent and are often used as front-ends to database systems stored on a non-PC platform (Elsmasri and Navathe, 2000). Data from government agencies, research groups or single researchers can be created on stand-alone databases and connected through data sharing methods (e.g., Microsoft® Open Database Connectivity (ODBC) standard) and thus facilitate data sharing within a research community.

4.2 Comparison Between Flat-File and Relational Format

Much of the data collected in geological studies, at least traditionally, have been stored in flat-file format. A database in a flat-file format is essentially one data file or table that can have numerous columns representing data parameters and rows representing a record in the database. Such formats are useful because they display data values for essentially an unlimited number of parameters within a single view and, more importantly, graphing or manipulating the data is made easy by simply comparing one or more columns. Additionally, maintaining a flat-file format database usually incurs little

overhead cost because the technologies for data storage and application software are readily available for personal computer formats and are reasonably understandable by the layperson. However, with a flat-file format, data management problems inflate as a database becomes more complex. Data relationships are complicated as more parameters are stored and required for analysis.

An increase in data volume also suggests a need for a RDBMS. For example, a Microsoft® Excel worksheet has a maximum storage capacity of 65,536 rows by 256 columns; time series data and most geophysical data can quickly exceed such a capacity after only a few acquisition events. Inefficient querying and obsolete data also become more apparent as the size of the database increases and redundancy increases from data sharing. Table 4.1 below highlights some major drawbacks and strengths of the flat-file format as compared with the relational format.

In contrast to the flat-file format, the relational format has more sophisticated aspects that make it advantageous for data management. Firstly, data are structured in such a way that the semantics, that is, the intention or meaning of the database, are understood intrinsically (Codd, 1970). Querying of the database takes advantage of relationships established between data tables. Moreover, data standards and rules are defined explicitly and enforced to ensure data integrity through an innate, declarative language, like SQL (Structured Query Language) (Elmasri and Navathe, 2000). SQL is also the language used for data querying and has become an industry standard for database management systems. With a common standard, data sharing between different software platforms is made simple.

Table 4.1 Comparing capabilities of a flat-file database and a relational database format. Modified from Hoffman, 2003.

	FLAT-FILE DATABASE	RELATIONAL DATABASE
Querying Data	Search application explores the entire file to give a result. Some applications allow for data indexing.	Uses SQL to efficiently search several related data tables.
Data Integrity	Some enforcement of data standards available through popular software though not implicit in the database.	Data integrity enforced through implicit data structure and constraints, which is stored in a system catalog or data dictionary for reference.
Updating Data	Files are only as current as when the file was last modified. Updates are normally executed manually to all pertinent fields and associated files.	Tables are always current. Updates are performed directly on the contents of a table and propagate throughout all related tables.
Data Sharing	Although files can be easily shared, concurrency of data cannot be assured. If an Applications must be developed to enforce concurrency control for a centralized file system.	Database truly centralized. Supports multiple access and views to data through concurrency control and recovery subsystems of a DBMS (Elmasri and Navathe, 2000).

A relational data model also eliminates storage of redundant or null data that is necessary in a flat-file format to maintain associations between several data parameters. It removes this necessity by the use of a primary key that uniquely identifies records in a particular table. Furthermore, a relational database is advantageous because it allows the casual user of the database to retrieve data with ad hoc queries without knowing how the data are structured. Quite often for databases stored in a flat-file format, only the database creator or principal user understands where particular data exists and how to retrieve pertinent information.

4.3 The Relational Data Model

4.3.1 Overview

Several data models or schemes exist for database systems. Some of the more notable ones are the hierarchical, network and relational data model. Codd (1970) presented the relational data model as an approach to data management that "protected" users and application programs from disruption of activity if the internal structure of the data were modified (Codd, 1970). Such modification could occur as a result of dataset expansion or altering of usual data access paths (Codd, 1970). Codd's relational model organized tabular data in a form most efficient for data retrieval and manipulation. Today, most database management systems utilize the relational model. However, the hierarchical model is still used commercially, including for operating system file structures (e.g., personal computer file systems).

In presenting the relational data model, we first need an appropriate definition of a database. A database is generally defined as an organized body of related information (The American Heritage® Dictionary, 2000). Elmasri and Navathe (2000) elaborate on this definition describing a database as:

- A collection of logically coherent data.
- Representing some aspect of the real world.
- Made for a specific purpose.

Therefore, we can think of a database in terms of differing structural formats (e.g., relational, hierarchical, network, etc.) but its essence should exhibit the above characteristics.

When creating a database, often the database software is considered as the same object. However, it is more precise to refer to the database as the physical storage of data files, while the software used to manage the database as part of a larger database management system (DBMS). Database software is utilized for managing the data in the database. Likewise a software application that a user interacts with is separate from the actual database but part of the larger database system. Figure 4.1 illustrates the general components of a database system. Whether we are referring to a relational or hierarchical database, a DBMS is a collection of programs that enables users to produce and maintain a database (Elmasri and Navathe, 2000).

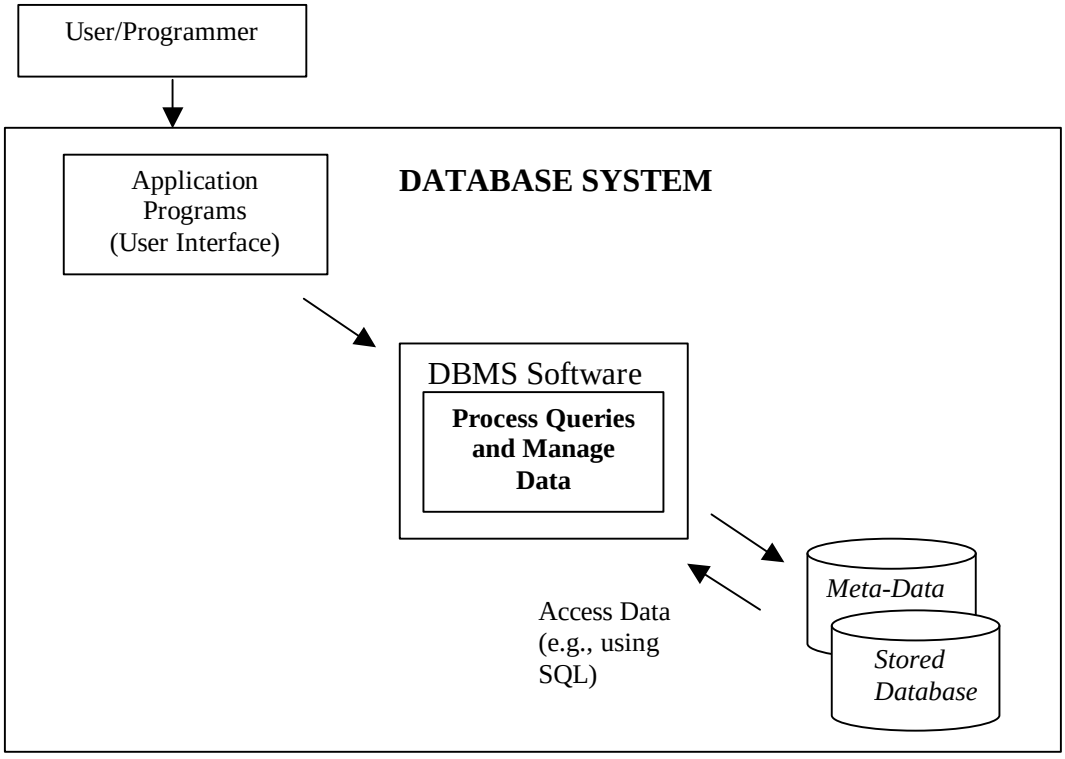


Figure 4.1 Diagramatic representation of a DBMS. A DBMS is a collection of software programs that enable a user to create and maintain a database. After Elmasri and Navathe, 2000.

4.3.2 The Relation

A relational database is generally thought of as a collection of tables where each table or *relation* represents an entity in the real world. Likewise, a record in a table represents a "fact" corresponding to that real world entity and the table name and column (attribute) heading clarify the meaning of each record (Elmasri and Navathe, 2000). Within an attribute, a *domain* is specified and is designated by the data type of its values and their format (Elmasri and Navathe, 2000). For example, an attribute that contains employee social security numbers would consist of a data type of all integers with the form 000-00-0000. The domain of this attribute is therefore, described by an all integer data type and the special digit format for its set of values. Likewise, specifications such as whether to allow null values are part of the domain of an attribute.

A relation is not strictly a table and therefore a database a collection of tables or flat-files. More formally, a relation has characteristics that make it different from a table. For example, with tables data are physically stored in a certain order. The relation concept does not require an order to its records or *tuples* as a tuple can be described by any ordering of an attribute name and its value (Figure 4.2) (Elmasri and Navathe, 2000). For this work it is sufficient to refer to relations as tables. In addition, the term *entity* is used to refer to a relation, and therefore, a table. These terms are used interchangeably in this thesis, although relation is used most often in its formal sense.

Table.Record = Employees.<(Name,Joshua Parker),(EmployeeID,145-79-0475),(BirthDate,10-20-1976),(HomePhone,302-831-6602)>
Table.Record = Employees.<(BirthDate,10-20-1976),(EmployeeID,145-79-0475),(HomePhone,302-831-6602),(Name,Joshua Parker)>

Figure 4.2 Identical rows showing that record ordering is not necessary when the attribute name is included with its value.

A relation is described by its name as well as by the attributes represented in the relation. A relation is designed such that an identifying attribute or the primary key is unique for a particular record in the table. Therefore, nulls or empty values cannot be recorded in the primary key because nulls are not unique. However, there may be several attributes in a table that could be used as a primary key (termed candidate keys), though only one or a combination of keys is designated as the primary key. Primary key and domain constraints within a relation constitute what are referred to as *entity integrity* constraints of a relational database. They restrict the kinds of data entered into a table and ensure that each record in a table is unique.

Figure 4.3 displays the components of a relation. There are three candidate keys displayed in Figure 4.3, EmployeeID, SSN, or Name if we assume that no two employees will have the exact same name. In the example, EmployeeID is identified as the primary key (designated with an underscore) even though SSN or employee Name could be an appropriate primary key.

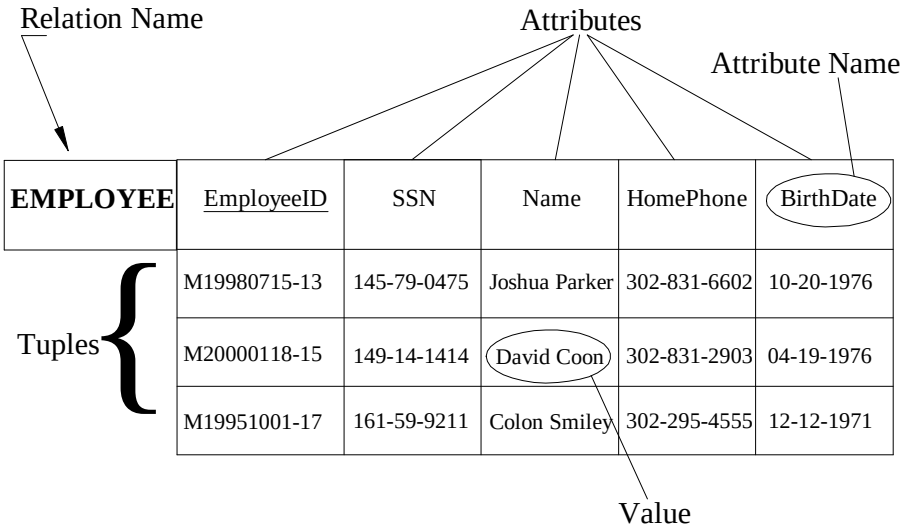


Figure 4.3 Components of a relation expressed as a table. After Elmasri and Navathe, 2000.

4.3.3 Table Relationships

With a primary key identifying each record within a table, several tables can be related based on the representation of a primary key in other tables. For instance, Figure 4.4 shows three tables, EMPLOYEE, DEPARTMENT and PROJECT that could be part of a company database (COMPANY). The primary key for each table is underlined and associations of unique records between tables are depicted by the existence of a *foreign key* in a table. A foreign key in a *child* table is the primary key of a *parent* table and exhibits the same domain as the parent's primary key. Table EMPLOYEE contains the attribute DeptID that references DeptID in DEPARTMENT (Figure 4.4). Likewise, DeptID is represented in the table PROJECT.

These key associations create the *referential integrity* inherent in a relational database; that is, they maintain record consistency between tables. Referential integrity

ensures that table relationship rules are adhered to. For instance in the example of Figure 4.4, a person exists in the COMPANY database as an employee working under a certain department; a department, which can have several employees, is responsible for a particular project. An employee can only work for departments represented in DEPARTMENT and PROJECT's association with DEPARTMENT specifies the job that an employee works on. Therefore, table relationships allow for updating or modifying the database within constraints specified by the requirements of the database.

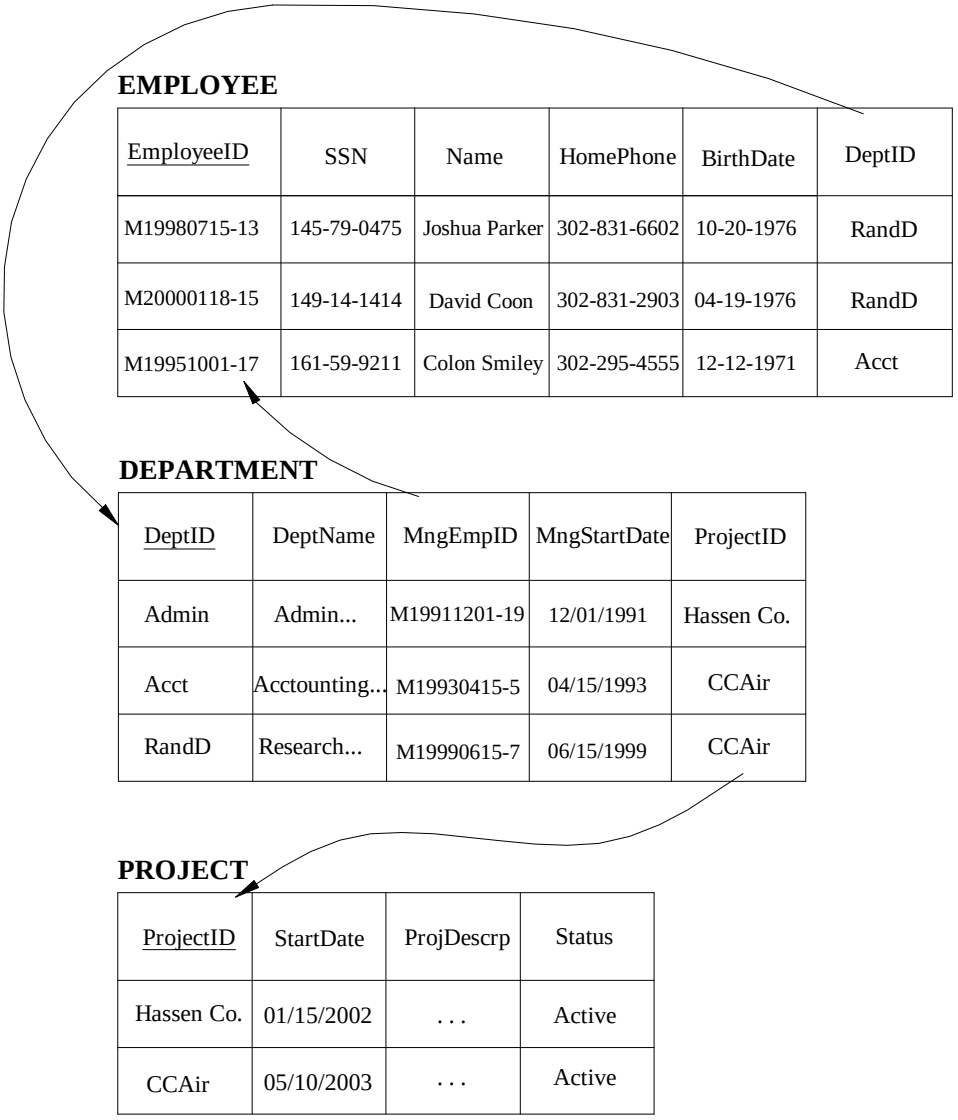


Figure 4.4 COMPANY database tables. Tables show how data could be stored with referential integrity constraints signified by leaders pointing to primary keys. The primary keys for each table are underlined. After Elmasri and Navathe, 2000.

4.3.4 Relationship Cardinality

The discussion above on referential integrity alluded to another aspect of relational databases, the concept of relationship *cardinality* or the set of all possible combinations of values between relations (Elmasri and Navathe, 2000). There are essentially four cardinality ratio possibilities between tables, one-to-one, one-to-many, many-to-one and many-to-many. The cardinality ratio between tables for a database would depend on the specifications of the database established ahead of time.

In our example from Figure 4.4, the requirements of the database may specify that an employee may not work for more than one department but that a department can employ several employees. The relationship between DEPARTMENT and EMPLOYEE would then be one-to-many (1:N). Generally, many-to-many (M:N) relationships in a database schema need to be resolved to some combination of 1:N possibilities. Creating a new entity that includes the primary keys of both entities in the M:N relationship often solves this. An example of this is given in Figure 4.5 as the new entity WORKS_ON, which resolves the M:N relationship between EMPLOYEE and PROJECT.

Figure 4.5 shows the relation schema for the COMPANY database as it might be designed. Lines from primary keys of parent tables to foreign keys in child tables designate table relationships and specify referential integrity. What is more, notation (1 and N) along these relationship lines signify the cardinality of these relationships and clarifies their meaning.

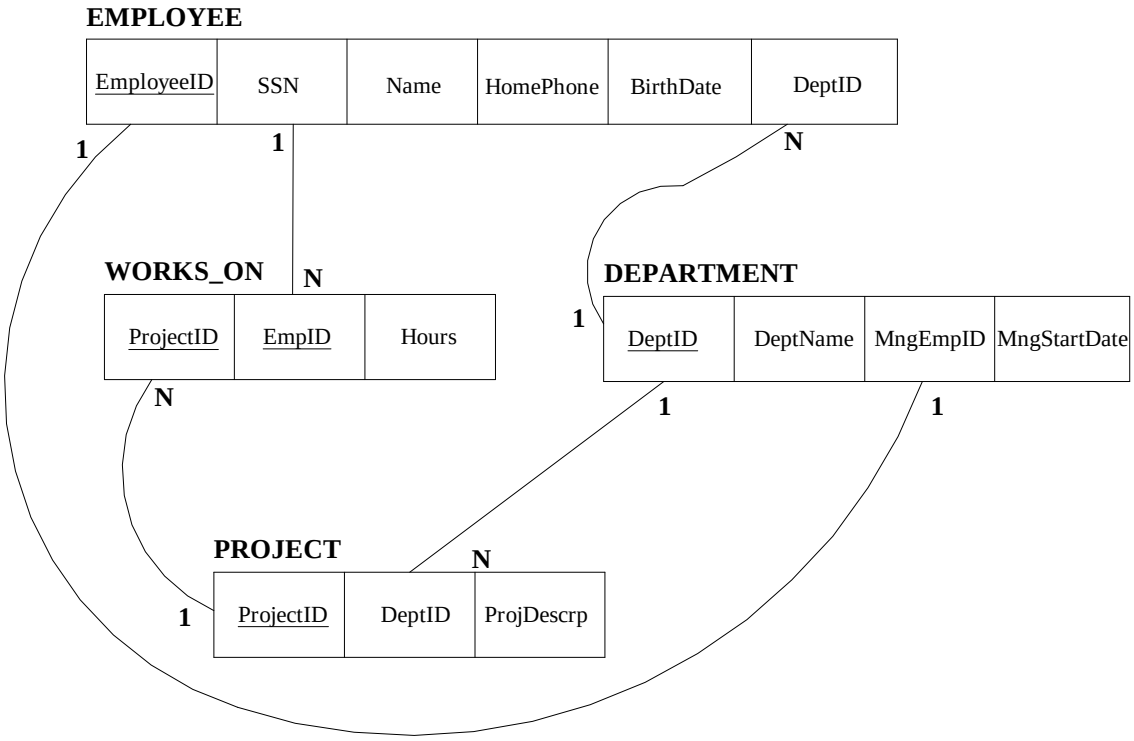


Figure 4.5 Possible relational schema for the COMPANY database used in the text as an illustration. Table relationships and cardinality ratios are exhibited. After Elmasri and Navathe, 2000.

4.4 Normalization

4.4.1 Overview

Designing a relational database is mostly intuitive. This is because we can think of a database as epitomizing an aspect of the real world with table relationships resulting from implicit associations between these entities in real life. Despite this intuition, there are subtleties involved with organizing a database when the semantics of relations and their relationships are not always clearly understood. Consequently, relational database design algorithms exist to aid the developer in normalizing data and therefore, properly structuring the database.

Normalization is a process of analyzing the relation schema of a database to minimize redundancy and eliminate anomalies that can occur from updating a database (Elmasri and Navathe, 2000). Codd (1972) developed tests of relation normalization he called normal forms to specifically address these issues. He developed three normal forms, first normal form (1NF), second normal form (2NF) and third normal form (3NF), with each successive form relying on the fulfillment of the previous. These forms propose increasingly stringent qualifications that relations should meet to be considered normalized (Elmasri and Navathe, 2000). Currently, there are more than three normal forms but the majority of data normalization can be accomplished through these first three (Elmasri and Navathe, 2000). These normalization concepts are summarized here and are applied as a guide to the design of AARDB (Appendix I).

4.4.2 First Normal Form

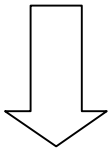
First normal form requires that values in attributes be atomic (i.e., indivisible). This first requirement simplifies the representation of "facts", and therefore their meaning, in table attributes. Figure 4.6 shows an example of normalizing a table according to 1NF.

Table SAMPLE (a), which records information pertaining to collected shells, does not satisfy 1NF because attribute **SampleType** contains data that are not atomic. Record JW2001-150 displays several mollusk genera for **SampleType**, suggesting more than one sample. This would create problems later when said mollusks are sub-sampled (i.e., cutting a shell fragment) and analyzed because **SampleID** and related information would have to be re-entered into table SAMPLE for each mollusk genus analyzed. Solving this problem requires the creation of a new table (COLLECTION) that holds identification values for collections of more than one sample mollusk. Therefore, in table SAMPLE, **SampleID** records only unique values (Figure 4.6).

(a)

SAMPLE

SamplingType	**SampleID**	**SampleType**
Beach	JW2001-150	2 *Mercenaria*, *Spisula*
Core	JW2001-060	*Mulinia*



1NF

(b)

COLLECTION

CollectionID
JW2001-150

SAMPLE

SamplingType	**SampleID**	**SampleType**	**CollectionID**
Beach	JW2001-150-001	*Mercenaria*	JW2001-150
Beach	JW2001-150-002	*Mercenaria*	JW2001-150
Beach	JW2001-150-003	*Spisula*	JW2001-150
Core	JW2001-060	*Mulinia*	

Figure 4.6 Illustration of 1NF. (a) Table SAMPLE does not comply with 1NF because the SampleType field contains multi-value data. (b) SAMPLE complies with 1NF by creating a new table, COLLECTION, and specifying distinct SampleID for SAMPLE such that SampleType is now indivisible.

4.4.3 Second Normal Form

In 2NF, tables should only store data that are entirely described by the primary key. 2NF is based on the concept of *functional dependency* of an attribute in a table with its primary key. Functional dependency is important to understand in relational design and references on database design (e.g., Elmasri and Navathe, 2000) in the References section of this work have thorough discussions on the topic. To simplify the concept, a table is said to satisfy 2NF if the primary key of a table uniquely determine the values of another attribute in that table.

To illustrate this, Figure 4.7 shows a table (SAMPLE LOCATION) with several attributes. The table's primary key is SampleID and the information contained specifies the type of sample and the location where the sample was collected. UDAMS is a geographic identifier given to all new sampling sites.

The problem in this example is that the primary key does not uniquely identify all of the attributes in the table. While a sample will have a specific location, as described by **UDAMS**, **Lat**, **Long** and **LocalityName**, **SampleID** does not functionally determine the location completely. That is, you can have several samples (i.e., **SampleID**) for a specific location. So location information would have to be repeated for each sample from the same sampling site.

In order to minimize redundancy, table SAMPLE LOCATION should be decomposed into two separate tables, SAMPLE and LOCATION (Figure 4.7). Now with two tables, **SampleID** functionally determines the set of attributes in SAMPLE, while **UDAMS** uniquely determines the set of location attributes (Figure 4.7).

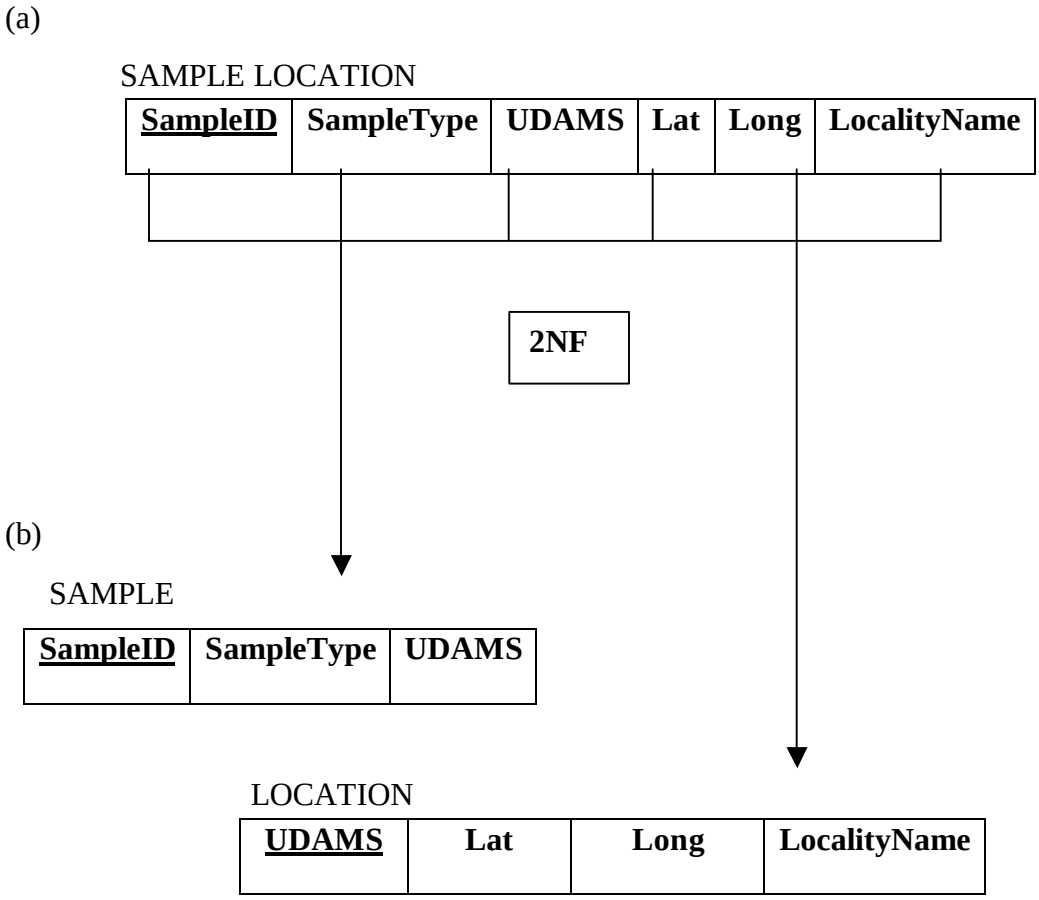


Figure 4.7 Diagram explaining the concept of 2NF. The SAMPLE LOCATION table consists of fields describing location and sample information. To satisfy 2NF the table must be decomposed into two tables, one for sample information (SAMPLE) and one for location information (LOCATION).

4.4.4 Third Normal Form

A table satisfies 3NF when the primary key functionally determines fully all non-primary key attributes (i.e., satisfies 2NF) and non-primary key attributes are not transitively determined by another non-primary key attribute. A table exhibits transitive dependency when the primary key and another non-key attribute uniquely determine an additional attribute. To solve this issue, the table should be decomposed such that the transitively dependent attribute is in a new table along with the attribute that uniquely determines it.

Figure 4.8 exhibits the concept of transitive dependency with the inclusion of a calculated field in table COLLECT SAMPLE. **SmplElev_Top** records the elevation (MSL) of a particular sampling and is determined both by the primary key (**CollectionID**) and the depth from which the sample was collected (**SmpldInt_Top**) within table COLLECT SAMPLE. Furthermore, **SmplElev_Top** is a calculated field that is determined by subtracting **SurfaceElevation** (table LOCATION) from **SmpldInt_Top** (Figure 4.8). We could decompose COLLECT SAMPLE such that **SmplElev_Top,** along with those attributes from COLLECT SAMPLE that fully determine it (**CollectionID** and **SmpldInt**_Top), would exist as a new table. However, that would not solve **SmplElev_Top**'s dependency on **SurfaceLocation**. In this example, to remove the transitive dependency SmplElev_Top should not be used as an attribute in a base table. Instead, this value should be stored in a database *query* or *view* and calculated by the DBMS whenever it is needed.

LOCATION

UDAMS	**Latitude**	**Longitude**	**SurfaceElevation**
07500	35.1235	-75.4254	5.35
07555	36.5355	-76.2356	-16.35

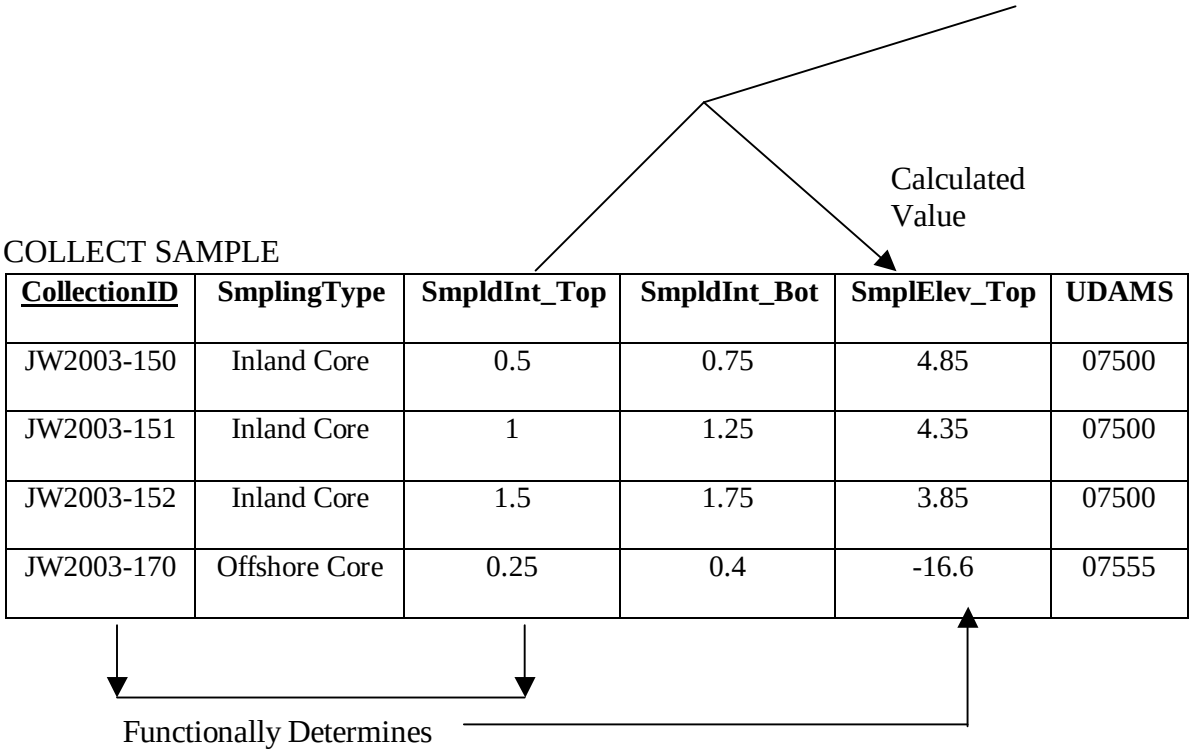


Figure 4.8 Illustrating the concept of transitive dependency. The LOCATION table consists of fields describing location and position of sampling, while the COLLECT SAMPLE table records samplings for a particular site. The attribute **SmplElev_Top** in COLLECT SAMPLE records the elevation of the sample collected. A transitive dependency exists for COLLECT SAMPLE because **SmplElev_Top** is not fully determined by **CollectionID** but also by **SmpldInt_Top**.

4.5 Choosing a DBMS

There are different commercial software packages for different scales of database system needs. These range from the personal computer platform to high-end database management systems for large organizations. Choosing the best system requires a clear understanding of your data management needs, as well as the capabilities of different software. Initial setup and sustaining costs, storage capacity requirements, and time available to users for learning new software are all aspects that would either deter or attract a researcher from particular DBMS software. Furthermore, platform specifications of certain software may also limit one's DBMS choices.

Several aspects were considered in choosing software for UDAL's database needs. First of all, we wanted full relational database functionality because of the advantages discussed earlier. Secondly, it was determined that our database needs could be met by a personal computer implementation of a relational database. This implementation was considered most effective bearing in mind overhead and personnel costs and software training time. Lastly, data sharing options in a Microsoft® Windows® environment were attractive considering that much of the data analysis currently performed runs on a Windows® platform.

While there are more robust commercial DBMS, such as Oracle®, utilization of a high-end database management system is currently deemed unnecessary. However, upgrading from a personal computer platform to a high-end system is a viable future option, as the database is software independent.

CHAPTER 5

PRESENTATION OF THE AAR DATABASE

5.1 Overview of MS Access

Microsoft® Access (2000) is the DBMS used for AARDB. Access is a PC platform implementation of a relational database management system (RDBMS). This chapter presents AARDB as it has been developed in Access for the Windows® environment.

Table 5.1 lists the specifications of MS Access (2000) found in the software's help files. The maximum size of an Access database is inconsequential as Access allows for linked tables. In addition, because Access is ODBC (Open DataBase Connectivity) compliant, several databases (whether MS Access or not) can be linked together expanding the size of a database system. ODBC is a data sharing protocol developed by Microsoft that allows information on a PC to be shared with other database systems (Hoffman, 2003).

AARDB is a single-user database that is currently about 10 megabytes in size and includes records for sites in North and South Carolina. Its size is expected to increase, as samples from all along the Atlantic Coastal Plain are included

Table 5.1 General specifications for a Microsoft® Access database. Because your database can include linked tables in other file, the maximum file size for an Access database file is essentially limited only by available storage capacity.

ATTRIBUTE	MAXIMUM
MS Access database (.mdb) file size	2 gigabytes
Number of objects in a database	32,768
Number of Modules	1,000
Number of characters in an object name	64
Number of characters in a password	14
Number of characters in a user name or group name	20
Number of concurrent users	255

Normally the user first interacts with the "database window" in MS Access when a database file is opened. This window (displayed in Figure 5.1) lists the available objects of the database and allows the user to create tables, queries or forms and compile data into reports for viewing. MS Access has ready to use wizards to aid with creating tables, queries, forms or reports and the beginning user is generally advised to make use of such wizards when creating or changing objects in the database. In addition, through the database window (Figure 5.1), the user can develop macros or Visual Basic for Applications (VBA) procedures to perform functions within the database. Knowledge of VBA is not required for developing a database in Access and using Access wizards makes development of a database simpler.

Because MS Access is a relational database, it makes use of SQL (Structured Query Language) for its data definition language (DDL) and data manipulation language (DML), although it is a particular dialect of SQL called Access SQL. An example of SQL is found in chapter 5.3.2 and in Figure 6.3.

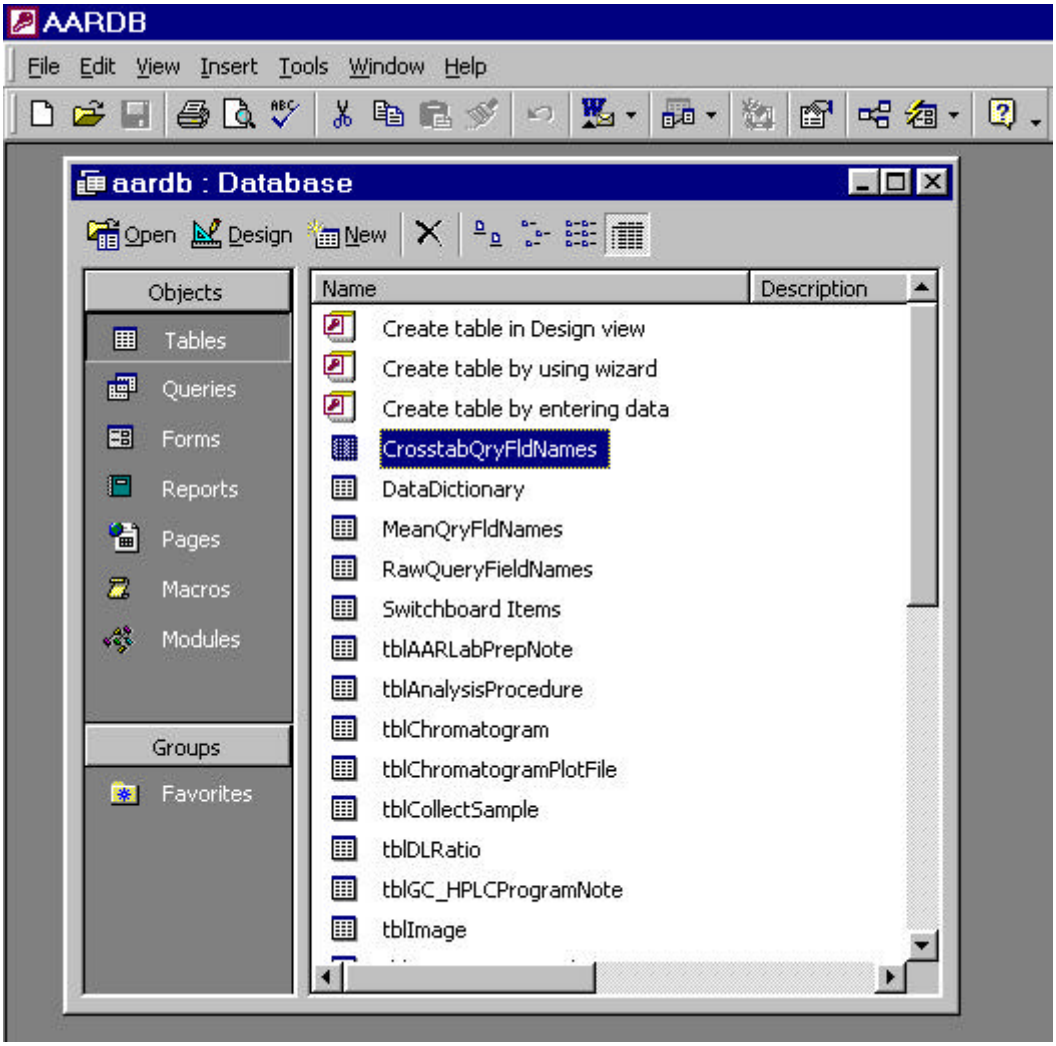


Figure 5.1 View of the Database Window in MS Access. Database objects appear in the left portion of window. Objects stored in the database file are displayed in the right portion of the Database Window.

5.2 Database Tables

Several tables contain the data that make up AARDB. Below is a comprehensive list of the base tables and brief descriptions of their purpose. When browsing the database (using Appendix II) it may be useful to refer to this list to understand the reasoning behind the database structure.

tblLocation

This table contains sampling site information. Information includes geographic identifier (UDAMS), an informal name of site and positional information, which includes latitude and longitude (decimal degrees) as well as elevation (from MSL). Horizontal datum and vertical datum are also specified.

tblSamplingLog

This table describes important positional information that is used to calculate elevation of samples and maximum depths of cores. This includes start of sampling position (i.e., top or bottom of section), length of measured section, thickness of overburden not considered part of a measured section and an ocean tide factor. Records in this table are identified by a location's informal name along with any section name.

tblSampleCollection

Specifies sample collection details including the sampling type (e.g., surface, inland cores, offshore cores, excavation/exposure, etc.) collection ID and the sampling interval in meters. This table is represents the act of sampling as well as the collection object itself. Therefore, both sample bags with multiple shells and single shell collections are represented in this table.

tblSample

This table holds information describing particular samples. This includes sample ID, which is usually an appended version of collection ID, and sample type (normally sample genus). All field notes about an individual sample are recorded in this table.

tblSubSample

Sub-samples that are created from a sample (or a sub-sample) are recorded in this table and described by a unique identifier. The creation date, position where sub-sample was collected, and any related literature references are stored in this table.

tblSubSamplingProcedure

This table describes the purpose of creating a sub-sample, whether for chromatographic analysis or another analysis. Fields include a procedure identifier, brief description of procedure, as well as the fraction to be analyzed (e.g., total or free amino acid content).

tblAnalysisProcedure

Describes the method of analysis used to obtain results from sub-samples. Includes method identifier, laboratory performing the procedure, accuracy limits of analysis method as well as the units measured for the particular analysis.

tblLaboratory

Contains reference information of the laboratory that created a sub-sample and/or performed an analysis. Information includes lab name, address and contact person.

tblChromatogram

This table contains information describing chromatograms that result from chromatographic analysis of sub-samples from UDAL. Chromatograms are described by the date of analysis, the method used for analysis (e.g., Gas Chromatography or High Pressure Liquid Chromatography) and the analysis device's internal run number.

tblDLRatio

This table stores the calculated D/L ratios for amino acids, as interpreted from a chromatogram. Distinction is made for ratios calculated from chromatogram peak areas or heights. Raw amino acid abundance is currently not stored in this database.

tblOtherResult

This table contains finalized results of various types of analyses other than those that produce chromatograms from UDAL. This includes radiometric and isotopic data performed on samples collected by UDAL. Although this table stores somewhat of a hodgepodge of data, AARDB's conceptual design allows for later partitioning of these data as separate entities (see Appendix I). Currently, chromatographic data for which no chromatograms were recorded in the database (e.g., data from students theses or dissertations) are also included in this table.

tblTaphonomicCharacter

Within this table taphonomic characteristics of samples are recorded for several taphonomic parameters. Currently taphonomic parameters are characteristics described in Kowalewski et al. (1995).

tblImage

All sample or sample collection (i.e., single or group) image names are stored within the database. Only the image names are stored to allow for the DBMS to retrieve images even if the image path varies. Images are stored separately in a folder within the same directory as the database file.

tblInterpretativeResult

This table includes any results of analysis that are not measured values but interpretations of other results (e.g., aminozone designations). The information contained in this table may duplicate information that can be retrieved from any related literature reference such as from an appendix of a student's thesis or dissertation.

tblLookupTable

This table contains parameters used by several tables to restrict the data input choices of the user. The table consists of three attributes; Choice, which holds parameter values, Category, which identifies the type of parameter, and an attribute for a brief description of the parameter. These choices appear as a drop-down menu on forms or the table themselves. All parameter choices are stored in this table in order to maintain referential integrity of some non-key attributes and to simplify database programming.

tblReference

This table stores any literature reference where sub-sample analysis results were published. Several references can be stored for an individual sample or many samples can correspond to a particular reference.

TblProject

Currently, no information is stored in this table but is available for future use. A project name and contact information are recorded here. Information in this table would be related to site location information, where several site locations may exist for one project.

DataDictionary

This is where the database metadata is stored. All attribute names along with their table name, data type and description are recorded in this table. This table is useful as documentation of the database. Currently, this table must be updated manually.

tblSampleCollectionInfo and tblSampleCollectionTapho

These two tables contain information on shell samples of a collection from beach transect surveys. Taphonomic characteristic data on these shells are maintained in this way because these data are currently not duplicated by tblTaphonomicCharacter.

5.3 Database Structure

5.3.1 Core Tables

The relational schema for this database is presented in Figure 5.3. To view the database schema in Access, one can choose the Relationships button on the Database tool bar (Figure 5.2) or go to Tools/Relationships in the Menu bar (Figure 5.2) (see Appendix II). The schema represents the conceptual organization of the database tables. If the tables are properly normalized (refer to Principles of Database Development chapter) than practically any data selection scenario can be achieved through the table relationships.

Several tables make up AARDB and numerous relationships are established between tables. Several branching tables are not shown in Figure 5.3 as only the core tables and their relationships are exhibited. The reader is referred to Appendix IV to view the complete table relationship structure of AARDB.

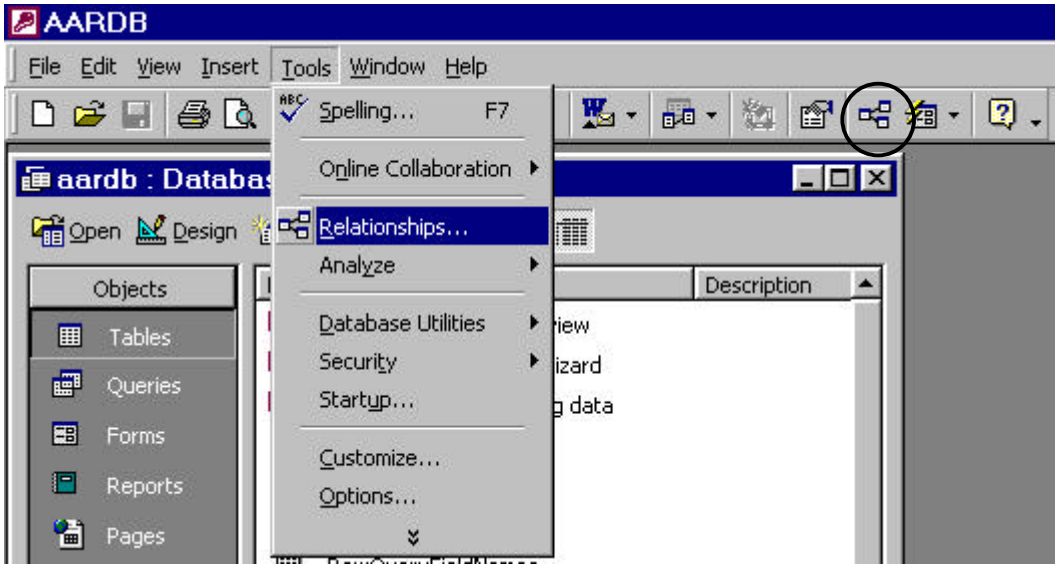


Figure 5.2 Screen capture displaying how to view table relationships, either by using the menu option Tools/Relationships… or by choosing the table relationship symbol on the database toolbar (circled in the figure).

Figure 5.3 exhibits the referential integrity constraints of the core database tables. The database schema is set up such that parent to child relationships flow from left to right. Each parent table exhibits a 1:N relationship with its child and one can follow the schema order from a sampling site location to sub-sample results (Figure 5.3). At tblSubSample the schema forks, where the child entities of tblSubSample are both tblChromatogram and tblOtherResult (Figure 5.3). From tblChromatogram, AAR D/L ratios are stored in table tblDLRatio.

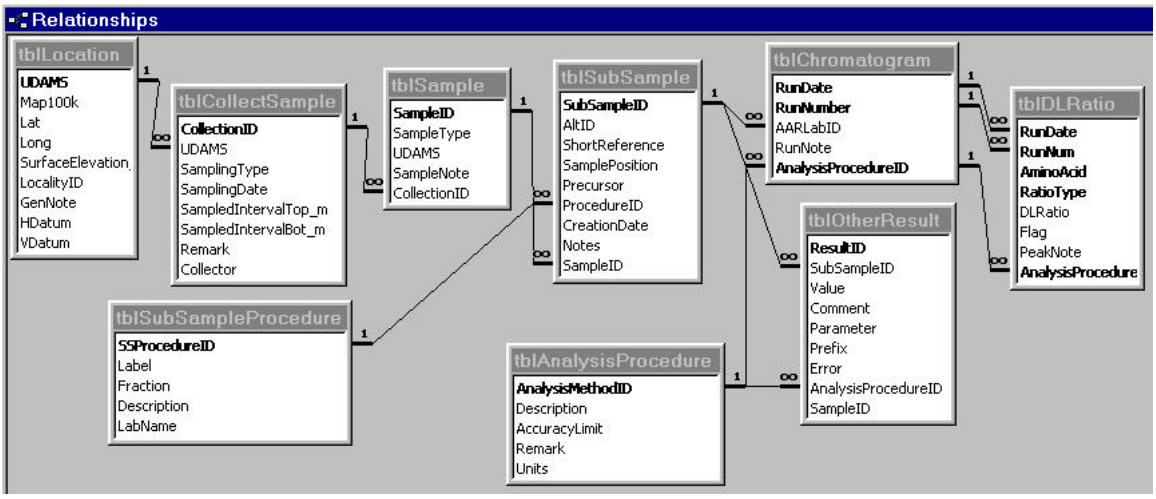


Figure 5.3 Table relationships window in MS Access. Only core tables of AARDB are shown with referential integrity designated by joining lines. **1** and the infinity symbol designate cardinality ratio for 1:N relationships.

Two methods tables that reference procedures performed on sub-samples are also a part of the database schema. They are tblSubSamplingProcedure and tblAnalysisProcedure and the distinction between them is evident from their participation in table relationships (Figure 5.3). One table contains information regarding the sub-sampling process (e.g., the laboratory responsible), while the other table (tblAnalysisProcedure) stores information pertaining to the actual analysis method used.

The logic behind the database design is easily understood from the relationship schema. At a particular sampling site, the spatial coordinates are recorded (i.e., latitude, longitude and elevation) and designated a unique geographic identifier (UDAMS). Next, sampling information for that site is recorded including the name of the collector, the type of sampling (i.e., whether a core, surface sample, etc.), and the sampled interval of the collected sample. Then specifics of individual samples including the genus of the sample (SampleType) are recorded for each collection made. Subsequently in the lab, as

samples are cut and sub-samples created, unique identifiers are given for each sub-sample and pertinent sub-sampling information is also recorded (e.g., position where sub-sample was taken). As sub-samples are analyzed, whether by UDAL or another laboratory, these data are recorded as AAR D/L ratios or other results such as radiometric ages. These results are relatable to specific sub-samples or to a particular mollusk sample, which allows results of several analyses to be compiled for sampling sites.

5.3.2 Minor Tables

Besides the core tables presented above, the database also contains minor tables, that is, tables that are called less frequently for standard database queries executed in AARDB. Nonetheless, these minor tables are still base relations and contain normalized raw data.

Figure 5.4 shows AARDB table relationships with tblSamplingLog and its N:1 relationship with tblLocation. Information stored in tblSamplingLog is used to calibrate the elevation of collected samples for a particular site. Each site location can have a surface elevation stored in the database (tblLocation) but for samples collected at depth calculations need to be made to determine a sample's elevation.

For a collection event, the sampling interval in meters is recorded in tblCollectSample. However, a sampled interval is a relative measure, generally from the top of a measured section, so this value must be subtracted from the site's surface elevation to calculate sample elevation. Sometimes further information is needed to obtain a sample elevation such as a tide correction (tblSamplingLog) for samples collected offshore.

A data select query can contain a simple algorithm to subtract the necessary factors in tblCollectSample and tblSamplingLog from a site's surface elevation to calculate the elevation of particular samples. For example, for samples collected at depth, the SQL algorithm to retrieve sample elevation would look like the following:

SELECT [tblLocation.SurfaceElevation_m] –

[tblSamplingLog.OverBurdenThickness_m] – [tblCollectSample.SampledIntervalTop_m]

AS SmplElev FROM . . .

In the SQL selection above, each factor found in tblSamplingLog is subtracted from the surface elevation recorded in tblLocation. SmplElev is the name assigned to the new attribute calculated from the query. Alternatively, if a record does not exist in tblSamplingLog, a particular sample's elevation would simply be calculated by subtracting the sampled interval from the surface elevation.

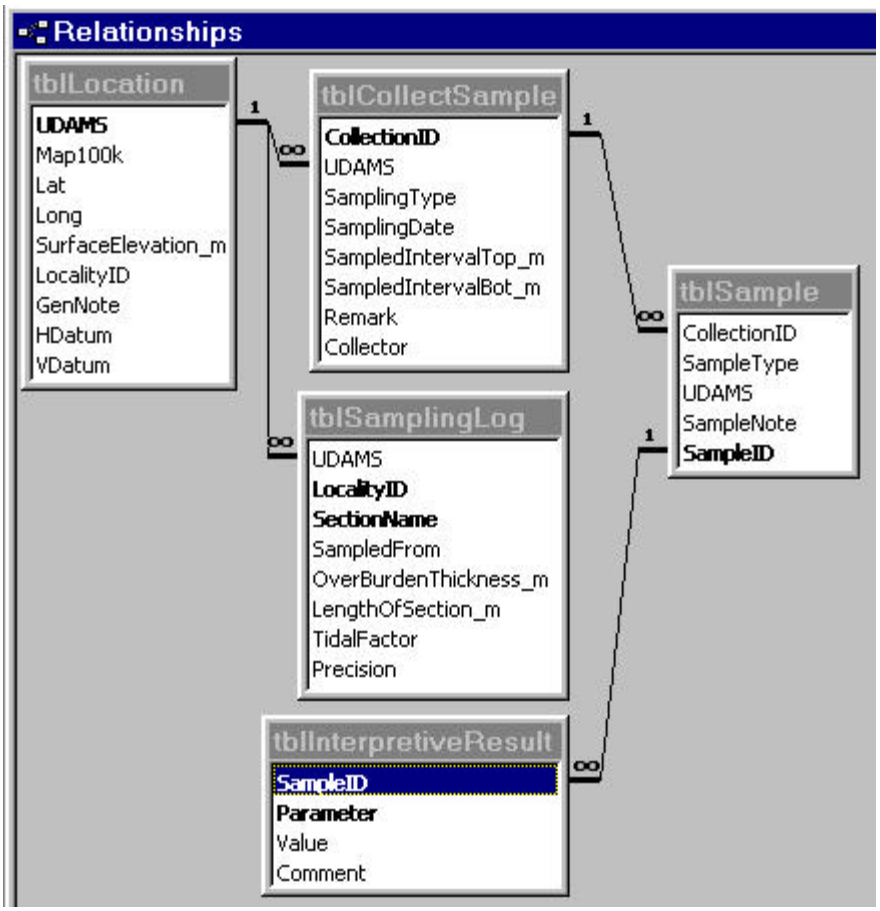


Figure 5.4 Minor tables and their relationship with tblLocation and tblSample. Important spatial information is recorded in tblSamplingLog, while tblInterpretiveResult contains interpretive information on samples such as aminozone designation or geologic unit.

Another aspect of AARDB is the collection of physical characteristic data for each archived mollusk sample. At present, over 1,200 individual *Mercenaria* and numerous *Mercenaria* collections have been characterized with taphonomic attributes such as abrasion, fragmentation and color. This information, especially when coupled with geochronologic data such as AAR ratios or radiometric ages, could be useful for taphonomic studies of macrofossils along the Atlantic Coastal Plain.

The database includes tblTaphonomicCharacter that records taphonomic characteristics for individual samples (Figure 5.5). Also included are taphonomic characteristics for entire shell collections amassed from beach transects, tblSampleCollectionInfo and tblSampleCollectionTapho. The latter two tables record collection totals for a particular taphonomic characteristic. Of course, totals for specific characteristics can be calculated from tblTaphonomicCharacter (see section 6.2.4) but tblSampleCollectionTapho represents a large dataset that is currently not duplicated in tblTaphonomicCharacter and so is maintained in the database.

Images of many individual shells and shell collections are also stored in the database for viewing. Table tblImages records the image file name for a collection or individual sample (Figure 5.5). These images can be called from the database or viewed through user forms within MS Access.

A number of table relationships exist at the sub-sample level because several aspects of the database are expressed at the sub-sample level (e.g., sub-sampling procedures and analysis results). Figure 5.6 shows how these relationships are structured within the database.

Often sub-samples can be generated from other sub-samples. To account for this a recursive relationship needs to be established within tblSubSample such that the precursor and its progeny are recorded and relatable. In Figure 5.6 this table is shown as a copy of tblSubSample (tblSubSample_1) with a 1:N relationship with tblSubSample. That is, for every sub-sample in tblSubSample, unlimited sub-samples can be created and stored. This is often the case when a sub-sample is analyzed using several methods.

Keeping track of numerous analyses and sub-analyses is made simple by relating a sub-sample with its precursor (Suckow and Ingolf, 2001) (Figure 5.6).

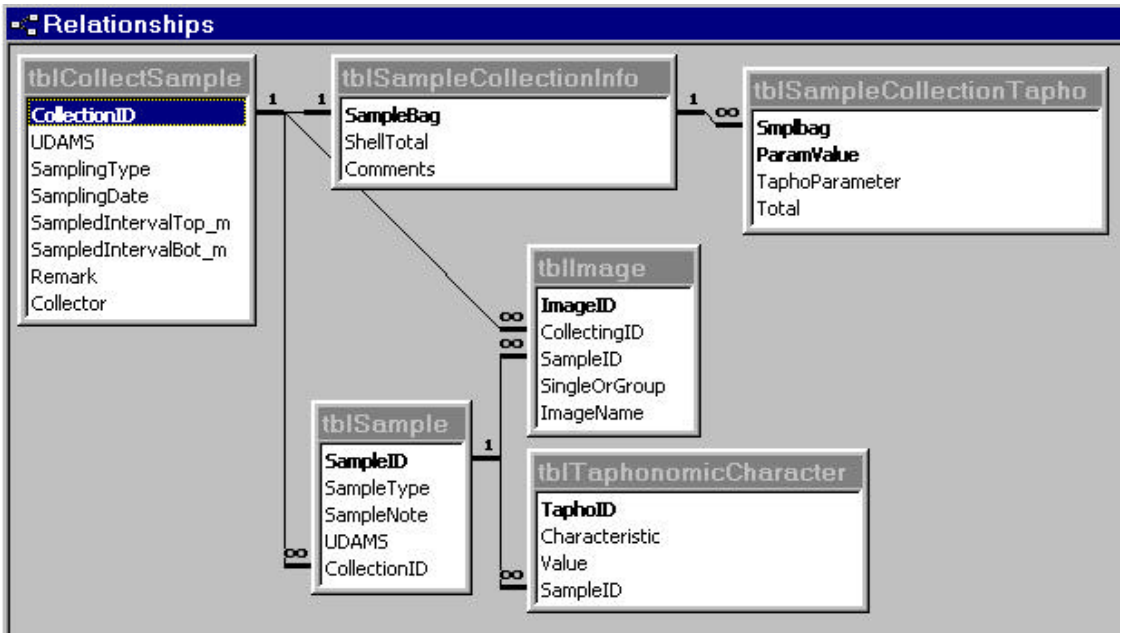


Figure 5.5 Table tblCollection and tblSample 1:N relationships with tblImage and tblTaphonomicCharacter. Digital images exist for both sample collections and individual samples.

5.4 AARDB User Interface

5.4.1 Entering the AARDB

On entering the database, the user is confronted with a switchboard and two options, Data Select or Data Input (Figure 5.7). This form helps to simplify usage of AARDB by constraining the user's main tasks, either querying the database or performing data entry. More experienced MS Access users can still browse through database objects by opening Access's Database Window (Figure 5.7).

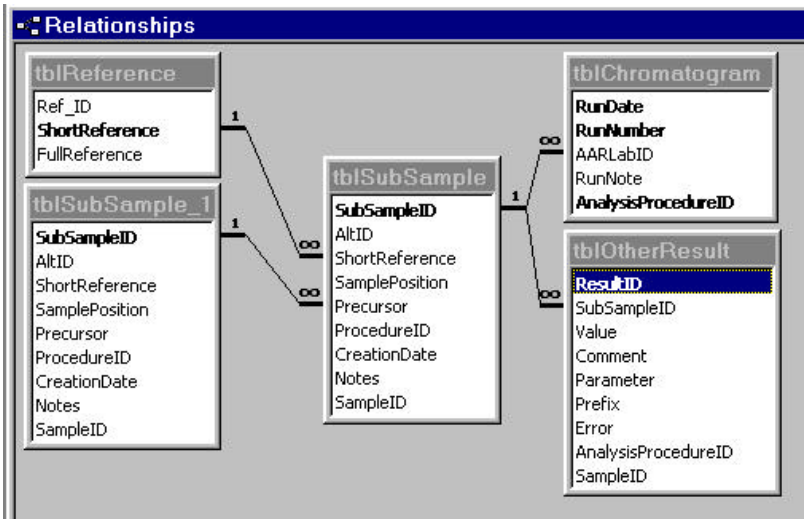


Figure 5.6 Table tblSubSample participation in other table relationships. The 1:N relationship established between tblSubSample_1 and tblSubSample represents a recursive relationship within tblSubSample.

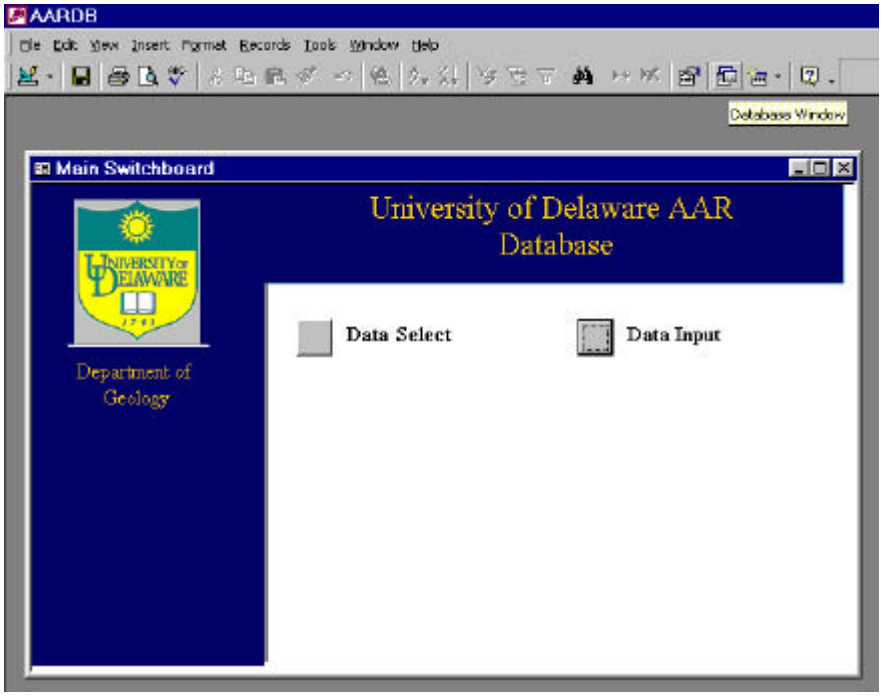


Figure 5.7 AARDB Main Switchboard. Two options are given to the user, Data Select or Data Input. Experienced MS Access users can navigate through the main Database Window as well as shown in the figure.

5.4.2 Using Data Input Forms

Choosing the Data Input option on the main switchboard will open up an input forms switchboard to the right of the screen (Figure 5.8). This toolbar makes use of toggle buttons to open or close data entry forms. The majority of data entry operations are possible by the options presented on this toolbar (Figure 5.8).

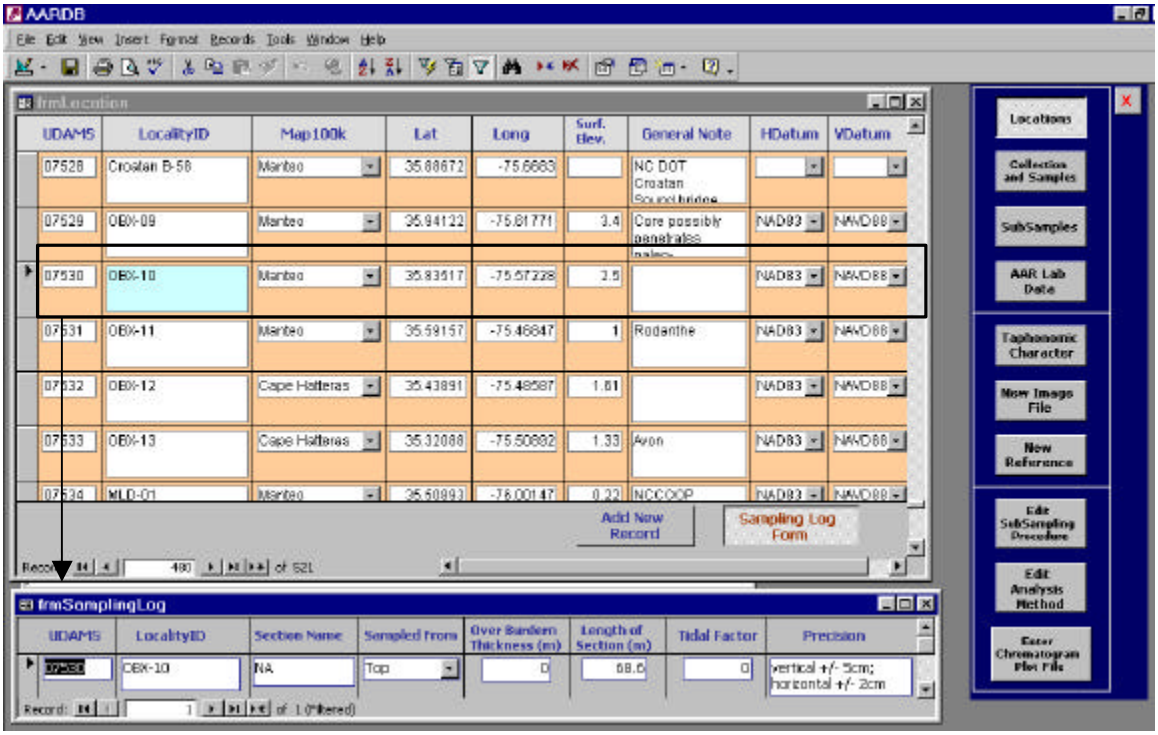


Figure 5.8 Data entry forms. Switchboard on right allows the user to choose the data input form. The Locations button is depressed and brings up the site location form that is linked to tblLocation. Also available through this form is the option to edit data in tblSamplingLog.

Figure 5.8 shows a screen capture of some data input forms opened for data entry or browsing. The input forms switchboard to the right of the screen shows that the Locations button is depressed, opening the Location form. This form is linked to tblLocation and is used to browse, edit or update data in this table. Also shown in Figure

5.8, the Location form is coupled to a form that is linked to tblSamplingLog. So that for a selected record in Location, the corresponding information in tblSamplingLog can be edited or displayed (Figure 5.8).

Data input forms are designed to aid the user entering data into the database. Because there are many tables existing in a relational database, it may not be apparent to the average user where information should be recorded, even though a relational design is said to be intuitive. Therefore, it is suggested to use the data input forms for entering data instead of directly through the database tables. Furthermore, data entry restrictions that cannot be enforced by database integrity constraints are possible by coding these restrictions in a form.

5.4.3 Querying the Database

Database queries retrieve data as specified by the user. They also represent database views, which are representations of the database but cannot be updated as if one were browsing the database tables. Queries can be formulated to bring up very specific information and so can be quite complex, including various relationships between numerous tables.

To query AARDB, the standard query design interface in Access is useful for most queries (Figure 5.9). Access also has query-building wizards to aid the less experienced user. For users more comfortable with SQL, MS Access has an SQL View to formulate queries (Figure 5.9).

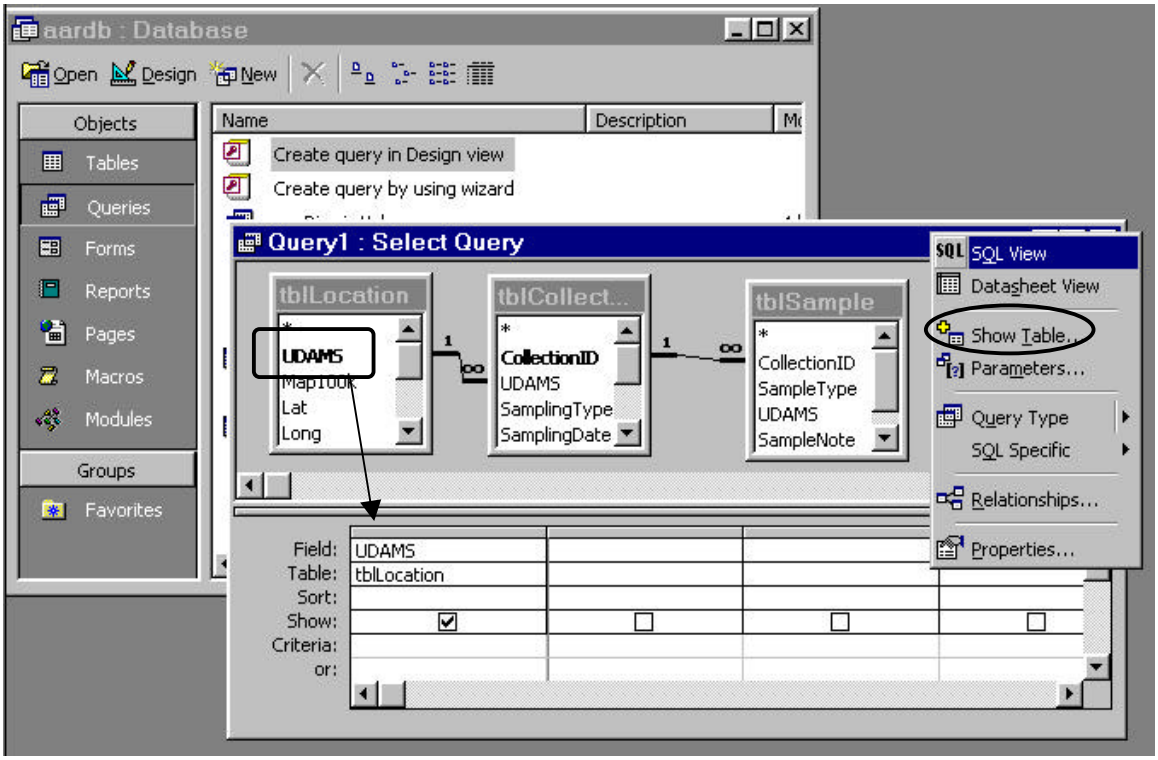


Figure 5.9 Designing a query in MS Access. Creating a query in design view entails adding tables of interest (circled in black) to the design window and then dragging an attribute of interest to the Field editor. Creating a query using SQL is possible by a right mouse click on the design window, which brings up an options menu.

Besides the normal database tools provided in Access, several user-friendly forms have been created to aid the user in retrieving data from AARDB. On choosing Data Select from the main switchboard (Figure 5.6) new options are presented to the user (Figure 5.10). These options open up forms to help the user browse or query the database. For example, the Main Parameter Query option opens up a form with multiple controls to specify query parameters (Figure 5.11). The user can fill in these controls with the desired constraints or leave them blank, signifying no constraints. Controls are ordered similar to how the data exists in the database (Figure 5.11).

The Main Parameter Query is based on a stored query in AARDB that calculates mean D/L ratios for amino acids (stored as vtblAARMeanRatios_Crosstab). Specifying desired parameters off of a stored query cuts down on the query execution time and simplifies the background Visual Basic coding necessary to execute the query. However, the user can still choose to add options to their query such as calculating sample elevation or retrieving radiometric and taphonomic data. These extra options combine the stored mean D/L ratios query with the base relations, tblSamplingLog, tblOtherResult and tblTaphonomicCharacter, respectively. Another option, including standard deviations of D/L ratios, builds off of a different stored query, vtblAARMeanRatios.

Similar to Mean Parameter Query, Raw D/L Ratio Query (Figure 5.10) brings up a form that can retrieve raw D/L ratios and laboratory notes from the database. Controls on this form are used to constrain or ignore parameters for the query.

All formulated queries can be saved in the database by means of the File menu/ Save As… option. Stored queries or database tables can either be opened from the List Queries and Tables option (Figure 5.12) on the Data Select switchboard (Figure 5.10) or through the List Saved Queries button on some forms (e.g., Figure 5.11). Because the SQL statement is only saved, the database view opened from a query always displays the most current data. Therefore, saved queries would represent the most commonly executed inquiries of the database.

5.4.4 Data Analysis

Several data analysis options are provided in MS Access. Access allows you to create reports with graphs and charts for queried data, although, the functionality of these features is limited. Another option would be to export data to other data analysis

software. Access allows the user to export a query to other Microsoft® Office products through the Tools menu/ Office Links option or data can be shared between software applications through an ODBC connection.

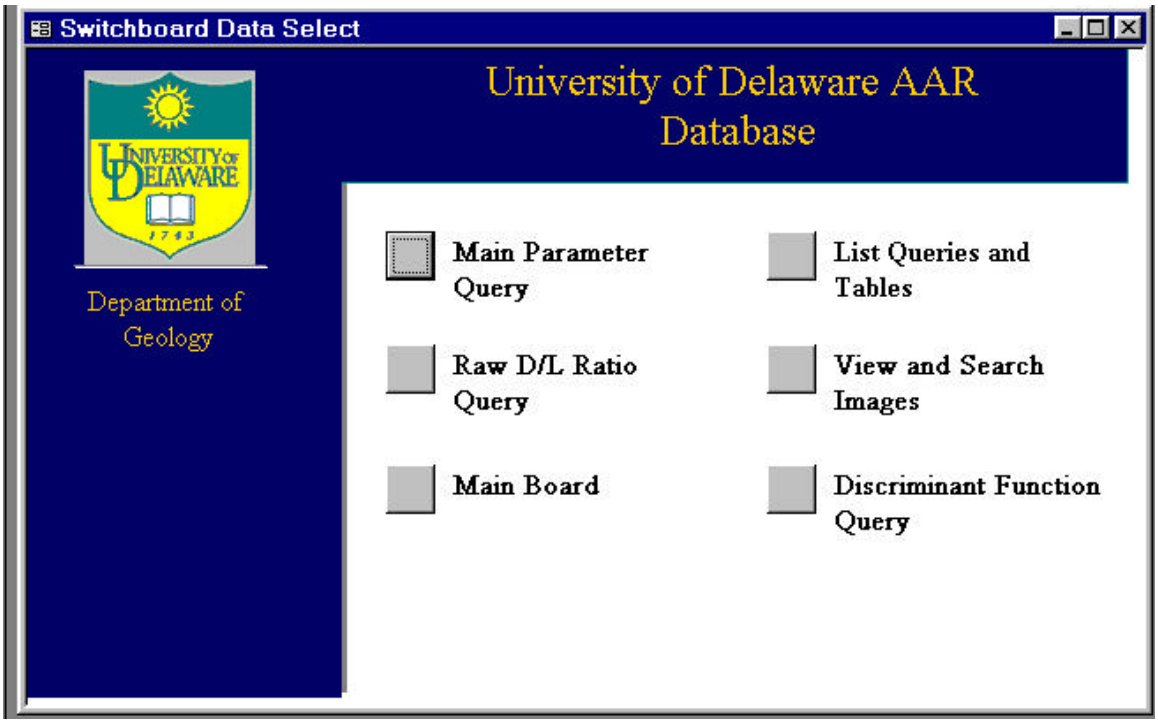


Figure 5.10 This figures shows the Data Select switchboard and the options available to the user. Currently, two user-friendly forms are available to develop a database query (Main Parameter Query and Raw D/L Ratio Query) and other forms for browsing AARDB. Discriminant Function Query is a form set up to query AAR records specifically for analysis in a preset Microsoft® Excel worksheet as an example of Automation.

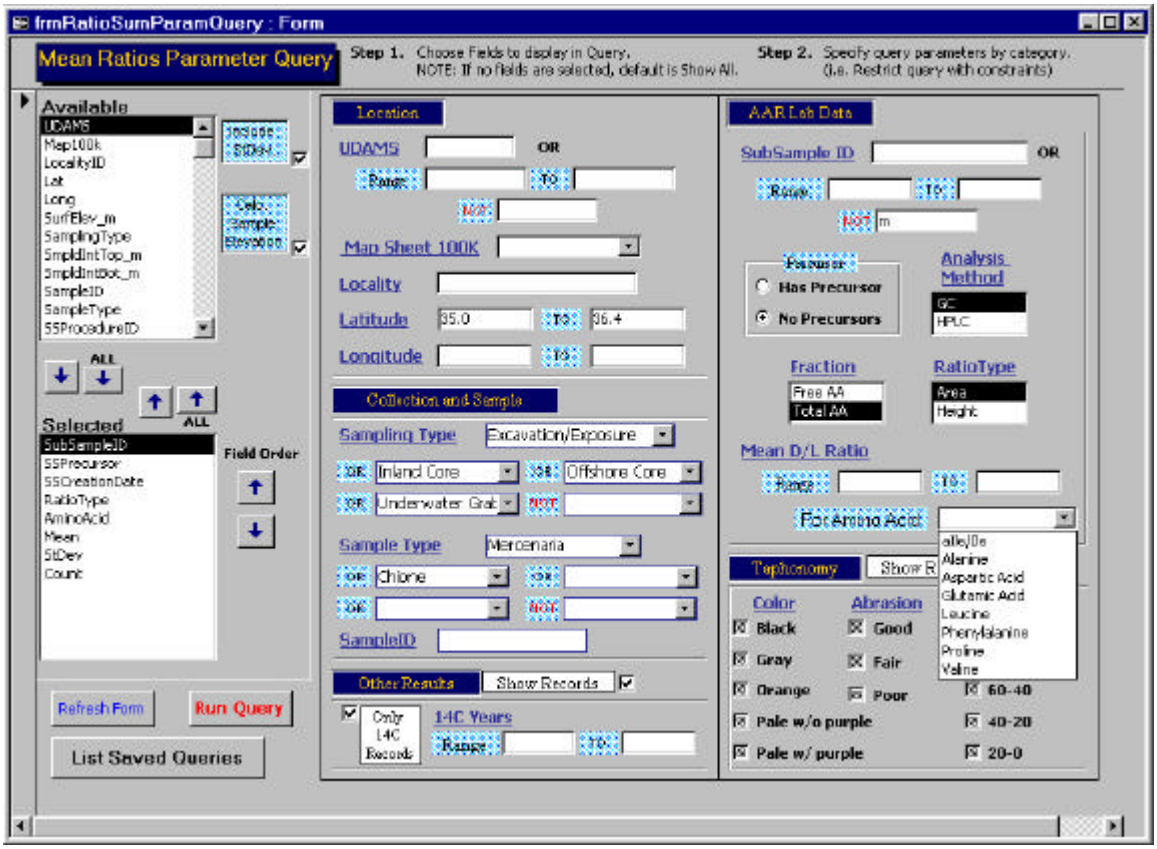


Figure 5.11 Main parameter query giving the user the option to query the database by filling in the form with the desired constraints. The figure displays the option to retrieve general D/L statistics (mean, standard deviation and number of analysis for a sub-sample). Most data querying tasks in AARDB can be accomplished through this form. However, more complex querying may require building a query in MS Access' built-in query builder.

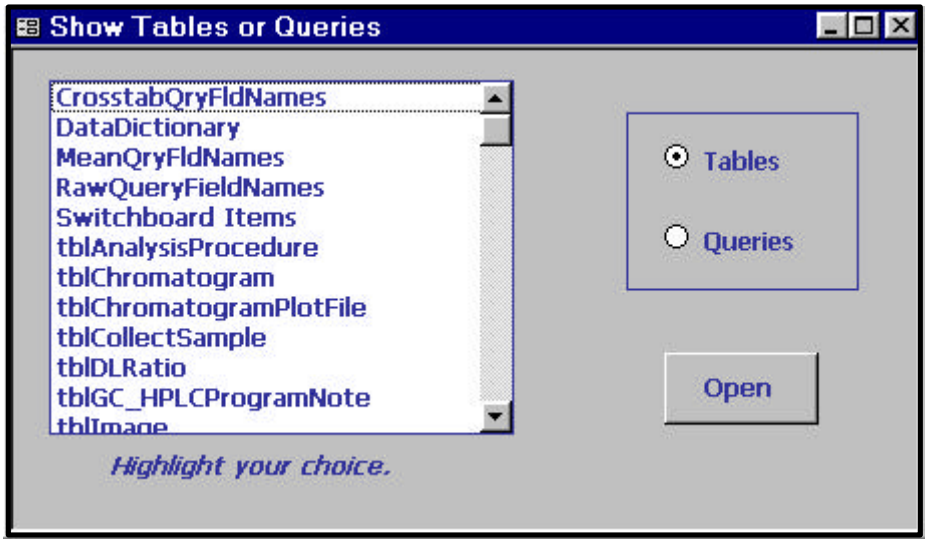


Figure 5.12 List Queries and Tables option on the Data Select switchboard brings up a selection form that shows query and table objects in AARDB.

Data analysis options presented in this thesis are reserved for the next chapter. In this section data export and analysis are performed in Microsoft® Excel and ESRI's ArcGIS® through automation. Automation allows MS Access to use objects in other software (or vice versa) via a common language, VBA, to perform data analysis tasks. The Discriminant Function Query option in the Data Select switchboard (Figure 5.10) is an example of automating data export from Access and analyzing the data in Microsoft® Excel.

CHAPTER 6

CASE STUDY: STATISTICAL AND SPATIAL TREATMENT OF DATA

6.1 Introduction

This section applies the functionality of AARDB to studying the aminostratigraphy of the northeastern portion of coastal North Carolina (Figure 1.2). Work is currently underway by the Coastal Carolina Project to characterize the framework geology of the Outer Banks barrier island-estuarine system. A combination of geophysical, geochemical litho- and bio-stratigraphic tools has been utilized in order to develop a regional Quaternary sea level/climate history (Thieler et al., 2003). We can take advantage of the abundant AAR analyses from this region by using advanced visualization techniques for examining aminostratigraphic results.

This chapter can be read while referencing Appendix IV, the CD included with this document. The CD includes a "readme.htm" file that details software and memory requirements as well as installation instructions. Other files included in Appendix IV are the database file (aardb.mdb), as well as other spreadsheet and GIS files used for the demonstrations described here. Examples of data retrieval and manipulation are presented using available data sharing methods such as ODBC and automation through VBA.

6.2 Gathering the Data

6.2.1 Overview

Most of the pertinent data can be retrieved from the database by using the Main Parameter Query from the Data Select switchboard (Figure 5.11). The constraints of our query are based on our geographic region of interest, the type of samples (i.e., genus) and the types of analysis results we are interested in. These points are summarized below as the input in the Main Parameter Query form.

- ?? All sites in the database within northeastern NC
- ?? All subsurface and surface sampling types.
- ?? All *Mercenaria* sample types.
- ?? All AAR mean D/L ratios, radiocarbon and taphonomic results.
- ?? Only GC analysis method for total amino acid fraction.
- ?? Exclude results that represent sub-samples of a sub-sample.

6.2.2 Geographic and Sample Parameters

A latitudinal range of approximately 35.0° N to 36.6° N will constrain the sampling sites for the region of interest. Then, choosing Inland Core, Offshore Core, Surface and Excavation/Exposure from the Sampling Type drop-down selection boxes satisfies the type of samplings constraint. We can also specify the genus *Mercenaria* from the Sample Type drop-down selection box for our query.

Besides AAR analyses, we want to include radiometric results for the region. Currently only radiocarbon analyses are recorded in the database and checking on the Include Other Results option (Figure 5.11) allows our query to also draw from this dataset. We do not want to restrict our query further by checking the "Only 14C Records," as this would only retrieve AAR records that have corresponding radiocarbon analyses.

6.2.3 Sub-sample and Method Parameters

Sub-samples recorded in AARDB are either created from samples or from other sub-samples. To date, the only sub-samples created from other sub-samples recorded in the database are from ILC standards (which we are not considering for our query) and for repeat GC analysis with methanol treated derivatives. The methanol treated sub-samples that come up in our query should be removed because they represent repeat values for the same AAR derivative and are not significantly different from their non-methanol treated precursors. A SubSampleID appended with an "m" identifies the methanol treated derivatives. These unwanted sub-samples could either be removed from our query by specifying the No Precursors option for sub-sample data (Figure 5.11) or by typing an "m" in the Not text box for SubSampleID selection (Figure 5.11).

Further constraints for AAR data involve specifying the analysis method used for deriving D/L amino acid ratios. GC analysis method, Total AA (amino acid) fraction, and Area ratio type should be selected using the list boxes within the AAR Lab Data section of Main Parameter Query (Figure 5.11).

Clicking on the Run Query button in the lower portion of the Main Parameter Query form (Figure 5.11) executes our query, thereby retrieving results based on our specifications. The results of our query are automatically saved in the database under the name "ResultQuery;" however, this query is overwritten every time a new query is executed from the Main Parameter Query form. The specifications of this query have also been saved as NENCResults in the stored queries of Appendix IV.

Accessing this query from outside MS Access is possible through an established ODBC connection. The steps necessary to set up an ODBC connection are described in

Appendix II, "Setting up an ODBC Connection to Your Data." With this connection we can later retrieve data from GIS software and display the records in our GIS project as current data.

6.2.4 Taphonomic Characteristic Data

Inclusion of taphonomic data could also be useful in our study. The combination of chronological and taphonomic data can bring insight to the distribution of fossil shell material on coastal plain beaches (Wehmiller et al., 2003). In particular, for *Mercenaria* shells, dark color is a good indicator of shell age (i.e., whether Holocene or Pleistocene) (Wehmiller et al., 1995; Bart, 2001). However, for our purposes it will not be too useful to gather taphonomic data for individual beach samples (currently taphonomic data are only recorded for *Mercenaria* samples collected from beaches). Instead we need to draw these data by building a new query in MS Access that will give us tallies of taphonomic attributes for beach transect collections.

To do this, we will need to select taphonomic characteristics for sample collections (rather than samples) from tblTaphonomicCharacter. In addition, some pertinent records are stored in tblSampleCollectionTapho as totals of taphonomic parameters for beach transect collections. We want to convert the records in tblTaphonomicCharacter into tallies of taphonomic parameters to match the data in tblSampleCollectionTapho. Then the records from these two tables can be combined into a single database view. Building this special query is a multi-step process and is illustrated here by using MS Access query design options and SQL code.

First we must build two queries from the Design view option (Database Window-Queries-Create query in Design view) (see Figure 5.1 for the Database Window). The

first query should be designed using table tblTaphonomicCharacter as the source data (see Chapter 5.4.3 for instructions on building a query in design view). Add the CollectionID, Characteristic and Value data fields.

Next, the records are to be converted to totals for each value (i.e., Black, Gray, Orange, etc.) of a taphonomic parameter (i.e., Color, Abrasion, Percent Shell). To do this, we need to specify a GROUP BY clause for the query. Instead of writing this out in SQL, a quick Design view trick is right-click the computer mouse with the cursor over the table window of the query Design view (Figure 6.1). Change the query type to Crosstab Query, then bring up the same menu and change the query type to a Select Query again (Figure 6.1). This causes the Total option in the Field editor window to appear (Figure 6.2). With this option we can specify that the records be grouped by CollectionID, then by Characteristic and then by Value. This causes each value of a taphonomic characteristic for a sample collection to represent a unique record in the view. Adding the Value field to the field editor a second time and specifying a COUNT function for that field generates the totals for each value (Figure 6.5). This query should then be saved (File/Save As…) and given a name.

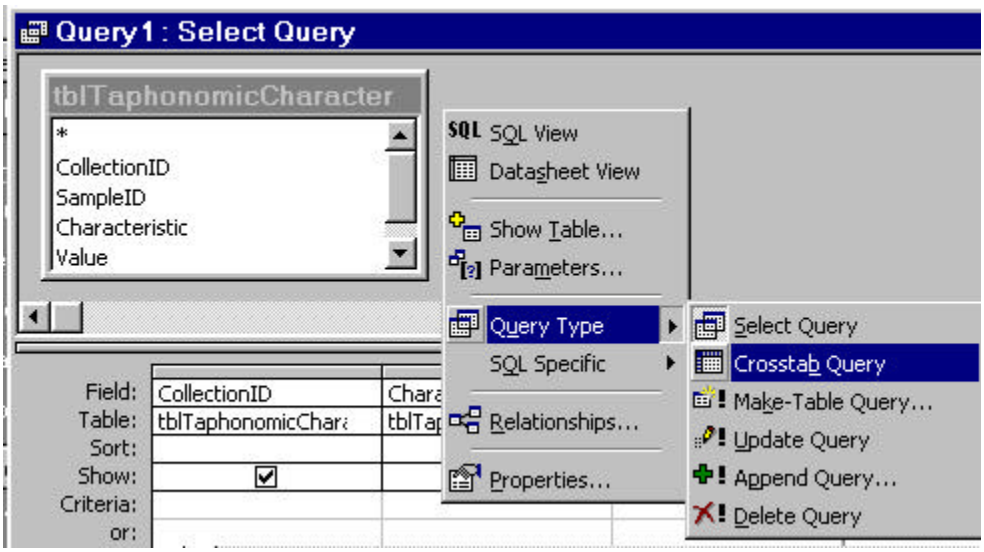


Figure 6.1 Screen capture showing the right mouse click menu of the Design query window.

The second step involves combining the query made in the first step with data from tblSampleCollectionTapho. Records in tblSampleCollectionTapho are already in the taphonomic characteristic, value and totals format similar to our first query. The data in our first query and tblSampleCollectionTapho must contain the same attribute types and have them in the same order to perform a combination or UNION query. However, a UNION query must be done in the SQL View, accessed through the new query Design view. The SQL code to perform the necessary UNION query is shown in Figure 6.3. This new query is then saved and given a name.

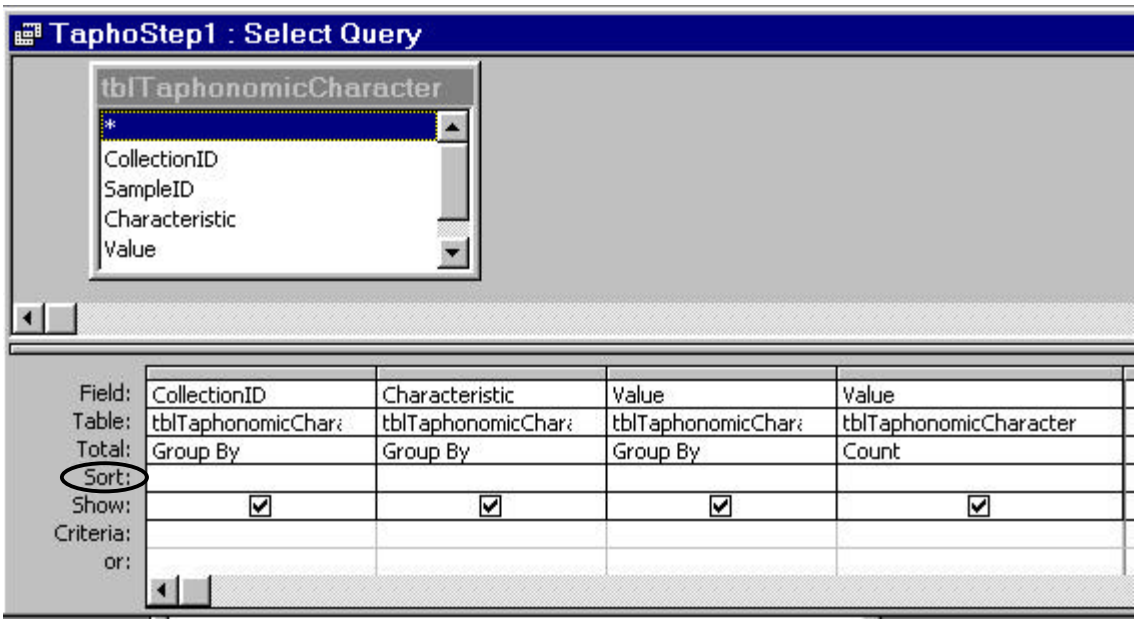


Figure 6.2 This screen capture represents the first step in creating a UNION query for gathering all taphonomic data stored in AARDB. The Total clause should be set to GROUP BY for each grouping attribute, with the repeated Value attribute subject to an aggregate function.

The third and final step of this process assigns the regional and sample constraints to the records of our taphonomic data query and reformats the data to a spreadsheet-like form. Design a new query using the query Design view and add the tables tblLocation, tblCollectSample and the query from step two (TaphoUNION in the example of Figure 6.3). Next we should add the attributes we are interested in, such as UDAMS, Lat, Long and all the attributes from the UNION query. Now we can specify the latitudinal range for our region of interest by typing in the constraint in the Criteria section of the field editor (Figure 6.4). To simplify the amount of data to work with, the Characteristic field of the query can be constrained to only giving "Color" records (Figure 6.4).

When this final query is executed all sample collection totals of taphonomic characteristics are retrieved from the database. The constraints specified above limit the

data to just the region of interest. These records can be brought into spreadsheet software for further analysis or accessed by a GIS for spatial display of mollusk taphonomy for the Atlantic Coastal Plain.

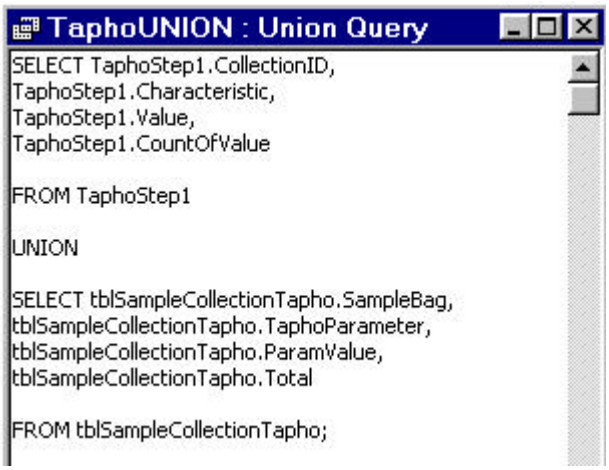


Figure 6.3 SQL View showing the SQL syntax to perform a UNION on the first two queries created in this section (TaphoStep1 and TaphoStep2).

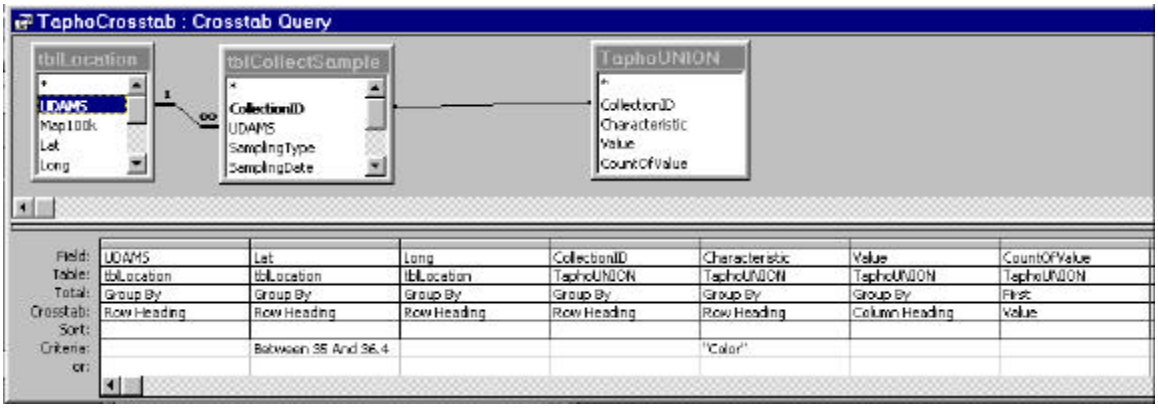


Figure 6.4 Query Design view for final step in taphonomic character totals. Notice the Latitude and Characteristic fields contain constraints to restrict the query.

6.2.5 Spreadsheet Analysis

There are several options for analyzing the data in our database. MS Access comes equipped with the ability to calculate simple aggregate functions and generate simple graphs. However for more robust analyses, it is necessary to use spreadsheet software such as Microsoft® Excel. Sharing data with other software in the Windows® environment can be done several ways, including cutting/pasting records, ODBC, data Import/Export options and through *Automation* via a host programming language. One example of Automation through VBA is presented in aardb.mdb using the Discriminant Function Query form available from the Data Select switchboard (Figure 5.10).

The Microsoft® Excel spreadsheet (DiscrimFunction.xls) in Appendix IV (the included CD) has stored functions to calculate a discriminate score for each AAR GC analysis imported from the Discriminant Function Query form. The query constraints would be the same as above, though we can also add that the query only retrieve records with a specific D/L value range. This is the way we narrow down the records to apply the discriminant analysis for a later join of all pertinent records in our GIS. Executing the Run button on the Discriminant Function Query form queries the database and automatically opens up the Excel spreadsheet and updates the table and graphs.

Discriminant scores for two different functions are included in the spreadsheet and are calculated by using the raw coefficients for Alanine, Glutamic Acid, Leucine, Phenylalanine (DScr4-Val) and Valine (DScr5), derived from the discriminant analysis (see Chapter 3). These two discriminant score calculations were chosen based on results of the test sample analysis (Appendix II) and for flexibility in analyzing samples without Valine resolved.

Once the records of interest are in the Excel spreadsheet, the discriminant scores of these analyses can be shared with other software through an established ODBC connection (see Appendix II). Therefore, we can retrieve this information from our GIS project as with data from AARDB, as a database source.

6.3 Data Exhibition and Analysis Using a GIS

6.3.1 Setting Up the Map

Now that the pertinent data are prepared, we can access the data through a GIS for spatial examination. As part of Appendix IV, an ArcGIS® version 8.3 project file (SiteMap.mxd) is included with a general map of the northeast North Carolina region. In addition, all solitary data tables, grids and digital line graphs (DLG) exhibited in this work are also available within the folder \AAR_Database\MyGIS of Appendix IV. A basic knowledge of ArcGIS® software is helpful for the remainder of this chapter, although basic instructions are provided in the text.

Figure 6.5 shows a screen capture of an opened ArcMap™ document with pertinent data already added. The content window in the left part of the view presents the data layers and tables available for display. Database views and tables are brought into our project through the Add Data button on the AARDB Toolbar (Figure 6.5) or through the File menu option and then specifying database connection as the source.

Geographic data can be plotted from data tables as specified for ArcGIS® ArcMap™, however, the Add ResultQuery button (gray button) on the AARDB Toolbar (Figure 6.5) automates import of the ResultQuery records we created earlier and plots sampling sites as an event theme. Once these data have been plotted on our map, we can join these data with records brought in from DiscrimFunction.xls (NENCDiscrim ODBC

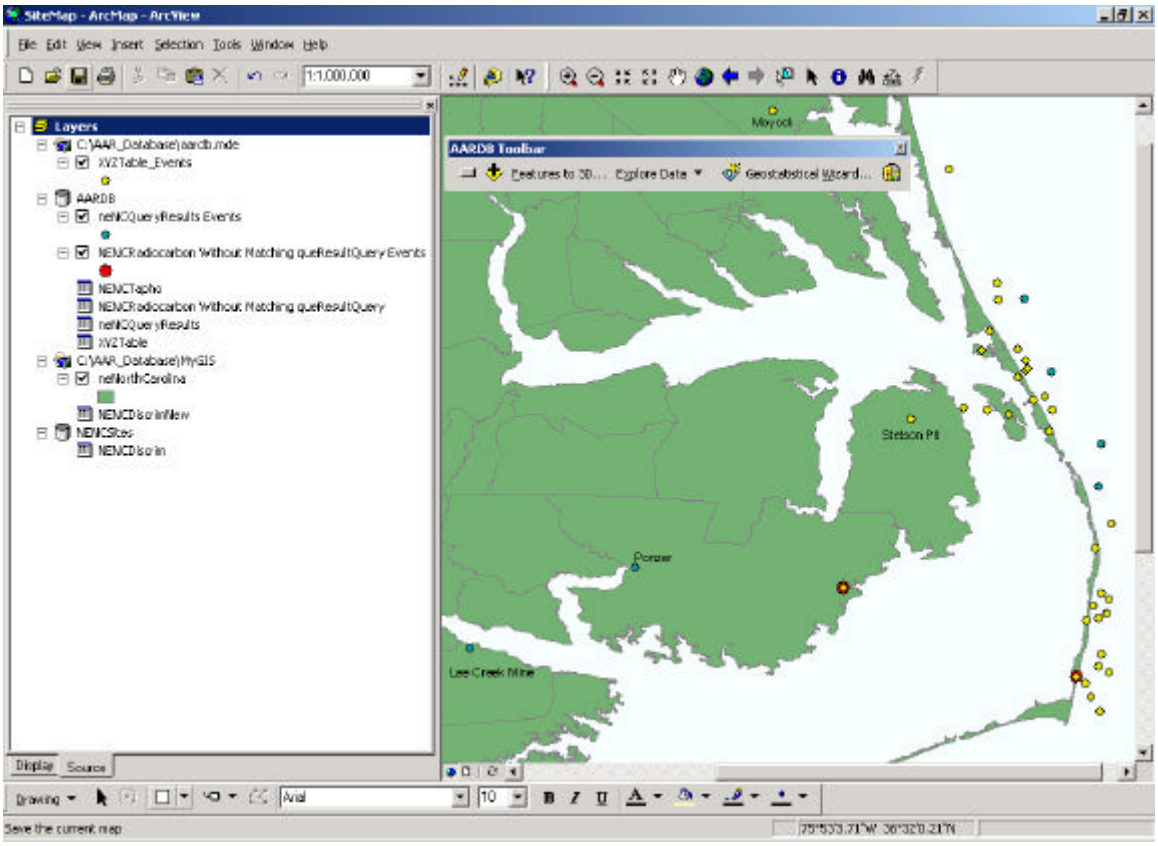


Figure 6.5 Screen capture of a GIS site map with specialized toolbar (AARDB Toolbar). The file from which this figure was taken is included as \AAR_Database\MyGIS\SiteMap.mxd in Appendix IV.

data source) or any other source based on identical attributes. For AAR analysis records, joining the data sets entails matching up SubSampleID from both data sets. If a join is specified for the ResultQuery data set, then data from the joined table will be displayed for any records corresponding in ResultQuery.

Further data manipulation may be necessary as one considers pertinent information for spatial analysis. ArcMap™ allows users to filter data tables using a query building graphical interface that employs SQL commands (Figure 6.6). Filtering refines the data set based on a definition query such that the original source query is not affected. However, analyses performed within ArcMap™ only consider filtered records.

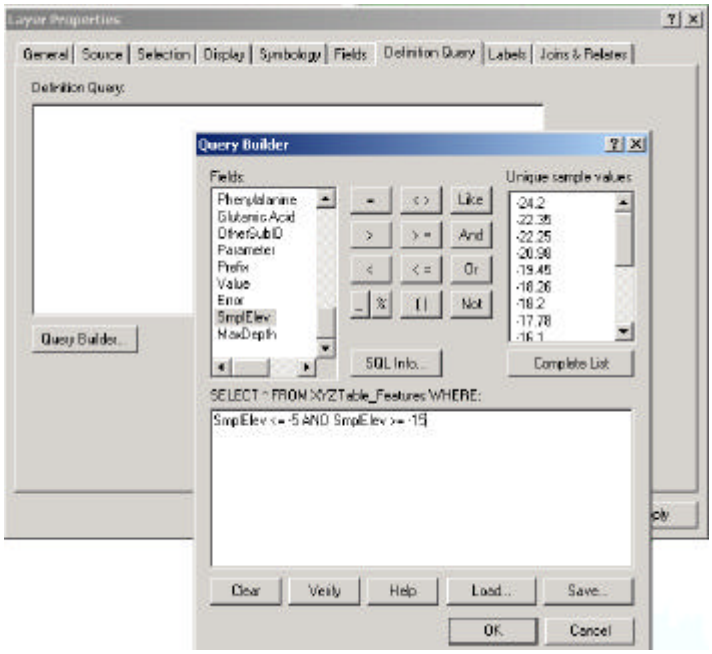


Figure 6.6 Screen capture of query builder in ArcMap™. SQL commands are utilized to specify a WHERE clause for filtering the desired records.

6.3.2 Spatial Analysis of Data

6.3.2.1 Presentation in 2D

Studies have been done looking at the occurrence of Pleistocene shell material washing up on modern beaches (e.g., Wehmiller et al., 1995; Bart, 2001). It is also hypothesized by Wehmiller et al. (1995) that the ages of beach shells might be used as tracers of the coastal processes where subsurface chronostratigraphy is known. Besides relative ages determined from AAR analyses, taphonomic characteristics have also given clues to the age of shells washing up onto beaches (Wehmiller et al., 1995). To study these concepts, we can examine distributions of D/L values and shell taphonomic characteristics for reworked mollusks within our region using different spatial and

statistical tools within ArcMap™. We can look at surface samples from AARDB for two-dimensional examination.

Using the definition query option for our ResultQuery, we can select records where the SamplingType attribute is equal to "Surface." Because several analyses are represented for a particular sampling site, we need to look at the distribution of D/L values (Leucine) for all of our sites of interest. These sites can be portrayed in several ways using the Symbology tab of the layer properties in ArcMap™. With the Explore Data option on the AARDB Toolbar (Figure 6.5) we can also make use of exploratory statistical methods for examining attributes of these beach samples.

Displayed in Figure 6.7 are surface sampling sites where we find *Mercenaria* with determined D/L Leucine values. Each site is represented by a gradational symbol based on the D/L Leucine value of a particular record. The histogram in Figure 6.7 shows the distribution of D/L values from sites along the Outer Banks. In the figure, the Leucine ratio bin of 0.34 to 0.395 is highlighted to show the location of these particular samples. This ratio range is significant because it represents at least a last interglacial age, approximately 70 to 120 ka, based on a U-series coral date from Stetson Pit (Szabo, 1985; York et al., 1989; Riggs et al., 1992). These last interglacial age shells are found along Hatteras Island at Rodanthe, Avon and along the southern shore of Cape Hatteras (Figure 6.7). As to be expected for a modern beach, Holocene shells are distributed throughout all of the sites. Moreover, the north most site, Currituck beach, contain only Holocene age shells.

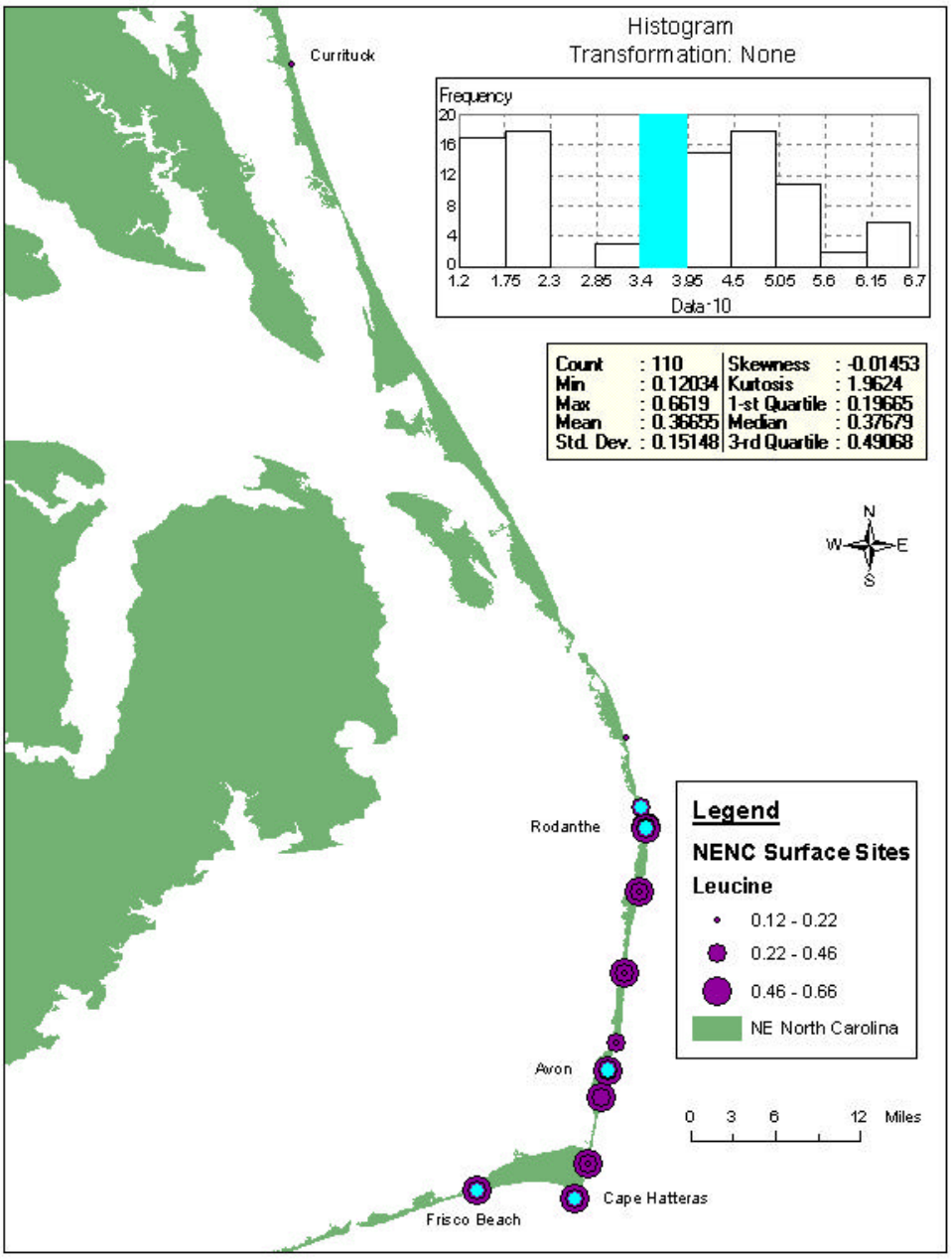


Figure 6.7 Frequency distribution of D/L Leucine values for *Mercenaria* surface samples. The histogram shows the distribution of D/L values (i.e., relative ages) for surface samples collected along the shore. The highlighted samples correspond to the highlighted D/L Leucine bin in the histogram.

If we consider the pattern of taphonomic characteristics, in particular color, for these sites, we can get a good picture of the distribution of reworked Pleistocene shells along our sites of interest. Figure 6.8 shows an interpolated map (inverse distance weighting method) of dark shell color for shells collected from beach transects. The interpolated grid was generated from the NENCTapho data set created from our UNION query (see above). Interpolation parameters were determined using the Geostastical Wizard on the AARDB Toolbar (Figure 6.5), interpolating percent of totals for the sum of black and gray colored shells.

A trend of increasing frequency of darker shells is apparent from north to south along Hatteras Island. Dark color is significant because all Pleistocene age *Mercenaria* shells analyzed by UDAL from coastal North Carolina beaches exhibit a dark gray color (Wehmiller pers. comm., 2003). This pattern corresponds well with the distribution of higher D/L values (i.e., older) we find for these southern most sites (Figure 6.7). The trend exhibited in Figure 6.8 also suggests the age of the sediment source just offshore of these beach collection sites.

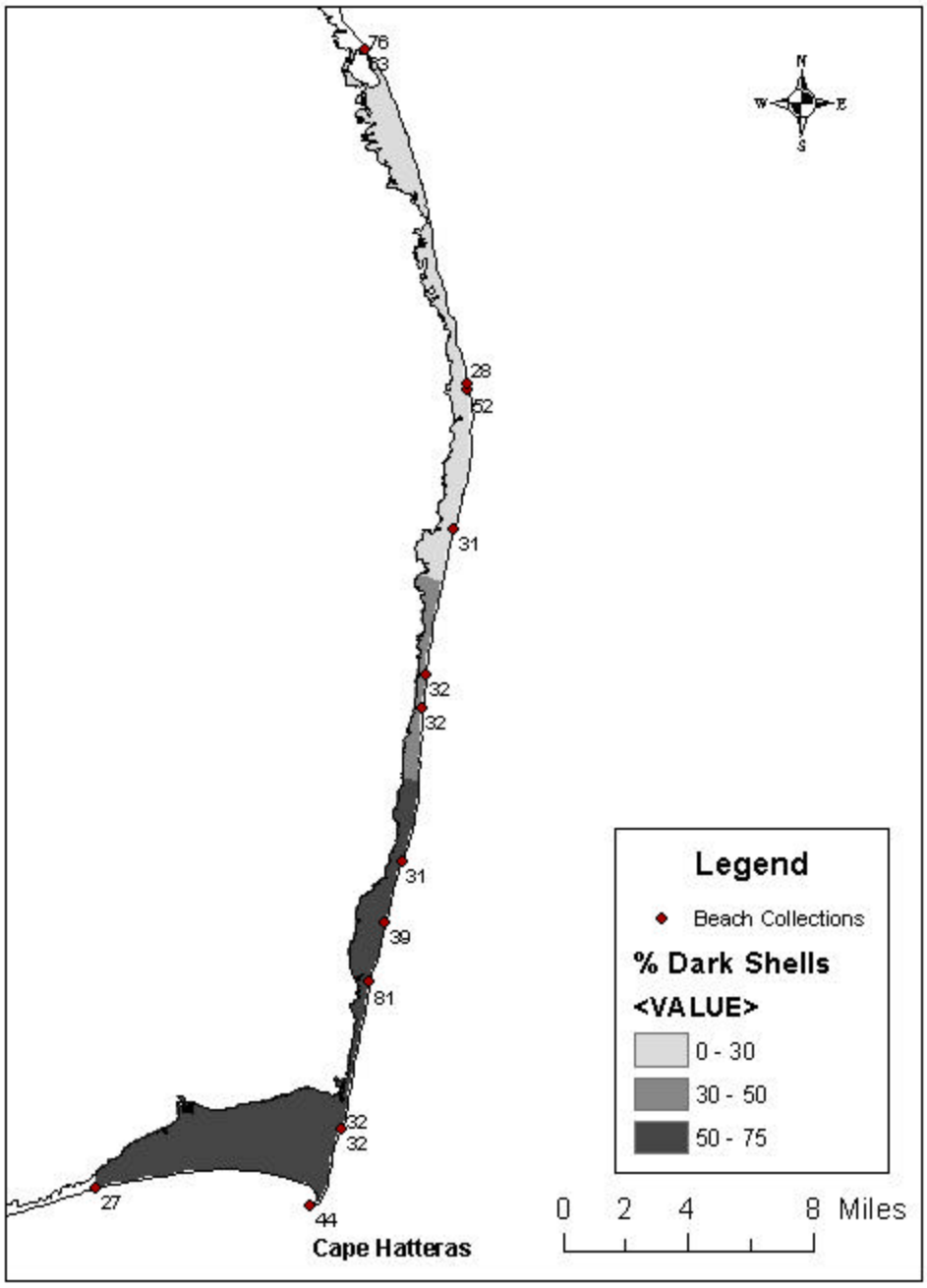
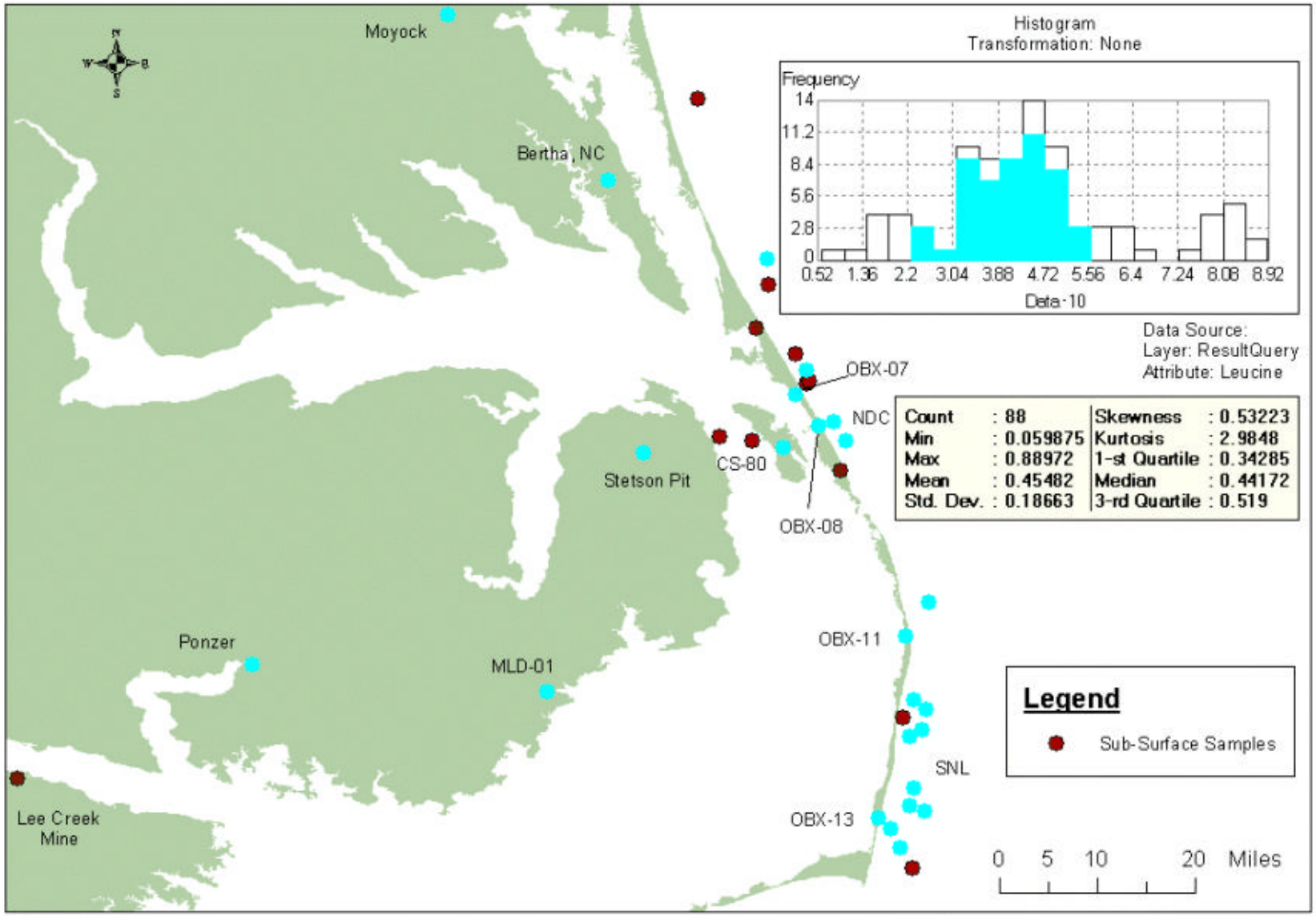


Figure 6.8 IDW interpolation grid of dark colored *Mercenaria* shells along Hatteras Island. The frequency of dark shells (i.e., pre-Holocene shells) increases from north to south along beach transects. Numbers adjacent sites represent the number of collected shells.

As with surface samples, we can generate exploratory statistics for ResultQuery samples with an elevation component. While examining these data would work best in three dimensions, exploratory statistics can identify patterns worth investigating further. Figure 6.9 shows the results of creating a histogram for only Inland Core, Offshore Core, Excavation/Exposure and Underwater Grab samples of ResultQuery. The frequency pattern is similar to that shown for the surface samples in Figure 6.7. Not surprisingly, what is preserved in the subsurface geologic record of a dynamic coastal system is also represented by what is washed up on shore. However, considerably fewer Holocene shells are represented at depth. Therefore, the median D/L Leucine value for samples at depth is greater than for surface samples (0.43 compared with 0.38).

By clicking on the bars of the histogram, we can highlight samples exhibiting a certain range of D/L Leucine. Figure 6.9 highlights sampling sites with Leucine values within a Pleistocene peak (approximate D/L Leucine range of 0.25 to 0.55). Considering the resolution of the AAR method this D/L range is very broad and it is likely that several significant aminozones are represented by this distribution. Unfortunately, any supposed aminozones are undifferentiated by simple exploratory methods. Nonetheless, it is useful to note the geographic distribution for this D/L Leucine range within the region (Figure 6.9).

Figure 6.9 A histogram for subsurface and excavation/exposure samples. Peaks similar to those exhibited for the surface samples are present within the subsurface. Histogram bins that are highlighted represent late Pleistocene samples with their sample locations also highlighted on the map. The histogram represents the frequency of particular D/L Leucine bins from about 0.05 to 0.89. Exploratory statistics were generated from only the highlighted samples.


Moyock
Bertha, NC
Histogram
Transformation: None
Frequency
14
11.2
8.4
5.6
2.8
0
0.52 1.36 2.2 3.04 3.88 4.72 5.56 6.4 7.24 8.08 8.92
Data-10
Data Source:
Layer: ResultQuery
Attribute: Leucine
OBX-07
NDC
CS-80
Stetson Pit
OBX-08
Count : 88
Min : 0.059875
Max : 0.88972
Mean : 0.45482
Std. Dev. : 0.18663
Skewness : 0.53223
Kurtosis : 2.9848
1-st Quartile : 0.34285
Median : 0.44172
3-rd Quartile : 0.519
Ponzer
MLD-01
OBX-11
SNL
OBX-13
Lee Creek
Mine
Legend
Sub-Surface Samples
0 5 10 20 Miles

6.3.2.2 Presentation in 3D

Because geologic data often have an elevation or depth component, it is more helpful to view the data with three-dimensional software. GIS software has progressively improved its rendering of 3D data and we can take advantage of this technology for investigating aminostratigraphic results.

Figure 6.10 shows a screen capture of ArcScene™, a three-dimensional viewing component of ArcGIS™. We can create three-dimensional GIS files (e.g., a "shapefile") in ArcMap™ by using the "Features to 3D" option on the AARDB Toolbar (Figure 6.5). Otherwise, most GIS files or event themes can be exhibited in three dimensions when a specific attribute is assigned for the Z value in the Base Heights tab of the layer's properties menu. We can also create grid files that represent an interpolated Z component to depict in ArcScene™. Database views are imported into ArcScene™ the same way as with ArcMap™, via the Add Data button (Figure 6.10). Interpolated maps for this study were created using the Geostatistical Wizard in ArcMap™ and then exported as a grid file.

Browsing sample points in three dimensions allows visualizing the absolute positions of samples and the spatial distribution of D/L values. With available radiocarbon analyses, D/L values and discriminant scores for AAR analyses with ambiguous D/L values, we can portray a three-dimensional distribution of geochronological results for the subsurface of coastal northeast North Carolina. The three-dimensional image portrayed in Figure 6.10 and available as 3DSiteMap.sdx in Appendix IV, show the distribution of Holocene and Pleistocene samples.

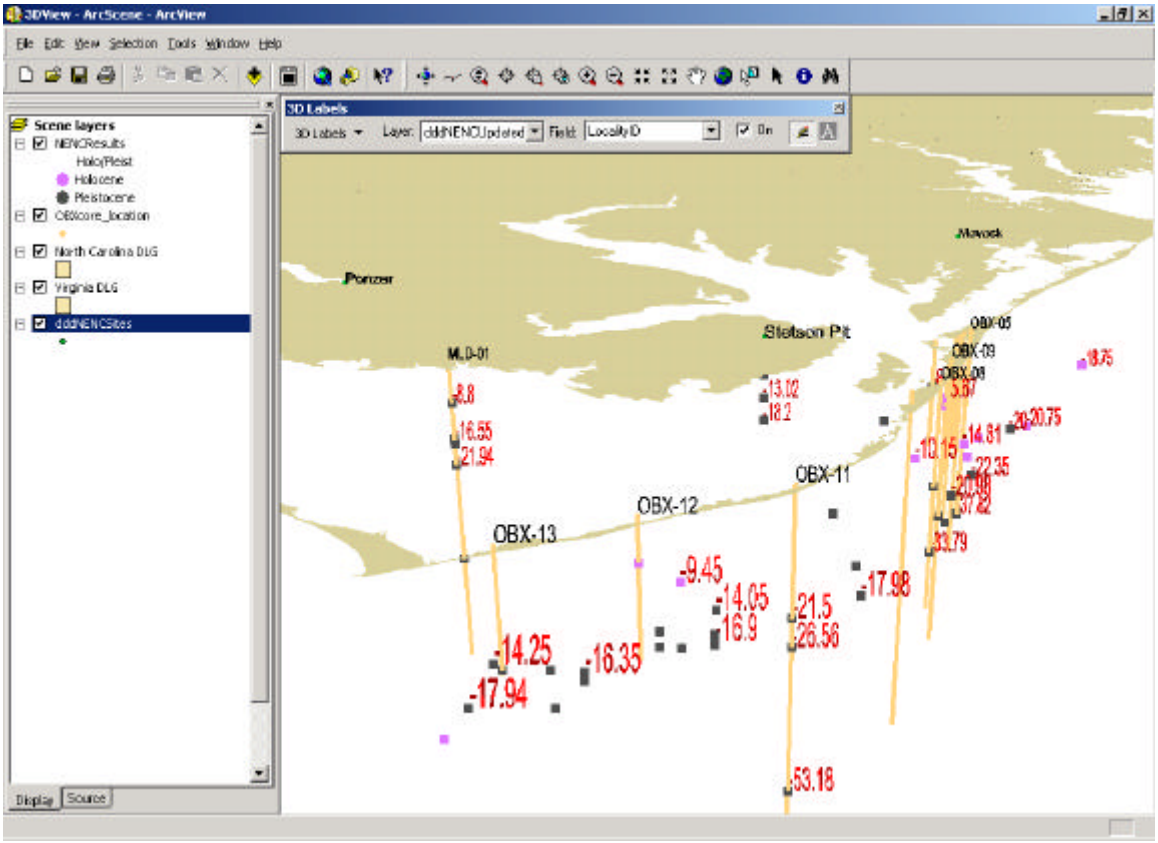


Figure 6.10 Screen capture of open ArcScene™ document with sample locations and cores displayed for northeast North Carolina. Visualization tools include 3D labeling and a Navigation toolbar for complete three-dimensional view rotation.

More sophisticated spatial analysis is also possible utilizing tools like the Geostatistical Wizard. For this work, spatial analysis was performed in ArcMap™ and then visualized within ArcScene™ (see Appendix IV, \MyGIS\3DSiteMap.sxd).

A surface representing late to mid-Pleistocene samples, according to geochronologic estimates of York et al. (1989), is portrayed in Figure 6.11 along with sample locations. This interpolated surface represents the elevations of samples that were within highlighted D/L Leucine bins of the histogram in Figure 6.9. Surface validation for the Pleistocene surface (Figure 6.11) is included in Appendix III. An attempt was

made to interpolate a Holocene surface but cross-correlation plots of the model exhibited a poor fit of predicted data with measured data. An example of an interpolated Holocene surface was nonetheless included with Appendix IV (\AAR_Database\MyGIS\SiteMap.mxd).

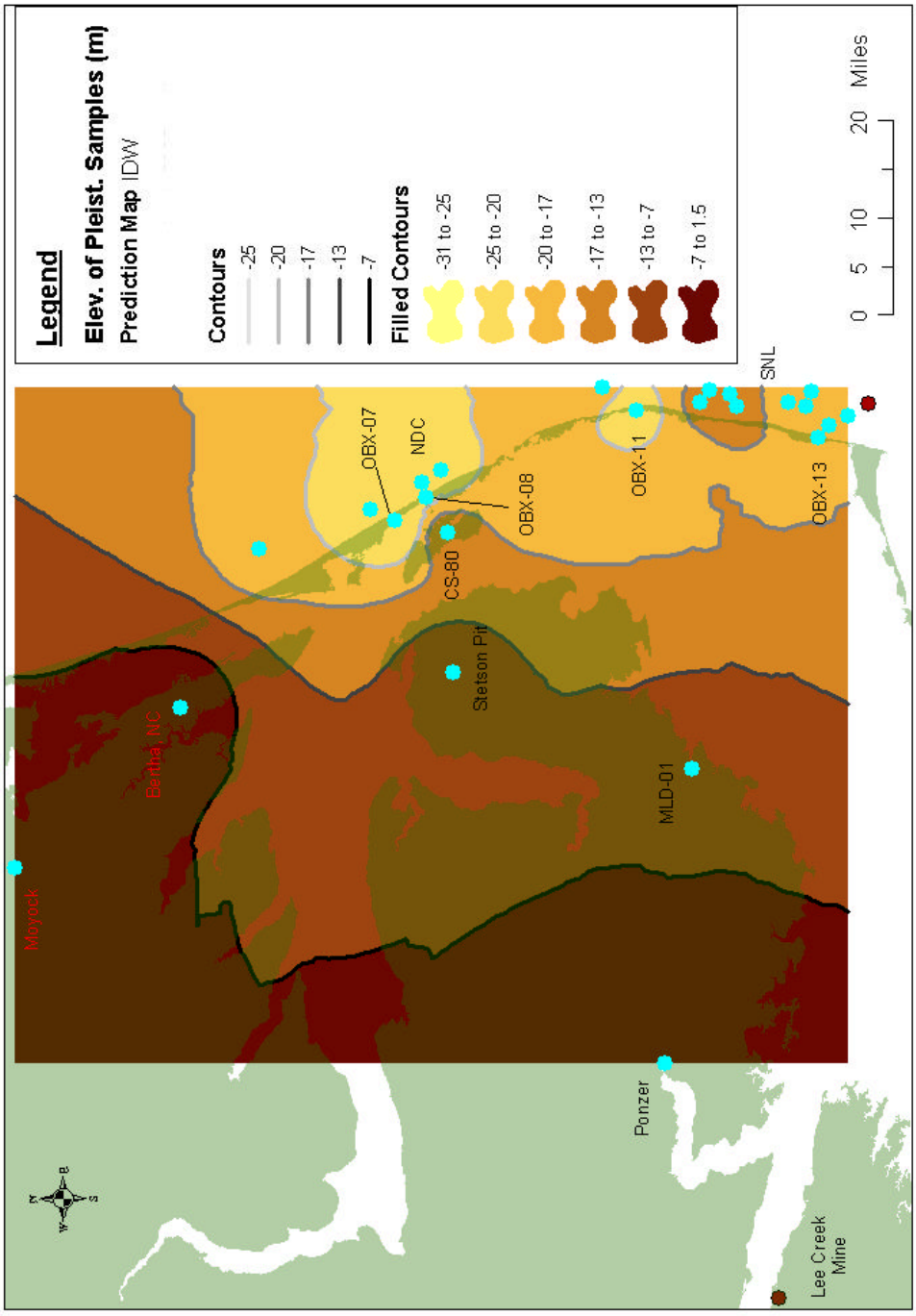
Pleistocene headlands (i.e., subaerial headlands) are apparent on the north end of the map at Stetson Pit (Dare Headland of Riggs et al., 1995) and as a gently sloping headland feature in the southern end of the map (Figure 6.11). Also apparent from Figure 6.11 are depressions in the late Pleistocene surface that represent a lack of Pleistocene age shells collected from cores. The northernmost surface low corresponds with inlet-filled channels described in Riggs et al. (1995), where Holocene samples are found as much as 20 to 30 meters below MSL. The southern most low surrounds the OBX-11 core hole, for which no samples above –20 MSL have so far been analyzed. Therefore, this low may be an artifact of the interpolation method.

Another attempt at interpolating the values between sampling sites was made from the D/L Leucine values of the same samples used to create the late Pleistocene surface (Figure 6.11). The D/L Leucine surface was then draped over the late Pleistocene surface for a three-dimensional context in ArcScene™. This surface represents the interpolated distribution of D/L values as calculated with the IDW method. This model is not presented here but can be viewed in Appendix IV (\AAR_Database\MyGIS\3DSiteMap.sxd).

The spatial interpolations attempted here may prove to be a useful way to model antecedent surfaces based on aminostratigraphic data. The cross-correlation plot (Appendix III) of the late Pleistocene surface (Figure 6.11) shows good agreement with

measured and predicted values. Cross-correlation plots for the D/L Leucine surface show fair correlation, however, a plot of the residuals against measured values suggests some heteroscedasticity of the data (Appendix III). Artifacts of the interpolation method such as isolated D/L value patches are also apparent (see Appendix IV, \AAR_Database\MyGIS\SiteMap.mxd). Method parameters for these interpolated surfaces are also included in Appendix III.

Figure 6.11 Interpolated late Pleistocene surface created from the elevations of sample locations with D/L Leucine values ranging from approximately 0.32 to 0.52. Subaerial headland areas are apparent below Dare County (Stetson Pit), the north bank of Albemarle Sound, and gently sloping headland to the south near core MLD-01. A Holocene valley-filled depression is apparent from northernmost depression in late Pleistocene surface.


Legend
Elev. of Pleist. Samples (m)
Prediction Map IDW
Contours
-25
-20
-17
-13
-7
Filled Contours
-31 to -25
-25 to -20
-20 to -17
-17 to -13
-13 to -7
-7 to 1.5
0 5 10 20 Miles
Moyock
Bertha, NC
Stetson Pit
CS-80
OBX-07
NDC
OBX-08
OBX-11
OBX-13
SNL
MLD-01
Ponzer
Lee Creek Mine

CHAPTER 7

DISCUSSION AND CONCLUDING REMARKS

7.1 Statistically Assessing Aminozones

An important objective of AAR geochronologic studies of Atlantic Coastal Plain sites is designating statistically distinct aminozones within the framework of local stratigraphy. Several workers have successfully determined the significance of D/L ratio clusters for particular coastal plain localities. For example York (1990) performed statistical t-test on *Mulinia* from Stetson Pit, North Carolina to differentiate three Pleistocene D/L ratio clusters. Similarly, Corrado et al. (1986) and Harris (2000) have explored statistically observable D/L ratio clusters for coastal plain sites in South Carolina. Such clusters, when considered with local stratigraphy can lead to reasonable estimates of aminozones exhibited in an area (e.g., Harris, 2000).

In this work, we investigated a statistical method of differentiating between early Holocene and late Pleistocene samples from the region using discriminant analysis (Appendix II). Besides overcoming the variability inherent with the AAR method as discussed in Chapter 2, resolution of Quaternary coastal plain stratigraphy by aminostratigraphic means is complicated by another factor; there is usually little relative difference in D/L values between early Holocene and late Pleistocene samples. This is not only a natural consequence of racemization kinetics but also stems from colder effective temperatures experienced by Pleistocene age shells (Wehmiller et al., 2002).

Furthermore, resolving Quaternary stratigraphy is challenging for Atlantic Coastal Plain sites because preserved Pleistocene deposits are generally confined to a topographic range of only 15 meters above sea level (Wehmiller and Miller, 2000). Consequently, geochronologic resolution of the AAR method must often distinguish units with uncertain stratigraphic control and with a paucity of samples representing the latest glacial period in the region (Wehmiller and Miller, 2000).

In using discriminant analysis, we utilized a multivariate statistical method for assigning a relative age (Holocene or Pleistocene) to an analyzed sample that fell within an ambiguous range of D/L values (see Figure 2.4). Application of discriminant analysis to samples collected from the study region proved useful for these samples of uncertain ages. For example, a *Mercenaria* sample, JW2003-156-004, near the top of the MLD-01 core (approximately 9 meters below MSL) gave a D/L Leucine value of 0.27, within the ambiguous range displayed in Figure 2.4. The discriminant analysis classified this sample, with a high probability, within the Pleistocene group (see Appendix IV, DiscrimFunction.xls). A radiocarbon analysis of a *Mercenaria*, JW2003-156-002, collected from the same depth gave a result of >52,000 radiocarbon years. Furthermore, a gastropod sample, JW2003-154, collected at about 3 meters below MSL and dated at 39,900 radiocarbon years helps to confirm the late Pleistocene age assignment for the *Mercenaria* sample JW2003-156-004.

The frequency distribution of D/L Leucine values for samples in this region (see histogram in Figure 6.9) highlight the necessity for applying more rigorous statistical methods to differentiate aminozones. It may be possible to separate late Pleistocene D/L clusters using methods like discriminant analysis. However, discriminant analysis

depends on *a priori* knowledge of the relations between observations (Davis, 1986). So the study region would have to exhibit good stratigraphic control in order to establish before hand the characteristics of groups to be differentiated.

Furthermore, it may not be possible to resolve these assumed clusters by exploiting intrageneric differences. For instance, some researchers have commented on the unreliability of Alanine for geochronologic analysis because it can be created from the degradation of more complex amino acids (Miller and Brigham-Grette, 1989), and still other amino acids have been shown to exhibit reversed racemization over time for some genera (Kimber and Griffin, 1987).

The successful application of discriminant analysis in this work, for which Alanine was determined to be a necessary variable for significant separation between group means (see Appendix II), may be because of the robustness of the particular genus utilized here, *Mercenaria*, but may also be due to the relatively young ages of the two groups of study (i.e., early Holocene and late Pleistocene). Significant degradation of more complex amino acids may yet have affected Alanine abundances and, therefore, unduly influence the discrimination processes. Moreover, at least for northeast North Carolina *Mercenaria* samples, a plot of D/L Alanine versus radiocarbon age shows a similar trend of increasing ratio with increasing age similar to D/L Leucine (Figure 2.4).

Pursuing more rigorous statistical methods for data analysis, while continuing to amass samples from this region, will likely improve our understanding of Quaternary stratigraphy of the North Carolina coastal plain. Surface models similar to the one shown in Figure 6.11 could be refined using statistically significant D/L clusters and become a useful visualization and analysis tool.

7.2 Assessment of Spatial Interpolation of AAR Ratios

The application of spatial interpolation techniques to AAR analyses for the region proved useful for visual inspection of regional aminostratigraphy. Though, one should be mindful of artifacts of the interpolation method used, such as the surface low surrounding OBX-11 in Figure 6.11. This pattern, however, would likely change with future analyses further up section in OBX-11. In addition, the inverse distance weighting method of interpolation tends to generate a bulls-eye pattern for solitary samples or areas with low sample densities.

The northernmost surface low in the interpolated late Pleistocene surface (Figure 6.11) likely represents a sum effect of several Holocene fill sequences, that is, a lack of subsurface Pleistocene age samples collected from this area. Corroborating this interpretation, coast parallel cross-sections in Riggs et al. (1992) depict inlet-filled channels and Holocene valley-fill of the paleo-Roanoke River/Albemarle estuarine system between Kitty Hawk and Oregon Inlet. Furthermore, work by the Coastal Carolina Project has also revealed similar facies in this area (Thieler et al., 2003). However, comparison of the Figure 6.11 model with a cross-section (Figure 7.1) generated from litho and biostratigraphy, along with geochronology (AAR and radiocarbon), by the Coastal Carolina Project team shows the severe problems with solely interpolating the AAR data.

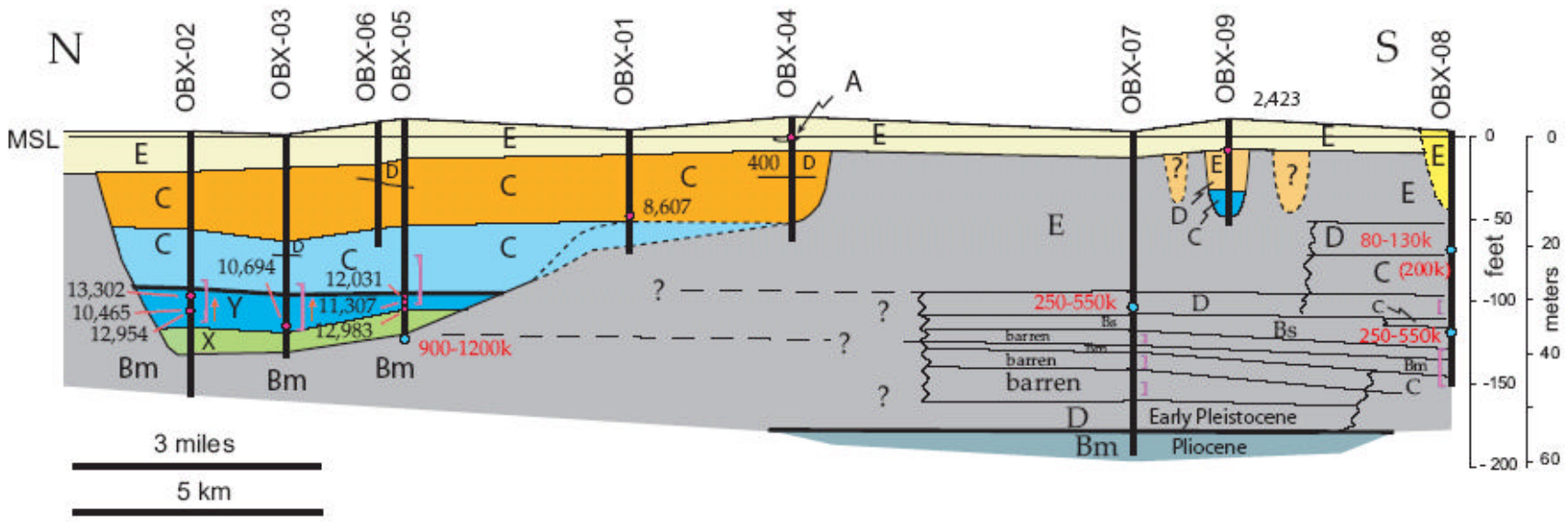
Figure 7.1 represents about a seventeen-mile shore parallel cross-section from Kitty Hawk, NC (OBX-02) to Whalebone, NC (OBX-08) (see Figures 1.2 and 6.8). Numerical ages are assigned to specific samples in the figure, black numbers are from radiocarbon analyses and red numbers represent AAR analyses (Figure 7.1). The main

units are differentiated by lithology and microfossil assemblages and are distinguished in the cross-section by various colors and black dividing lines (Figure 7.1).

The dimensions of the Holocene valley-fill of the paleo-Roanoke River/Albemarle estuary apparent in Figure 6.11 do not match that of the detailed cross-section (Figure 7.1). For example, Figure 7.1 shows the deepest part of the Holocene valley-fill running through Kitty Hawk and Nags Head, NC (about -40 meters from MSL, while the Pleistocene surface depths in Figure 6.11 are about 10-15 meters too high (about -25 meters from MSL). Moreover, Figure 6.11 shows the Pleistocene units much deeper in the area of OBX-07 and OBX-08, while detailed studying of the OBX cores shows a shallower marine Pleistocene unit dissected by smaller Holocene channel fill deposits (Figure 7.1).

Clearly, the AAR data are not as densely distributed enough to define the units with the detail displayed in Figure 7.1. Furthermore, Figure 6.11 was generated from Pleistocene samples with a large range of D/L Leucine values (about 0.32 to 0.52, see Figure 6.9). More narrowly chosen D/L clusters, only representing latest Pleistocene samples, for instance, along with more AAR analyses would likely generate a more accurate late Pleistocene surface map. Future models like the one shown in Figure 6.11 will need to take advantage of statistical methods to meaningfully classify or discriminate D/L clusters and allow for more detailed models to be developed.

Figure 7.1 A seventeen mile, shore parallel cross-section generated from studying OBX cores 01 through 09. Numerical ages are assigned to specific samples in the figure, black numbers are from radiocarbon analyses and red numbers represent AAR analyses. Lithology, foraminiferal assemblages and the presence of diatoms differentiate the main units in the figure and are distinguished by various colors and solid or hashed black dividing lines. Cross-section produced by members of the Coastal Carolina Project.



Foraminifera

Assemblage A salt marsh
Assemblage B open shelf
Assemblage C open shelf/bay
Assemblage D open shelf/bay/inlet (?)
Assemblage E barrier island

muddy, low oxygen (m)
shelly sand (s)

Diatoms

[intervals sampled for diatoms;
increasing marine influence upcore within tidalite unit in OBX-02 and OBX-03.

7.3 Data Manipulation and Sharing Methods

Data sharing methods utilized for this work were very useful for a comprehensive investigation of aminostratigraphic results for the northeastern North Carolina region. The ability to combine several data analysis techniques in a single spatial context can be a powerful tool for stratigraphic interpretation.

However, data sharing between software on a single platform was not without its flaws. Attribute data types did not always translate well when utilizing Automation from AARDB to both Microsoft® Excel and ArcMap™. Workarounds creating stand-alone files in ArcMap™ were implemented to resolve this issue. For example, importing data from DiscrimFunction.xls (NENCDiscrim ODBC source) into ArcMap™ changed the attribute data type of SubSampleID, therefore hindering the ability to join this dataset with records from AARDB in ArcMap™.

Furthermore, attribute names from data sources are often truncated when viewed in ArcGIS® software. While this was mostly a minor inconvenience, attributes with similar names are indistinguishable except for an arbitrary number appended to field names by the ArcGIS® software. ArcMap™ also cannot handle attribute names with spaces and often generates errors when analysis is attempted on these fields.

Finally, for examining AAR ratios queried from AARDB in ArcGIS™, it was often necessary to filter the data further than the initial query constraints (see Chapter 6.1). This was often apparent after attempts at spatial analysis identified troublesome data points. These data points included reworked fossil mollusks from a subsurface section and records from the database where AAR sub-sampling incorporated shell material other than the middle carbonate layer (see Chapter 3). These data points

exhibited discordant values that had to be filtered out for any spatial analysis. The northeast North Carolina query developed in Chapter 6 and saved in AARDB (Appendix IV) as NENCResults excludes the sub-samples that were not produced utilizing the middle umbo layer of *Mercenaria*.

7.4 Concluding Remarks

In conclusion, this work demonstrates the utility of efficient data organization for comprehensive data exploration. AARDB allows for multi-parameter searching and querying of UDAL's AAR database through a user-friendly graphical interface. Additionally, data sharing methods, enhanced by a relational design, applied here enable more sophisticated data exploration including advanced visualization techniques using GIS.

AAR and radiocarbon analyses analyzed for the North Carolina coastal plain since the early 1980's represent a wealth of geochronologic data that is fundamental for stratigraphic characterization of the Albemarle Embayment. Efforts by the Coastal Carolina Project would benefit greatly from a synthesis of geochronological results for framework geology studies coastal plain deposits. The database management effort presented here suggests a platform for a more quantitative assessment of AAR data for stratigraphic analysis and builds on previous efforts (e.g., Wehmiller et al., 1988) to synthesize geochronologic data for the region.

Future endeavors will likely include more in depth statistical and spatial analysis techniques for aminostratigraphic investigations. In addition, AARDB will likely grow to include more samples from coastal North Carolina sites, as well as sites all along the

Atlantic Coastal Plain. Such efforts will likely improve the resolution of the aminostratigraphy for coastal North Carolina.

It would also be useful to take advantage of existing technology for Web access to AARDB as a means to facilitate data sharing among interested parties outside of UDAL. Application of an Internet map server (IMS) as the web front-end of AARDB, thereby presenting the data within a spatial context, may prove to be functionally most practical. As for the cooperative research currently underway by the Coastal Carolina Project, a Web based aminostratigraphy database may act as a springboard for future data management and sharing efforts endeavored by the project.

APPENDIX I

DEVELOPMENT OF AARDB

A-1 Overview

The development of the AAR database for UDAL entailed reorganizing an existing database of AAR analyses from coastal plain sites. Work along the preliminary level of the database contains sites from coastal North Carolina, South Carolina and southeastern Virginia and represents data collected over the past two decades.

Earlier reports described details of the working database of coastal plain AAR analyses, most notably Wehmiller et al. (1988), and presented the classification of sites based on geographic regions of similar temperature regimes (aminostratigraphic regions I through V, Wehmiller et al., 1988). From these regions a system of identification for sampling sites (UDAMS) and samples collected (SampleID) were specified to keep track of samplings. AAR derivatives created in the lab from these samples were further classified and identified (AAR lab number) to help keep track of laboratory processes and analyses.

As is often the case, designing a database entails beginning with a working prototype that needs to be restructured to further meet the growing demands of an active dataset. When moving from flat-file database to a relational format there are guidelines one can follow to ensure proper relational design. Similarly, creating a database from scratch would also follow these same steps. Database design steps are presented here and are adapted from practical design steps presented in Elmasri and Navathe (2000). These

steps can be summarized as, 1) data requirements and database functionality, 2) conceptual design, 3) Choosing a DBMS, 4) database mapping and physical design, and 5) implementation and fine-tuning.

The guidelines presented here do not attempt to represent a comprehensive treatment of relational database development procedures. Numerous other texts deal with this topic and its details more thoroughly and should be consulted for a more adequate description of the database design process. This section merely describes the major steps toward the design of AARDB and can be used for a general overview of database design procedures.

A-2 Data Requirements and Database Functionality

The first step for the designer of a database is to become intimately familiar with the data and data types of the proposed database, as well as with the database's expected functionality. If creating a database from scratch, the data requirements would be based on the design of forms used to collect data. These forms (digital or paper) represent all the aspects of the real-world entity the database would represent.

For AARDB, there existed a legacy of data and constraints that defined the use and transactions of the database, although in a less organized form. This is often the case when developing a database (Elmasri and Navathe, 2000); the database designer inherits the work of those before him or her and must incorporate legacy data with new data management demands. The database designers must familiarize themselves with the data content and processing requirements of the primary users. Often this entails a process of lengthy interviews with the primary users of the database to fully determine data management expectations (Elmasri and Navathe, 2000).

Below is a concise description of most of UDAL's AAR database requirements. The descriptions also include the functionality of certain attributes and how attributes are related in the context of the real-world entity they are describing. A database designer would use such a description to aid the conceptual modeling of the database as well mapping out the physical database.

1. The laboratory collects mollusk samples for chromatographic analysis from a variety of localities. Localities are assigned names and unique numeric identifiers (UDAMS No.) that designate a unique sample place. Corresponding latitude and longitude (and sometimes elevation) are recorded to designate their position in geographic space. Localities are also specified according to type such as a core location, burrow pit, underwater, and beach or ground surface.

2. As samples are collected, information is recorded regarding the relative position of sampling and collections are assigned unique alphanumeric identifiers that contain the initials of the collector or principal investigator. All sample collections will have corresponding locality information.

3. Samples within a collection are designated with their own sample ID that represents the parent collection ID appended with an alphanumeric identifier. Sometimes solitary samples are collected at a locality, in which case the collection ID would also act as a sample ID. Individual samples are also labeled with a sample type identifier based loosely on the genus of the sample (e.g. Mercenaria, Mulinia, etc.).

4. Information stored for samples also includes its physical characteristics at the time of sampling. This includes a sample's taphonomic character and should be based on currently utilized characteristics (i.e. abrasion, fragmentation, color) but allow for changes to the types of observed characteristics collected. Additionally, samples should be photographically documented, though not all samples will have a photograph representing them. Some photographs may represent a collection of samples.

5. Samples are further sub-sampled, generally in a lab though not exclusively, and are uniquely identified. Information regarding the person or entity creating the sub-sample, date sub-sample is created and position on sample from which sub-sample was collected should also be recorded. Sub-samples are created/analyzed by a particular method in a particular laboratory. Sub-samples can also be generated from other sub-samples, for example, in the form of aliquots.

6. Laboratory methods or procedures creating sub-samples are given unique identifiers and are described by the name of the laboratory, the name of the method, generic parameter identification and accuracy limits.

7. Sub-samples from laboratories or sources represent a myriad of analyses that need to be identified by a unique Lab procedure ID. Analysis results from these sources need to accommodate legacy data that at times represent calculated values. Results should be recorded such that they correspond to a single sub-sample with standard error and units also recorded when available.

8. A sub-sample analysis result from the UD laboratory always produces chromatographic information in the form of chromatograms (digital and/or analog). The date of the analysis, a machine reference ID and a general designation of the type of chromatographic device used generally characterize chromatograms.

9. Calculated D/L ratios for specific amino acids are recorded in the database for maintaining legacy data and should be related to its corresponding chromatogram.

10. AAR statistics, mean, standard deviation and number of analyses, should be computed for all chromatographic sub-samples and relatable to other analyses on corresponding samples or sub-samples.

11. Analysis results or other parameters should all be related to a particular sample mollusk so that parameters within the database can be compared.

A-3 Conceptual Design of AARDB

A conceptual design of the database uses a model to graphically describe the database including relations and aspects of entity relationships. Visualization can clarify database functionality and is often used to communicate the database design to non-technical users (Elmasri and Navathe, 2000). A conceptual model needs to include all aspects of the database but should be flexible enough such that changes to the database can be accommodated.

There are several different data models one could use to conceptualize their database. Common ones include the Entity-Relation (ER) model and the enhanced ER

(EER) model. The ER model was chosen for the conceptual design of AARDB with some borrowed aspects of EER (specialization and generalization). Several texts describe the structural concepts of these models, some of which can be found in the Reference section. The concepts of the ER model are only briefly described here.

Figure A-1 is a key to the components of the ER model used for the conceptual design of AARDB. Rectangles represent entities, which are equivalent to tables and a diamond shape represents relationships. A relationship in the ER model does not only refer to the sharing of key attributes between two tables for they can also represent tables with attributes of their own.

Entity names and the roles of relationships are specified in the center of the component. Attributes of entities or relationships are designated by an oval with a stem connected to the corresponding entity or relationship. The names of entities are specified in the center of each oval.

Entities and relationships with double lines signify a weak entity and its identifying relationship, respectively. A weak entity is one that does not have key attributes of its own but instead borrows another entity's primary key (or set of keys) for its identifying attribute, termed a partial key.

Entity relationships are designated by connecting lines and annotated to specify the cardinality ratio of a relationship. Double connecting lines signify total participation of an entity in a relationship, while a single line specifies only partial participation.

Figure A-2 exhibits the conceptual schema of AARDB. This schema was created from the database requirements established above. In the design process, designers may move several times from step 2 (conceptual design) to step 1 back to step 2 in order to

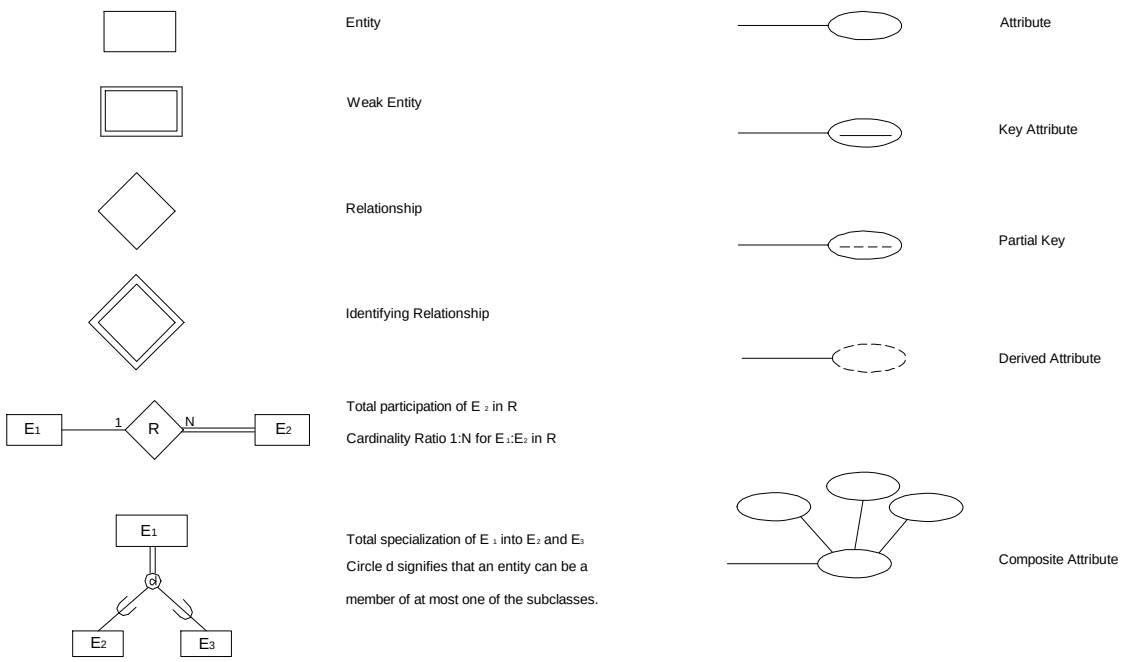
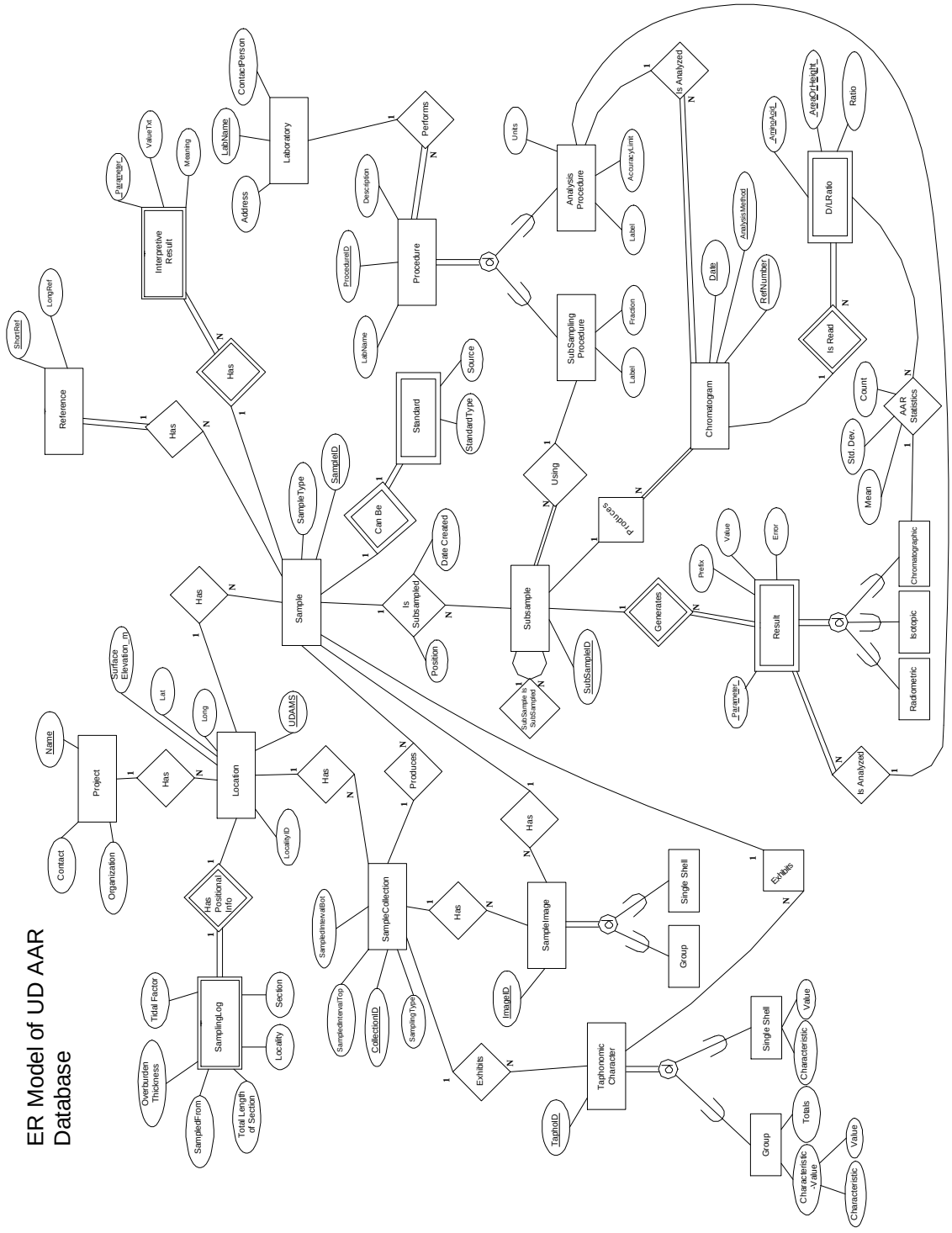


Figure A-1 Summary of the Entity-Relationship model notation. Component explanations are to the right of each model component. After Elmasri and Navathe, 2000.

Figure A-2 Entity-Relationship (ER) model of the UDAL's AAR database. Important database components are explained in the text of Appendix I. This conceptual model is used to guide database designers in creating the necessary tables and relationships of the database.

ER Model of UD AAR Database



refine the database design. The list of data content and functional requirements for AARDB from step 1 could be further refined such that entities and their key attributes are specifically identified.

Generally, the grouping of related attributes creates entities in the conceptual schema. There may be more than one option for grouping attributes and is influenced by the intention of the database and the knowledge of the designer for that particular database. One can get a general sense of the entities to create simply from the process in step 1 and from normalization techniques described in chapter 4. Following through with the conceptual design process further clarifies database entities and relationships.

Conceptualizing the database requirements can be approached using different strategies. The strategy applied for AARDB was an inside-out methodology because we were working with several clearly evident entity types (Location, Sample, Sub-sample). New entities and their relationship with the already evident entities were then added to the schema systematically.

Other schema design methodologies include the top-down strategy, which begins with a set of high-level entity types that are gradually specialized to subclass entities. The bottom-up strategy takes the opposite approach and begins with simpler entities or at least a set of attributes and systematically builds more robust entities. Development of the AARDB conceptual model actually entailed a mixture of the above strategies as several iterations of the data requirement and conceptualization steps were fulfilled. The primary strategy was to build from the set of centralized entities (i.e., Location, Sample and Sub-sample), with specialization and generalization techniques to augment the process.

Some aspects of AARDB's conceptual schema should be particularly noted. For instance, a rank view of entity relationships between **Location**, **SampleCollection**, **Sample**, **SubSample** and **Chromatogram** does not adequately describe all aspects of the database. These entities can have multiple relationships, such as with **Location** specifying spatial coordinates for a sample collection (denoting a sampling event or a group of samples) in **SampleCollection** or for an individual mollusk in **Sample** (Figure A-2). Likewise, **Sample** has relationships with most entities in AARDB (Figure A-2), suggesting the significance of a sample in AARDB. This reflects the database requirement that parameters stored in AARDB should be relatable to an individual sample so that analyses can be coupled or compared.

The database requirement that AAR statistics be calculated for all chromatographic analyses is conceptualized in the model as a relationship class with attributes Mean, Std. Dev. and Count (AAR Statistics in Figure A-2). This relationship is an example of a relationship entity that contains attributes, signifying that the relationship could be mapped out as a table. In AARDB, however, this relationship entity is actually a database view (a saved query) because the values of the table can be calculated on the fly and need not be stored in the database.

Other aspects of interest in the AARDB schema are instances of specialization. Specialization is the process of defining subclasses of an entity type, referred to as a superclass (Elmasri and Navathe, 2000). There are four instances of specialization of a superclass in AARDB, **TaphonomicCharacter**, **SampleImage**, **Result**, and **Procedure** (Figure A-2). Specialization of **Result** designates the kind of finalized analyses that can be stored, radiometric, isotopic or chromatographic. It is designated as a weak entity type

because it borrows SubsampleID from the entity **SubSample**; SubSampleID and Parameter make up the composite key for the relation **Result** (Figure A-2).

The entity type **TaphonomicCharacter** exhibits specialization in the form of taphonomic designations for a group of shells or individual shells. Individual samples are described by the attributes Characteristic and Value, while sample collections are described by the composite attribute of Characteristic and Value along with tallies (attribute Totals) for a sample collection (Figure A-2). This distinction was made because of separate plans for categorizing the taphonomy of sample shells. Earlier efforts at UDAL assigned characteristic totals to whole sample collections; the number of shells in a collection with a certain abrasion type, shell color, and so on. Later, it was decided that the extra effort early on to assign individual shells with particular taphonomic characteristics was more useful as this detailed data could then be translated to collection totals. This association forms the basis of the UNION query performed in chapter 6 of this report.

A-4 Choosing a DBMS

Considerations for choosing a DBMS were discussed in the last section of Principles of Database Development (chapter 4). In addition, factors influencing the move from a flat-file format database to relational database were discussed in section 4.2 of this work.

Often database development tools are incorporated into several DBMS software to aid the developer when moving from conceptualizing to actualizing the database. The existence of these tools, or more generally, the user-friendly nature of the DBMS, may be a good incentive for choosing one system over another.

A-5 Database Mapping and Physical Design

Mapping out the database tables from the conceptual model can often be performed utilizing tools available from the DBMS software. However, this step need not be tailored to a specific DBMS. Creation of tables and their relationships could simply be in the form of DDL (data definition language) statements through SQL or other such languages (Elmasri and Navathe, 2000). With MS Access (version 2000), there are no development tools for creating a conceptual model and, therefore, no way to translate conceptual entities into database tables. Tables must be created either using SQL or through a graphical interface that can be facilitated by software aids (Access wizards).

Construction of a table schema should be worked out using the steps of normalization described in chapter 4 before physical creation of these tables in the DBMS. The normalization process is useful for coming up with well-designed relation schemas (i.e., table designs) and an ER-to-Relational Mapping algorithm, such as presented in Elmasri and Navathe (2000), can help the developer create the physical database tables. Only the steps of Elmasri and Navathe (2000) pertinent to the development of AARDB are presented here.

A-5.1 ER-to-Relational Mapping

Step 1

For every strong entity (as opposed to a weak entity), create a table that includes all simple attributes of the entity. Composite attributes should be decomposed and stored as atomic attributes. One the attributes should be chosen as the primary key (following rules of first and second normal form).

For example, the entity **Location** formed the table *tblLocation* with its unique identifier, UDAMS number, as the primary key and LocalityID, Lat, Long and SurfaceElevation_m included as non-key attributes. A general note or remark attribute is also included for recording notes pertaining to a sampling site.

Step 2

For every weak entity, create a table that includes all simple attributes (as in Step 1 above). Include the primary key attributes of the owner entity as foreign keys in the new table. The primary key for the weak entity will include the foreign keys (primary keys of owner entity) and any partial keys.

The table *tblDLRatio* in AARDB was created from the weak entity **DLRatio** using the attributes Date, RefNumber, AnalysisMethod, AminoAcid and AreaOrHeight as the composite (primary) key. Date, RefNumber and AnalysisMethod are primary keys of the owner entity **Chromatogram**. Attributes AminoAcid and AreaOrHeight are partial keys because they clarify the meaning of table *tblDLRatio* and are necessary for unique records; establishing these two attributes as the composite key without including the foreign keys would create non-unique records and so would not be a suitable primary key.

Step 3

Mapping of Relationships. For a 1:1 relationship between two entities, include the primary key of one entity as a foreign key in the other entity. Generally, the entity that contains the foreign key will have a total participation in the relationship. If both entities have total participation in the relationship than one table, as opposed to two, can be created containing the attributes of both entities and the relationship.

A 1:1 relationship exists between relations **SamplingLog** and **Location**. Not every **Location** record will have corresponding log information but all **SamplingLog** records correspond with one and only one **Location** record. This means that **SamplingLog** exhibits total participation in the **Has** relationship with **Location** (see Figure A-2). The table *tblSamplingLog* (see Chapter 5) is created with all its atomic attributes along with the primary key of **Location** (UDAMS) as a foreign key. **SamplingLog** is also a weak entity so the foreign key (UDAMS) also represents *tblSamplingLog*'s primary key.

For a 1:N relationship between two entities, determine which entity represents the parent entity (1 side) and which represents the child entity (N side). The child entity becomes a table with the primary key of the parent entity as a foreign key.

There are several examples of 1:N relationships in the conceptual model of AARDB. Each of these entity relationships corresponds to parent/child tables in AARDB that exhibit the parent's primary key as the foreign key in the child table.

For a N:M relationship between two entities, a new table needs to be created that represents the relationship between the N:M entities. This new table contains the foreign keys from both participating entities. The combination of the foreign keys makes up the primary key of the new table. Any pertinent simple attributes should also be included in this new table. No examples of N:M relationships exist in AARDB but it is a common issue in database design.

The guidelines above for considering entity and relationship types of differing cardinality ratios were all for binary relationships, that is, for relationships between two entities. Binary relationships represent the simplest scenario for database design and n-

ary relationships can often be expressed as several binary relationships (Elmasri and Navathe, 2000). When several binary relationships does not fully explain the semantics of the database (as described by the database requirements stage or stage 1) than other steps need to be taken to create tables that satisfy all possible data relationships in the database. Discussion on mapping out n-ary relationships can be found in Elmasri and Navathe (2000).

A-5.2 Specialization and Generalization

One final point on mapping out database tables from a conceptual model deals with handling specialization or generalization. The designer can either create a table for each subclass entity or have one table representing the superclass. The primary key of the superclass would be the primary key for each of the subclass tables. Generally, separate subclass tables are constructed when there are attributes that pertain to individual subclasses, such that in a superclass table an attribute would remain blank for every record pertinent to other subclasses.

In AARDB, the superclass entity **Procedure** was split into two tables, *tblSubSamplingProcedure* and *tblAnalysisProcedure*. This was considered appropriate because the sub-sampling method required distinctive information to be stored (e.g. total or free amino acid content). Likewise, the analysis procedure required a "units" attribute designating the kind of measurement made in the analysis.

The superclass entity **Result** is an example where it is unnecessary to subdivide the entity into subclass tables. This is because the attributes of **Result** appropriately describe each subclass. However, this may change in the future if more specific data is to be recorded in the database for each of the subclass entities. The conceptual model of

AARDB does allow for this, and would only entail adding the necessary attributes to the appropriate subclass entity of the model. The conceptual change would then translate into construction of subclass tables with the corresponding records.

A-5.3 Automating Database Editing

Further physical database construction includes steps such as specifying whether cascading updates and deletions are allowed for a particular table relationship. Cascading updates and deletions specify to the DBMS that when an attribute of a record is edited or deleted, that change should propagate throughout the database such that an edit need only be made once. Otherwise, the DBMS would not allow an edit to an attribute in a parent table with a corresponding attribute in a child table (i.e., foreign key) to occur. The edit must first be made to the child table (assuming that referential integrity constraints are not violated) and then to the parent table. Generally, it is more prudent to allow cascading updates but not cascading deletions. This is because a user may inadvertently delete information in a child table if a deletion is made in the parent table.

Another aspect of record updates propagating throughout the database uses the concept of a general lookup table for set record values. Often, values stored in a database need to be constrained to a specific set of values from which data entry personnel can select. Two benefits are evident from using a general-purpose lookup table that is linked to the database tables. The first is that referential integrity is enforced for non-key attributes because an entered value must correspond to values documented in the lookup table. For example, in AARDB, the table *tblSample* contains a non-key attribute that describes the genus of a particular mollusk (SampleType). Values entered for this attribute are restricted to a prescribed set of genera established in *tblLookupTable* (see

Chapter 5) as arranged by a one-to-many relationship between *tblLookupTable* and *tblSample*. New genera values would be added to *tblLookupTable* before they could be entered in *tblSample*.

The second benefit of a general-purpose lookup table is that prescribed values can be edited once within the lookup table and that edit propagate to records within core database tables via a one-to-many relationship. For instance, there are eight amino acids presently stored in AARDB. These amino acid choices are recorded once in *tblLookupTable* and are accessed by the table *tblDLRatio*. The amino acid ratio of alloIsoleucine and Isoleucine is recorded in the database as "aIle/Ile". However, often software that imports data from the database does not accept special characters for attribute names, such as "/". Even though numerous records in the database contain the amino acid ratio item in the "aIle/Ile" form, a change dropping the "/" would only need to take place in *tblLookupTable* once; all records containing the old amino acid name would be changed to the new name in order to maintain referential integrity.

A-6 Implementation and Fine Tuning

The final step in the relational database design process involves implementing the database with actual data and testing different transactions. This involves populating the database tables and testing different query scenarios. The designer will mainly want to assess querying time efficiency and whether queries fully deliver the requests of the database users.

Symptoms of poor database design include unnecessarily long query execution times and generation of spurious results. Generally, the solution is to go back to the conceptual design and mapping stage to make certain that tables are normalized properly.

Often, poor querying time efficiencies are resolved with good use of table indices (indexes) (Elmasri and Navathe, 2000). An index should be established for all primary and foreign keys, and then for attributes that are accessed often.

While this topic deserves further development, such an effort is outside the scope of this work. Additional guidelines for database fine-tuning can be found in Elmasri and Navathe (2000) or other similar texts.

APPENDIX II

DISCRIMINANT ANALYSIS SUMMARY AND OUTPUT

Discriminant analysis was performed on AAR from GC analysis for sites within northeastern North Carolina. The results of the analysis are discriminant functions used to calculate raw discriminant scores for a particular sample. Several discriminant functions were determined from the analysis based on the retention or discarding of variables (amino acids). All calibration analyses were 100% accurate in differentiation of groups (Holocene or Pleistocene).

Test samples were also analyzed for each discriminant function with varying results. The discriminant functions that performed the best were those that discarded Aspartic Acid. The function containing Alanine (Ala), Glutamic Acid (Glu), Leucine (Leu), Phenylalanine (Phe) and Valine (Val) discriminated correctly 100% of the time (out of 13 observations). The function containing Ala, Glu, Leu and Phe discriminated correctly 95% of the time (out of 19 observations).

The discriminant analysis output is given below, followed by statistical tests of significance for Mahalanobis' distances and variables. These tests were used to determine which variables should be retained and which could be discarded.

MINITAB Output

Discriminant Analysis: Holo/Pleist versus Ala, Asp, Glu, Leu, Phe, Val

```
Linear Method for Response:   Holo/Ple
Predictors:  Ala  Asp  Glu  Leu  Phe  Val

Group     Holo   Pleist
Count       20        4

Summary of Classification

Put into     ....True Group....
Group         Holo   Pleist
Holo            20        0
Pleist           0        4
Total N         20        4
N Correct       20        4
Proportion   1.000    1.000

N =   24     N Correct =   24     Proportion Correct = 1.000

Squared Distance Between Groups
             Holo   Pleist
Holo       0.0000  26.9421
Pleist    26.9421   0.0000

Linear Discriminant Function for Group
             Holo   Pleist
Constant   -87.03   -92.89
Ala        -68.18   185.31
Asp        567.44   439.48
Glu       -246.94  -219.40
Leu        -50.80   108.09
Phe        -32.36  -262.87
Val        -87.72  -193.59

Variable   Pooled    Means for Group
             Mean     Holo   Pleist
Ala       0.37417  0.33950  0.54750
Asp       0.49250  0.48300  0.54000
Glu       0.20000  0.19100  0.24500
Leu       0.21542  0.20200  0.28250
Phe       0.25875  0.23900  0.35750
Val       0.14208  0.13350  0.18500

Variable   Pooled    StDev for Group
             StDev     Holo   Pleist
Ala       0.06821  0.06573  0.08221
Asp       0.04662  0.04131  0.07165
Glu       0.03182  0.02864  0.04726
Leu       0.04348  0.03968  0.06238
Phe       0.05537  0.05230  0.07182
Val       0.03086  0.02720  0.04796
```

Pooled Covariance Matrix

	Ala	Asp	Glu	Leu	Phe	Val
Ala	0.0046532					
Asp	0.0026105	0.0021736				
Glu	0.0018073	0.0013155	0.0010127			
Leu	0.0020293	0.0015218	0.0010959	0.0018907		
Phe	0.0035393	0.0020845	0.0014895	0.0019893	0.0030661	
Val	0.0018311	0.0011723	0.0008923	0.0010505	0.0015327	0.0009525

Covariance Matrix for Group Holo

	Ala	Asp	Glu	Leu	Phe	Val
Ala	0.0043208					
Asp	0.0023437	0.0017063				
Glu	0.0015532	0.0010179	0.0008200			
Leu	0.0021695	0.0012095	0.0010189	0.0015747		
Phe	0.0032574	0.0016663	0.0012274	0.0018653	0.0027358	
Val	0.0015334	0.0008784	0.0006805	0.0010242	0.0012721	0.0007397

Covariance Matrix for Group Pleist

	Ala	Asp	Glu	Leu	Phe	Val
Ala	0.0067583					
Asp	0.0043000	0.0051333				
Glu	0.0034167	0.0032000	0.0022333			
Leu	0.0011417	0.0035000	0.0015833	0.0038917		
Phe	0.0053250	0.0047333	0.0031500	0.0027750	0.0051583	
Val	0.0037167	0.0030333	0.0022333	0.0012167	0.0031833	0.0023000

Summary of Classified Observations

Observation	True Group	Pred Group	Group	Squared Distance	Probability
1	Holo	Holo	Holo	4.953	1.000
			Pleist	38.667	0.000
2	Holo	Holo	Holo	6.801	1.000
			Pleist	45.571	0.000
3	Holo	Holo	Holo	15.70	1.000
			Pleist	33.41	0.000
4	Holo	Holo	Holo	1.437	1.000
			Pleist	33.165	0.000
5	Holo	Holo	Holo	3.710	1.000
			Pleist	23.121	0.000
6	Holo	Holo	Holo	1.356	1.000
			Pleist	38.924	0.000
7	Holo	Holo	Holo	3.419	1.000
			Pleist	29.186	0.000
8	Holo	Holo	Holo	4.065	1.000
			Pleist	50.814	0.000
9	Holo	Holo	Holo	2.895	1.000
			Pleist	40.081	0.000
10	Holo	Holo	Holo	1.336	1.000
			Pleist	25.669	0.000
11	Holo	Holo	Holo	0.8555	1.000
			Pleist	29.8764	0.000
12	Holo	Holo	Holo	6.324	1.000
			Pleist	50.611	0.000
13	Holo	Holo	Holo	4.222	0.999
			Pleist	18.948	0.001
14	Holo	Holo	Holo	6.887	0.999
			Pleist	21.211	0.001
15	Holo	Holo	Holo	1.473	1.000
			Pleist	27.954	0.000
16	Holo	Holo	Holo	2.954	1.000
			Pleist	19.284	0.000

17	Holo	Holo	Holo	9.445	0.993
			Pleist	19.293	0.007
18	Holo	Holo	Holo	1.012	1.000
			Pleist	26.255	0.000
19	Holo	Holo	Holo	3.387	1.000
			Pleist	21.788	0.000
20	Holo	Holo	Holo	14.15	1.000
			Pleist	41.40	0.000
21	Pleist	Pleist	Holo	46.631	0.000
			Pleist	8.716	1.000
22	Pleist	Pleist	Holo	42.74	0.000
			Pleist	14.72	1.000
23	Pleist	Pleist	Holo	21.434	0.001
			Pleist	8.037	0.999
24	Pleist	Pleist	Holo	32.583	0.000
			Pleist	4.146	1.000

Prediction for Test Observations

SubSampleID	Pred Group	From Group	Sqrd Distnc	Probability
990162	Holo			
		Holo	37.213	1.000
		Pleist	79.870	0.000
2003168	Holo			
		Holo	18.519	1.000
		Pleist	80.097	0.000
960282	Holo			
		Holo	12.263	1.000
		Pleist	38.894	0.000
960180	Holo			
		Holo	14.752	1.000
		Pleist	55.819	0.000
2000214	Holo			
		Holo	5.103	1.000
		Pleist	42.251	0.000
2001012	Holo			
		Holo	6.070	1.000
		Pleist	50.639	0.000
* 960388 *	Holo			
		Holo	32.268	0.998
		Pleist	44.579	0.002
2000021	Pleist			
		Holo	41.419	0.000
		Pleist	13.642	1.000
950218	Pleist			
		Holo	39.938	0.000
		Pleist	13.696	1.000
2000108	Pleist			
		Holo	58.309	0.000
		Pleist	12.356	1.000
2000113	Pleist			
		Holo	42.499	0.000
		Pleist	22.675	1.000
2000114	Pleist			
		Holo	45.392	0.000
		Pleist	12.483	1.000
* 2000035 *	Holo			
		Holo	49.518	0.999
		Pleist	62.889	0.001

*SubSamples surrounded by * * are misclassified*

2 misclassified out of 13 sub-samples

85% success rate

Discriminant Analysis: Holo/Pleist versus Ala, Glu, Leu, Phe, Val

Linear Method for Response: Holo/Ple
Predictors: Ala Glu Leu Phe Val

Group	Holo	Pleist
Count	20	4

Summary of Classification

Put into	....True Group....	
Group	Holo	Pleist
Holo	20	0
Pleist	0	4
Total N	20	4
N Correct	20	4
Proportion	1.000	1.000

N = 24 N Correct = 24 Proportion Correct = 1.000

Squared Distance Between Groups

	Holo	Pleist
Holo	0.0000	20.6652
Pleist	20.6652	0.0000

Linear Discriminant Function for Group

	Holo	Pleist
Constant	-25.31	-55.86
Ala	138.74	345.57
Glu	290.49	196.85
Leu	98.81	223.97
Phe	-172.09	-371.10
Val	-230.74	-304.36

Variable	Pooled Mean	Means for Group Holo	Pleist
Ala	0.37417	0.33950	0.54750
Glu	0.20000	0.19100	0.24500
Leu	0.21542	0.20200	0.28250
Phe	0.25875	0.23900	0.35750
Val	0.14208	0.13350	0.18500

Variable	Pooled StDev	StDev for Group Holo	Pleist
Ala	0.06821	0.06573	0.08221
Glu	0.03182	0.02864	0.04726
Leu	0.04348	0.03968	0.06238
Phe	0.05537	0.05230	0.07182
Val	0.03086	0.02720	0.04796

Pooled Covariance Matrix

	Ala	Glu	Leu	Phe	Val
Ala	0.0046532				
Glu	0.0018073	0.0010127			
Leu	0.0020293	0.0010959	0.0018907		
Phe	0.0035393	0.0014895	0.0019893	0.0030661	
Val	0.0018311	0.0008923	0.0010505	0.0015327	0.0009525

Covariance Matrix for Group Holo

	Ala	Glu	Leu	Phe	Val
Ala	0.0043208				
Glu	0.0015532	0.0008200			

Leu	0.0021695	0.0010189	0.0015747		
Phe	0.0032574	0.0012274	0.0018653	0.0027358	
Val	0.0015334	0.0006805	0.0010242	0.0012721	0.0007397

Covariance Matrix for Group Pleist

	Ala	Glu	Leu	Phe	Val
Ala	0.0067583				
Glu	0.0034167	0.0022333			
Leu	0.0011417	0.0015833	0.0038917		
Phe	0.0053250	0.0031500	0.0027750	0.0051583	
Val	0.0037167	0.0022333	0.0012167	0.0031833	0.0023000

Summary of Classified Observations

Observation	True Group	Pred Group	Group	Squared Distance	Probability
1	Holo	Holo	Holo	3.219	1.000
			Pleist	37.254	0.000
2	Holo	Holo	Holo	4.998	1.000
			Pleist	44.219	0.000
3	Holo	Holo	Holo	14.68	1.000
			Pleist	31.17	0.000
4	Holo	Holo	Holo	1.428	1.000
			Pleist	27.346	0.000
5	Holo	Holo	Holo	3.396	1.000
			Pleist	19.339	0.000
6	Holo	Holo	Holo	0.5947	1.000
			Pleist	27.5160	0.000
7	Holo	Holo	Holo	3.334	1.000
			Pleist	24.284	0.000
8	Holo	Holo	Holo	3.061	1.000
			Pleist	38.513	0.000
9	Holo	Holo	Holo	0.5010	1.000
			Pleist	23.6584	0.000
10	Holo	Holo	Holo	0.6847	1.000
			Pleist	22.7836	0.000
11	Holo	Holo	Holo	0.8526	1.000
			Pleist	23.3242	0.000
12	Holo	Holo	Holo	1.651	1.000
			Pleist	28.830	0.000
13	Holo	Holo	Holo	3.661	0.998
			Pleist	15.862	0.002
14	Holo	Holo	Holo	6.341	0.898
			Pleist	10.683	0.102
15	Holo	Holo	Holo	1.244	1.000
			Pleist	19.050	0.000
16	Holo	Holo	Holo	2.714	0.998
			Pleist	15.224	0.002
17	Holo	Holo	Holo	9.445	0.861
			Pleist	13.093	0.139
18	Holo	Holo	Holo	1.012	1.000
			Pleist	20.106	0.000
19	Holo	Holo	Holo	3.046	0.990
			Pleist	12.243	0.010
20	Holo	Holo	Holo	13.61	1.000
			Pleist	38.28	0.000
21	Pleist	Pleist	Holo	32.088	0.000
			Pleist	7.005	1.000
22	Pleist	Pleist	Holo	32.41	0.000
			Pleist	14.22	1.000
23	Pleist	Pleist	Holo	20.714	0.000
			Pleist	5.291	1.000
24	Pleist	Pleist	Holo	27.977	0.000

```
                              Pleist       4.017         1.000

Prediction for Test Observations

 SubSampleID   Pred Group  From Group Sqrd Distnc Probability
      990162        Holo
                                Holo      21.686       1.000
                              Pleist      77.811       0.000
     2003168        Holo
                                Holo      18.011       1.000
                              Pleist      69.741       0.000
      960282        Holo
                                Holo       7.948       1.000
                              Pleist      38.710       0.000
      960180        Holo
                                Holo      10.389       1.000
                              Pleist      55.646       0.000
     2000214        Holo
                                Holo       4.880       1.000
                              Pleist      38.118       0.000
     2001012        Holo
                                Holo       6.040       1.000
                              Pleist      45.205       0.000
      960388      Pleist
                                Holo      30.779       0.490
                              Pleist      30.702       0.510
     2000021      Pleist
                                Holo      34.565       0.000
                              Pleist      13.630       1.000
      950218      Pleist
                                Holo      34.465       0.000
                              Pleist      13.668       1.000
     2000108      Pleist
                                Holo      51.564       0.000
                              Pleist      12.347       1.000
     2000113      Pleist
                                Holo      42.489       0.000
                              Pleist      16.884       1.000
     2000114      Pleist
                                Holo      39.252       0.000
                              Pleist      12.482       1.000
     2000035      Pleist
                                Holo      45.748       0.211
                              Pleist      43.114       0.789
```

0 misclassified out of 13 sub-samples
100% success rate

Discriminant Analysis: Holo/Pleist versus Ala, Asp, Glu, Leu, Phe

```
Linear Method for Response:   Holo/Ple
Predictors:  Ala  Asp  Glu  Leu  Phe

Group     Holo   Pleist
Count       20        4

Summary of Classification

Put into     ....True Group....
Group         Holo   Pleist
Holo            20        0
```

```
Pleist           0        4
Total N         20        4
N Correct       20        4
Proportion   1.000    1.000

N =   24     N Correct =   24     Proportion Correct = 1.000

Squared Distance Between Groups
             Holo   Pleist
Holo       0.0000  25.7275
Pleist    25.7275   0.0000

Linear Discriminant Function for Group
             Holo   Pleist
Constant   -86.61   -90.85
Ala        -73.10   174.46
Asp        573.69   453.27
Glu       -296.11  -327.91
Leu        -52.33   104.71
Phe        -49.90  -301.58

Variable   Pooled    Means for Group
             Mean     Holo   Pleist
Ala       0.37417  0.33950  0.54750
Asp       0.49250  0.48300  0.54000
Glu       0.20000  0.19100  0.24500
Leu       0.21542  0.20200  0.28250
Phe       0.25875  0.23900  0.35750

Variable   Pooled    StDev for Group
            StDev     Holo   Pleist
Ala       0.06821  0.06573  0.08221
Asp       0.04662  0.04131  0.07165
Glu       0.03182  0.02864  0.04726
Leu       0.04348  0.03968  0.06238
Phe       0.05537  0.05230  0.07182

Pooled Covariance Matrix
               Ala       Asp       Glu       Leu       Phe
Ala       0.004653
Asp       0.002610  0.002174
Glu       0.001807  0.001315  0.001013
Leu       0.002029  0.001522  0.001096  0.001891
Phe       0.003539  0.002085  0.001490  0.001989  0.003066

Covariance Matrix for Group Holo
               Ala        Asp        Glu        Leu        Phe
Ala       0.0043208
Asp       0.0023437  0.0017063
Glu       0.0015532  0.0010179  0.0008200
Leu       0.0021695  0.0012095  0.0010189  0.0015747
Phe       0.0032574  0.0016663  0.0012274  0.0018653  0.0027358

Covariance Matrix for Group Pleist
               Ala        Asp        Glu        Leu        Phe
Ala       0.0067583
Asp       0.0043000  0.0051333
Glu       0.0034167  0.0032000  0.0022333
Leu       0.0011417  0.0035000  0.0015833  0.0038917
Phe       0.0053250  0.0047333  0.0031500  0.0027750  0.0051583
```

Summary of Classified Observations

Observation	True Group	Pred Group	Group	Squared Distance	Probability
1	Holo	Holo	Holo	4.567	1.000
			Pleist	38.436	0.000
2	Holo	Holo	Holo	6.697	1.000
			Pleist	44.962	0.000
3	Holo	Holo	Holo	8.824	0.995
			Pleist	19.540	0.005
4	Holo	Holo	Holo	1.282	1.000
			Pleist	30.926	0.000
5	Holo	Holo	Holo	2.453	1.000
			Pleist	23.121	0.000
6	Holo	Holo	Holo	1.349	1.000
			Pleist	37.523	0.000
7	Holo	Holo	Holo	2.236	1.000
			Pleist	29.185	0.000
8	Holo	Holo	Holo	3.305	1.000
			Pleist	46.918	0.000
9	Holo	Holo	Holo	2.885	1.000
			Pleist	39.076	0.000
10	Holo	Holo	Holo	1.171	1.000
			Pleist	25.184	0.000
11	Holo	Holo	Holo	0.5719	1.000
			Pleist	29.5520	0.000
12	Holo	Holo	Holo	5.384	1.000
			Pleist	46.320	0.000
13	Holo	Holo	Holo	3.762	0.999
			Pleist	18.768	0.001
14	Holo	Holo	Holo	6.714	0.998
			Pleist	18.905	0.002
15	Holo	Holo	Holo	1.223	1.000
			Pleist	27.592	0.000
16	Holo	Holo	Holo	1.987	0.998
			Pleist	14.935	0.002
17	Holo	Holo	Holo	5.068	0.999
			Pleist	18.313	0.001
18	Holo	Holo	Holo	0.6023	1.000
			Pleist	23.2182	0.000
19	Holo	Holo	Holo	3.264	1.000
			Pleist	19.677	0.000
20	Holo	Holo	Holo	14.14	1.000
			Pleist	39.88	0.000
21	Pleist	Pleist	Holo	43.456	0.000
			Pleist	8.254	1.000
22	Pleist	Pleist	Holo	42.69	0.000
			Pleist	12.95	1.000
23	Pleist	Pleist	Holo	20.699	0.002
			Pleist	7.977	0.998
24	Pleist	Pleist	Holo	28.586	0.000
			Pleist	3.341	1.000

Prediction for Test Observations

Observation	Pred Group	From Group	Sqrd Distnc	Probability
990162	Holo			
		Holo	36.572	1.000
		Pleist	79.779	0.000
2003168	Holo			
		Holo	17.672	1.000
		Pleist	80.064	0.000
960282	Holo			

```
                              Holo      11.304       1.000
                             Pleist      38.878       0.000
     960180        Holo
                              Holo      12.878       1.000
                             Pleist      55.748       0.000
    2000214        Holo
                              Holo       5.034       1.000
                             Pleist      41.547       0.000
    2001012        Holo
                              Holo       6.065       1.000
                             Pleist      49.256       0.000
   * 960388 *      Holo
                              Holo      22.565       0.892
                             Pleist      26.797       0.108
    2000021      Pleist
                              Holo      40.420       0.000
                             Pleist       9.226       1.000
     950218      Pleist
                              Holo      39.934       0.000
                             Pleist      12.355       1.000
    2000108      Pleist
                              Holo      57.204       0.000
                             Pleist       7.719       1.000
    2000113      Pleist
                              Holo      40.938       0.000
                             Pleist      22.654       1.000
    2000114      Pleist
                              Holo      45.300       0.000
                             Pleist      11.846       1.000
  * 2000035 *      Holo
                              Holo      48.339       0.992
                             Pleist      58.103       0.008
    2001002        Holo
                              Holo       8.293       1.000
                             Pleist      31.286       0.000
    2000190        Holo
                              Holo      51.428       1.000
                             Pleist     102.861       0.000
     960217        Holo
                              Holo       7.345       1.000
                             Pleist      47.013       0.000
  * 2000210 *      Holo
                              Holo      11.594       0.974
                             Pleist      18.860       0.026
    2000145        Holo
                              Holo       5.010       1.000
                             Pleist      26.148       0.000
    2000205        Holo
                              Holo       1.053       1.000
                             Pleist      23.102       0.000
```

*SubSamples surrounded by * * are misclassified*
3 misclassified out of 13 sub-samples
77% success rate

Discriminant Analysis: Holo/Pleist versus Ala, Glu, Leu, Phe

```
Linear Method for Response:   Holo/Ple
Predictors:  Ala  Glu  Leu  Phe
```

```
Group     Holo   Pleist
Count       20        4

Summary of Classification

Put into     ....True Group....
Group         Holo   Pleist
Holo            20        0
Pleist           0        4
Total N         20        4
N Correct       20        4
Proportion   1.000    1.000

N =   24     N Correct =   24     Proportion Correct = 1.000

Squared Distance Between Groups
             Holo   Pleist
Holo       0.0000  20.0672
Pleist    20.0672   0.0000

Linear Discriminant Function for Group
             Holo   Pleist
Constant   -22.37   -50.75
Ala        131.66   336.24
Glu        174.65    44.04
Leu         99.13   224.38
Phe       -223.19  -438.50

Variable   Pooled    Means for Group
             Mean     Holo   Pleist
Ala       0.37417  0.33950  0.54750
Glu       0.20000  0.19100  0.24500
Leu       0.21542  0.20200  0.28250
Phe       0.25875  0.23900  0.35750

Variable   Pooled    StDev for Group
            StDev     Holo   Pleist
Ala       0.06821  0.06573  0.08221
Glu       0.03182  0.02864  0.04726
Leu       0.04348  0.03968  0.06238
Phe       0.05537  0.05230  0.07182

Pooled Covariance Matrix
               Ala       Glu       Leu       Phe
Ala       0.004653
Glu       0.001807  0.001013
Leu       0.002029  0.001096  0.001891
Phe       0.003539  0.001490  0.001989  0.003066

Covariance Matrix for Group Holo
               Ala        Glu        Leu        Phe
Ala       0.0043208
Glu       0.0015532  0.0008200
Leu       0.0021695  0.0010189  0.0015747
Phe       0.0032574  0.0012274  0.0018653  0.0027358


Covariance Matrix for Group Pleist
               Ala        Glu        Leu        Phe
Ala       0.0067583
Glu       0.0034167  0.0022333
Leu       0.0011417  0.0015833  0.0038917
Phe       0.0053250  0.0031500  0.0027750  0.0051583
```

Summary of Classified Observations

Observation	True Group	Pred Group	Group	Squared Distance	Probability
1	Holo	Holo	Holo	3.017	1.000
			Pleist	37.149	0.000
2	Holo	Holo	Holo	4.978	1.000
			Pleist	43.822	0.000
3	Holo	Holo	Holo	6.940	0.997
			Pleist	18.526	0.003
4	Holo	Holo	Holo	1.260	1.000
			Pleist	25.944	0.000
5	Holo	Holo	Holo	2.282	1.000
			Pleist	19.259	0.000
6	Holo	Holo	Holo	0.5935	1.000
			Pleist	26.9717	0.000
7	Holo	Holo	Holo	2.215	1.000
			Pleist	24.203	0.000
8	Holo	Holo	Holo	2.507	1.000
			Pleist	36.210	0.000
9	Holo	Holo	Holo	0.4052	1.000
			Pleist	23.4432	0.000
10	Holo	Holo	Holo	0.5947	1.000
			Pleist	22.5596	0.000
11	Holo	Holo	Holo	0.5558	1.000
			Pleist	23.2719	0.000
12	Holo	Holo	Holo	1.181	1.000
			Pleist	26.700	0.000
13	Holo	Holo	Holo	3.321	0.998
			Pleist	15.826	0.002
14	Holo	Holo	Holo	6.238	0.835
			Pleist	9.487	0.165
15	Holo	Holo	Holo	0.9199	1.000
			Pleist	19.0079	0.000
16	Holo	Holo	Holo	1.593	0.994
			Pleist	11.868	0.006
17	Holo	Holo	Holo	4.996	0.959
			Pleist	11.309	0.041
18	Holo	Holo	Holo	0.5896	1.000
			Pleist	18.0810	0.000
19	Holo	Holo	Holo	2.970	0.983
			Pleist	11.144	0.017
20	Holo	Holo	Holo	13.55	1.000
			Pleist	37.27	0.000
21	Pleist	Pleist	Holo	30.444	0.000
			Pleist	6.745	1.000
22	Pleist	Pleist	Holo	31.97	0.000
			Pleist	12.15	1.000
23	Pleist	Pleist	Holo	20.152	0.001
			Pleist	5.291	0.999
24	Pleist	Pleist	Holo	24.994	0.000
			Pleist	3.107	1.000

Prediction for Test Observations

Observation	Pred Group	From Group	Sqrd Distnc	Probability
990162	Holo			
		Holo	21.611	1.000
		Pleist	77.562	0.000
2003168	Holo			
		Holo	16.960	1.000
		Pleist	69.678	0.000
960282	Holo			

		Holo	7.448	1.000
		Pleist	38.706	0.000
960180	Holo			
		Holo	9.181	1.000
		Pleist	55.540	0.000
2000214	Holo			
		Holo	4.840	1.000
		Pleist	37.791	0.000
2001012	Holo			
		Holo	6.030	1.000
		Pleist	44.445	0.000
960388	Pleist			
		Holo	21.909	0.067
		Pleist	16.627	0.933
2000021	Pleist			
		Holo	32.709	0.000
		Pleist	9.068	1.000
950218	Pleist			
		Holo	34.326	0.000
		Pleist	12.355	1.000
2000108	Pleist			
		Holo	49.570	0.000
		Pleist	7.571	1.000
2000113	Pleist			
		Holo	40.933	0.000
		Pleist	16.660	1.000
2000114	Pleist			
		Holo	39.251	0.000
		Pleist	11.839	1.000
2000035	Pleist			
		Holo	45.055	0.094
		Pleist	40.534	0.906
2001002	Holo			
		Holo	3.016	1.000
		Pleist	31.279	0.000
2000190	Holo			
		Holo	32.197	1.000
		Pleist	98.836	0.000
960217	Holo			
		Holo	7.289	1.000
		Pleist	42.424	0.000
* 2000210 *	Holo			
		Holo	4.469	0.999
		Pleist	18.775	0.001
2000145	Holo			
		Holo	3.675	1.000
		Pleist	24.650	0.000
2000205	Holo			
		Holo	1.011	1.000
		Pleist	16.427	0.000

*SubSamples surrounded by * * are misclassified*
1 misclassified out of 19 sub-samples
95% success rate

Discriminant Analysis Dataset

Samples with D/L Leucine range

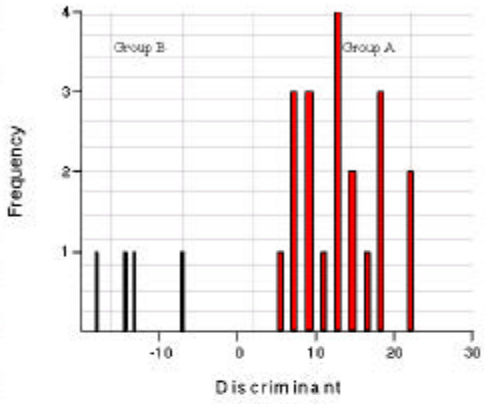
between 0.15 and 0.36

ObservID	UDAMS	SubSampleID	SampleType	Prefix	C14ID	Value	Group	Ala	Asp	Glu	Leu	Phe	Val
1	7246	2000192	Mercenaria	EqualTo	Beta62755	2610	Holo	0.257	0.429	0.176	0.179	0.192	0.111
2	7182	2000200	Mercenaria	EqualTo	CAMS#88133	3620	Holo	0.208	0.389	0.148	0.156	0.155	0.094
3	7228	960390	Mercenaria	EqualTo	Beta65463	4300	Holo	0.370	0.514	0.229	0.284	0.271	0.191
4	7270	990029	Mercenaria	EqualTo	AA36918	4350	Holo	0.306	0.450	0.174	0.165	0.210	0.115
5	7246	2000191	Mercenaria	EqualTo	Beta62754	5150	Holo	0.357	0.507	0.220	0.226	0.246	0.144
6	7439	2001035	Mercenaria	EqualTo	CAMS#87997	5370	Holo	0.296	0.478	0.182	0.181	0.208	0.116
7	7246	2000193	Mercenaria	EqualTo	Beta62756	5510	Holo	0.359	0.515	0.219	0.208	0.247	0.137
8	7266	2000018	Mercenaria	EqualTo	AA36921	5693	Holo	0.246	0.451	0.159	0.165	0.181	0.111
9	7439	2001036	Mercenaria	EqualTo	CAMS#87998	5785	Holo	0.337	0.506	0.186	0.193	0.244	0.130
10	7180	2000157	Mercenaria	EqualTo	AA40279	5938	Holo	0.298	0.444	0.170	0.168	0.198	0.112
11	7182	2000126	Mercenaria	EqualTo	CAMS#88135	5965	Holo	0.331	0.470	0.184	0.175	0.226	0.117
12	7182	2001001	Mercenaria	EqualTo	CAMS#88277	6160	Holo	0.268	0.482	0.163	0.155	0.179	0.111
13	7488	2002066	Mercenaria	EqualTo	OS-38553	6230	Holo	0.401	0.482	0.193	0.219	0.295	0.137
14	7327	960218	Mercenaria	EqualTo	Beta94231	7160	Holo	0.380	0.494	0.174	0.204	0.251	0.134
15	7383	2000162	Mercenaria	EqualTo	AA40281	7290	Holo	0.318	0.475	0.169	0.172	0.214	0.114
16	7239	2000039	Mercenaria	EqualTo	AA12730	7345	Holo	0.380	0.484	0.195	0.207	0.256	0.152
17	7280	2001075	Mercenaria	EqualTo	AA12726	7435	Holo	0.451	0.569	0.246	0.257	0.312	0.159
18	7266	2000062	Mercenaria	EqualTo	AA36920	7485	Holo	0.323	0.466	0.177	0.197	0.215	0.134
19	7271	990033	Mercenaria	EqualTo	AA36919	7575	Holo	0.425	0.551	0.234	0.230	0.288	0.171
20	7488	2002065	Mercenaria	EqualTo	OS-38552	8180	Holo	0.448	0.524	0.229	0.291	0.379	0.187
21	7086	2000115	Mercenaria	EqualTo	AA39341	34150	Pleist	0.442	0.437	0.182	0.217	0.251	0.118
22	7118	2000211	Mercenaria	GreaterThan	AA7322	39700	Pleist	0.639	0.558	0.283	0.242	0.402	0.226
23	7167	2000036	Mercenaria	EqualTo	AA37920	39800	Pleist	0.549	0.611	0.278	0.353	0.389	0.215
24	7534	2003154	Mercenaria	GreaterThan	OS-41113	52000	Pleist	0.564	0.550	0.239	0.316	0.386	0.183
25	7422	990160	Mercenaria	EqualTo	AA40284	30040	Pleist	0.43	0.55	0.22	0.24	0.32	0.16

Observation 25 (ObservID 25) was removed before determining discriminant function because it was originally misclassified

Results of Discriminant Analysis for two groups (Holocene and Pleistocene) performed in PAST statistical software

ObservID	Score	Group	Classification
1	16.866	A	A
2	19.394	A	A
3	8.8634	A	A
4	15.873	A	A
5	9.7145	A	A
6	18.793	A	A
7	12.892	A	A
8	23.384	A	A
9	18.602	A	A
10	12.176	A	A
11	14.52	A	A
12	22.153	A	A
13	7.3719	A	A
14	7.1705	A	A
15	13.25	A	A
16	8.1739	A	A
17	4.9329	A	A
18	12.63	A	A
19	9.2096	A	A
20	13.632	A	A
21	-18.949	B	B
22	-14.001	B	B
23	-6.6895	B	B
24	-14.21	B	B



Group A = Holocene
Group B = Pleistocene

Discriminant

Function $D^2 = (-253.49)(Ala) + (127.96)(Asp) + (-27.543)(Glu) + (-158.89)(Leu) + (230.52)(Phe) + (105.87)(Val)$

Observations

ObID	Ala	Asp	Glu	Leu	Phe	Val	
1	0.257	0.429	0.176	0.179	0.192	0.111	
2	0.208	0.389	0.148	0.156	0.155	0.094	
3	0.370	0.514	0.229	0.284	0.271	0.191	
4	0.306	0.450	0.174	0.165	0.210	0.115	
5	0.357	0.507	0.220	0.226	0.246	0.144	
6	0.296	0.478	0.182	0.181	0.208	0.116	
7	0.359	0.515	0.219	0.209	0.247	0.137	
8	0.246	0.451	0.159	0.165	0.181	0.111	
9	0.337	0.506	0.186	0.193	0.244	0.130	Group A
10	0.298	0.444	0.170	0.168	0.198	0.112	
11	0.331	0.470	0.184	0.175	0.226	0.117	
12	0.268	0.482	0.163	0.155	0.179	0.111	
13	0.401	0.482	0.193	0.219	0.295	0.137	
14	0.380	0.494	0.174	0.204	0.251	0.134	
15	0.318	0.475	0.169	0.172	0.214	0.114	
16	0.380	0.484	0.195	0.207	0.256	0.152	
17	0.451	0.569	0.246	0.257	0.312	0.159	
18	0.323	0.466	0.177	0.197	0.215	0.134	
19	0.425	0.551	0.234	0.230	0.288	0.171	
20	0.446	0.524	0.229	0.291	0.379	0.187	
21	0.442	0.437	0.182	0.217	0.251	0.118	
22	0.639	0.558	0.283	0.242	0.402	0.226	Group B
23	0.549	0.611	0.278	0.353	0.389	0.215	
24	0.564	0.550	0.239	0.316	0.386	0.183	

Group A Means

Ala	Asp	Glu	Leu	Phe	Val
0.3395	0.483	0.191	0.202	0.239	0.1335

R_a 7.616257

Group B Means

Ala	Asp	Glu	Leu	Phe	Val
0.5475	0.54	0.245	0.2825	0.3575	0.185

R_b -19.325

(Group A - Group B)

Ala	Asp	Glu	Leu	Phe	Val
-0.208	-0.057	-0.054	-0.0805	-0.1185	-0.0515

R_o -5.85436

D^2 26.94124

T^2 89.80414

F 11.56568

Critical value with degrees of freedom = 6 and 17 4.1

$F_{0.01,6,17}$ > Critical value so reject Null Hypothesis

Therefore, the distance between Group A and Group B is significant.

Tests of Significance

Ho = The multivariate means of Group A and Group B are equal
(i.e. the distance between the two groups is zero)
Ha = There is a significant distance between Group A and Group B

Hotelling's T^2 test (T^2)
$T^2 = (n_a n_b / n_a + n_b) * D^2$

F test statistic
$F = ((n_a + n_b - m - 1) / (n_a + n_b - 2)m) * T^2$
Degrees of freedom = m and $(n_a + n_b - m - 1)$

$D^2_m - D^2_{m-1}$ Test of significance between two calculated Mahalanobis' distance (D^2)
Null Hypothesis is NO difference between D^2_m and D^2_{m-1}
$F = (n_a+n_b-m-1)\ (D^2_m - D^2_{m-1}) / (((n_a+n_b-2)\ (n_a+n_b)/n_a n_b) + D^2_{m-1})$
Degrees of freedom = 1 and $n_a + n_b - m - 1$

$D_i = A_i - B_i / s_{pi}$ General guide for determining the discriminating power of the variables

Ala	Asp	Glu	Leu	Phe	Val
-2.00894	-1.12877	-1.4473137	-3.53573	-1.68124	-1.43085

Confidence interval of the F statistic is 0.01 with degrees of freedom
Tests of significance from Davis (1986) and Swan and Sandilands (1995)
Critical values of F distribution were determined from Swan and Sandilands (1995), Appendix 2.5

Discriminant Function Equations

$D^2 = \lambda_1 y_1 + \lambda_2 y_2 + \lambda_3 y_3 + \ldots + \lambda_i y_i$

$R_a = \lambda_1((\underline{A}_1 + \underline{B}_1)/2) + \lambda_2((\underline{A}_2 + \underline{B}_2)/2) + \ldots + \lambda_i((\underline{A}_i + \underline{B}_i)/2)$

$R_b = \lambda_1 \underline{A}_1 + \lambda_2 \underline{A}_2 + \ldots + \lambda_i \underline{A}_i$

$R_b = \lambda_1 \underline{B}_1 + \lambda_2 \underline{B}_2 + \ldots + \lambda_i \underline{B}_i$

Where D^2 is the Mahalanobis' distance
R_a is the centroid of Group A
R_b is the centroid of Group B
R_c is the discriminant index
λ represents coefficient of discriminant function
y represents value of a variable of the discriminant function
$\underline{A}_i$ is the mean of Group A for a variable
$\underline{B}_i$ is the mean of Group B for a variable
n_a is the number of observations for Group A
n_b is the number of observations for Group B
m is the number of variables
D_i is the standardized difference for variable i
s_{pi} is the pooled standard deviation of variable i

Remove Alanine
D^2 -23.7847
T^2 -85.9489
F -14.0644
$F_{0.01,5,38}$ = 4.25
F < Critical value so DO NOT reject Null Hypothesis
Can not discriminate between Groups A and B without Ala

Remove Leucine R_a 39.71204
D^2 14.1506 R_b 25.56144
T^2 47.16866 R_c 32.63674
F 7.718507
$F_{0.01,5,38}$ = 4.25
F > Critical value so reject Null Hypothesis

$D^2_m - D^2_{m-1}$
= 12.79065
F 1.815782
$F_{0.01,1,38}$ = 8.29
F < Critical value, DO NOT reject Null Hypothesis
So we can remove Leu and only have 5 amino acids

Remove Aspartic Acid and Valine R_a -68.3221
D^2 39.68727 R_b -108.009
T^2 132.2909 R_c -88.1657
F 20.56281
$F_{0.01,5,38}$ = 4.5
F > Critical value so reject Null Hypothesis

$D^2_m - D^2_{m-1}$
[Asp] - [Asp] = -5.452305
F -0.6754996
$F_{0.01,1,39}$ 8.18
F < Critical value, DO NOT reject Null Hypothesis
[Val] - [Asp] = -7.29372
F -0.9036371
degrees of $F_{0.01,1,39}$ 8.18
F < Critical value, DO NOT reject Null Hypothesis
So we can remove Asp, Val and only have 4 amino acids

Remove Aspartic Acid R_a -54.1884
D^2 34.23496 R_b -89.4234
T^2 114.1165 R_c -71.3059
F 18.67362
$F_{0.01,5,38}$ = 4.25
F > Critical value so reject Null Hypothesis

$D^2_m - D^2_{m-1}$
= -7.29372
F -0.8867113
$F_{0.01,1,38}$ = 8.29
F < Critical value, DO NOT reject Null Hypothesis
So we can remove Asp and only have 5 amino acids

Remove Phenylalanine R_a -47.478
D^2 54.25786 R_b -101.736
T^2 180.8595 R_c -74.607
F 29.5952
$F_{0.01,5,38}$ = 4.25
F > Critical value so reject Null Hypothesis

$D^2_m - D^2_{m-1}$
= -27.3166
F -2.90493
$F_{0.01,1,38}$ = 8.29
F < Critical value, DO NOT reject Null Hypothesis
So we can remove Phe and only have 5 amino acids

Remove Aspartic Acid and Phenylalanine R_a -109.283
D^2 61.55158 R_b -170.834
T^2 205.1719 R_c -140.058
F 44.29849
$F_{0.01,5,38}$ = 4.5
F > Critical value so reject Null Hypothesis

$D^2_m - D^2_{m-1}$
[Asp] - [Asp] = -27.31662
F -2.9416363
$F_{0.01,1,39}$ 8.18
F < so DO NOT reject Null Hypothesis
[Phe] - [Asp] = -7.29372
F -0.7854365
$F_{0.01,1,39}$ 8.18
F < Critical value, DO NOT reject Null Hypothesis
So we can remove Asp, Phe and only have 4 amino acids

Remove Glutamic Acid R_a 12.87607
D^2 25.45392 R_b -12.577
T^2 84.8464 R_c 0.15001
F 13.88396
$F_{0.01,5,38}$ = 4.25
F > Critical value so reject Null Hypothesis

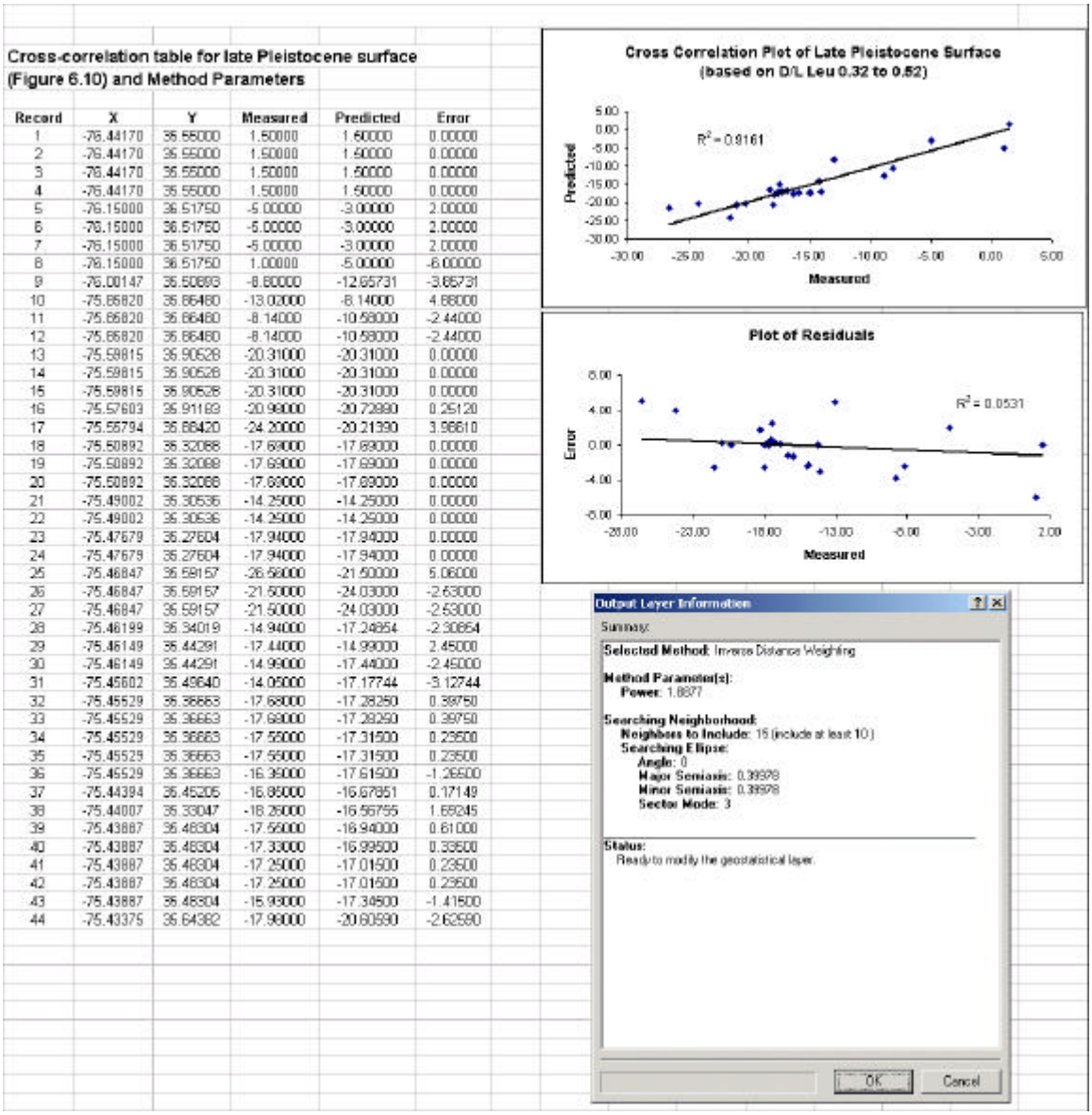
$D^2_m - D^2_{m-1}$
= 1.487322
F 0.192981 84
$F_{0.01,1,38}$ = 8.29
F < Critical value, DO NOT reject Null Hypothesis
So we can remove Glu and only have 5 amino acids

Remove Valine R_a -6.51739
D^2 32.39355 R_b -38.9109
T^2 107.9785 R_c -22.7142
F 17.66921
$F_{0.01,5,38}$ = 4.25
F > Critical value so reject Null Hypothesis

$D^2_m - D^2_{m-1}$
= -5.45231
F -0.67169
$F_{0.01,1,38}$ = 8.29
F < Critical value, DO NOT reject Null Hypothesis
So we can remove Val and only have 5 amino acids

APPENDIX III

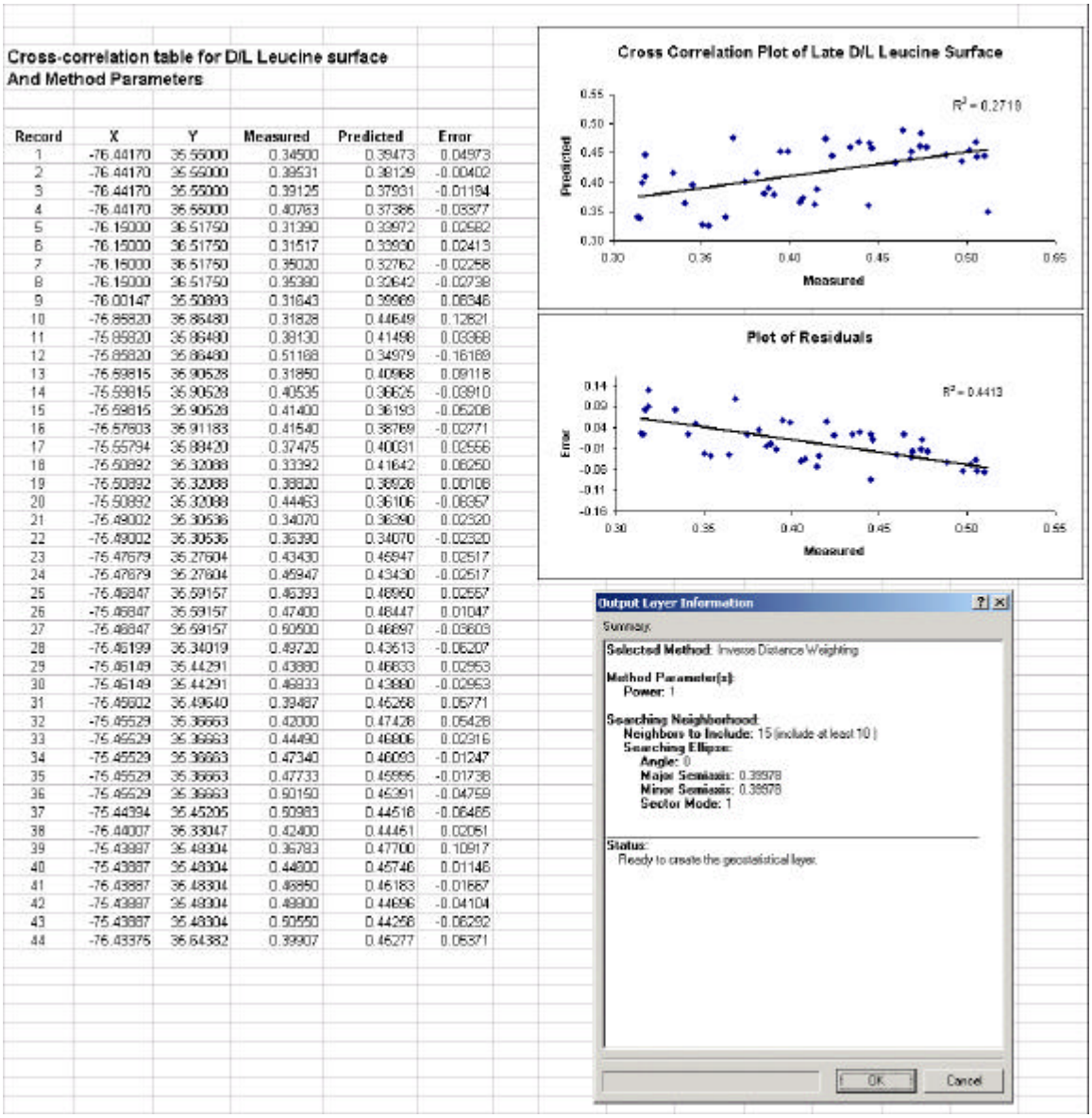
VALIDATION OF INTERPOLATED SURFACES



Cross-correlation table for late Pleistocene surface (Figure 6.10) and Method Parameters

Record	X	Y	Measured	Predicted	Error
1	-76.44170	35.55000	1.50000	1.60000	0.00000
2	-76.44170	35.55000	1.50000	1.60000	0.00000
3	-76.44170	35.55000	1.50000	1.50000	0.00000
4	-76.44170	35.55000	1.50000	1.50000	0.00000
5	-76.15000	36.51750	-5.00000	-3.00000	2.00000
6	-76.15000	36.51750	-5.00000	-3.00000	2.00000
7	-76.15000	36.51750	-5.00000	-3.00000	2.00000
8	-76.15000	36.51750	1.00000	-5.00000	-6.00000
9	-76.00147	35.50693	-8.80000	-12.65731	-3.85731
10	-75.86820	35.86480	-13.02000	-8.14000	4.88000
11	-75.86820	35.86480	-8.14000	-10.58000	-2.44000
12	-75.86820	35.86480	-8.14000	-10.58000	-2.44000
13	-75.59815	35.90528	-20.31000	-20.31000	0.00000
14	-75.59815	35.90528	-20.31000	-20.31000	0.00000
15	-75.59815	35.90528	-20.31000	-20.31000	0.00000
16	-75.57603	35.91189	-20.96000	-20.72880	0.25120
17	-75.55794	35.86420	-24.20000	-20.21390	3.98610
18	-75.50892	35.32088	-17.69000	-17.69000	0.00000
19	-75.50892	35.32088	-17.69000	-17.69000	0.00000
20	-75.50892	35.32088	-17.69000	-17.69000	0.00000
21	-75.49002	35.30536	-14.25000	-14.25000	0.00000
22	-75.49002	35.30536	-14.25000	-14.25000	0.00000
23	-75.47679	35.27604	-17.94000	-17.94000	0.00000
24	-75.47679	35.27604	-17.94000	-17.94000	0.00000
25	-75.46847	35.59157	-26.56000	-21.50000	5.06000
26	-75.46847	35.59157	-21.50000	-24.03000	-2.53000
27	-75.46847	35.59157	-21.50000	-24.03000	-2.53000
28	-75.46199	35.34019	-14.94000	-17.24954	-2.30954
29	-75.46149	35.44291	-17.44000	-14.99000	2.45000
30	-75.46149	35.44291	-14.99000	-17.44000	-2.45000
31	-75.45602	35.49640	-14.05000	-17.17744	-3.12744
32	-75.45529	35.36663	-17.68000	-17.28250	0.39750
33	-75.45529	35.36663	-17.68000	-17.28250	0.39750
34	-75.45529	35.36663	-17.55000	-17.31500	0.23500
35	-75.45529	35.36663	-17.55000	-17.31500	0.23500
36	-75.45529	35.36663	-16.35000	-17.61500	-1.26500
37	-75.44394	35.45205	-16.85000	-16.67851	0.17149
38	-75.44007	35.33047	-18.26000	-16.56755	1.69245
39	-75.43887	35.48304	-17.55000	-16.94000	0.61000
40	-75.43887	35.48304	-17.33000	-16.99500	0.33500
41	-75.43887	35.48304	-17.25000	-17.01500	0.23500
42	-75.43887	35.48304	-17.25000	-17.01500	0.23500
43	-75.43887	35.48304	-15.93000	-17.34500	-1.41500
44	-75.43375	35.64382	-17.96000	-20.60590	-2.62590

Figure A-3 Validation table and graphs for the late Pleistocene surface (Figure 6.11) interpolated from elevations of AAR determined Pleistocene *Mercenaria* (see Figure 6.9). Interpolation method parameters are also included.



Cross-correlation table for D/L Leucine surface
And Method Parameters

Record	X	Y	Measured	Predicted	Error
1	-76.44170	35.55000	0.34500	0.39473	0.04973
2	-76.44170	35.55000	0.39531	0.39129	-0.00402
3	-76.44170	35.55000	0.39125	0.37931	-0.01194
4	-76.44170	35.55000	0.40763	0.37386	-0.03377
5	-76.15000	35.51750	0.31390	0.33972	0.02582
6	-76.15000	35.51750	0.31517	0.33930	0.02413
7	-76.15000	35.51750	0.35020	0.32762	-0.02258
8	-76.15000	35.51750	0.35380	0.32642	-0.02738
9	-76.00147	35.50893	0.31643	0.39989	0.08346
10	-75.85820	35.86480	0.31828	0.44649	0.12821
11	-75.85820	35.86480	0.38130	0.41498	0.03368
12	-75.85820	35.86480	0.51168	0.34979	-0.16189
13	-75.59815	35.90528	0.31850	0.40968	0.09118
14	-75.59815	35.90528	0.40535	0.36625	-0.03910
15	-75.59815	35.90528	0.41400	0.36193	-0.05208
16	-75.57603	35.91183	0.41540	0.38769	-0.02771
17	-75.55794	35.88420	0.37475	0.40031	0.02556
18	-75.50892	35.32088	0.33392	0.41642	0.08250
19	-75.50892	35.32088	0.38820	0.38928	0.00108
20	-75.50892	35.32088	0.44463	0.36106	-0.08357
21	-75.49002	35.30536	0.34070	0.36990	0.02920
22	-75.49002	35.30536	0.36390	0.34070	-0.02320
23	-75.47679	35.27604	0.43430	0.45947	0.02517
24	-75.47679	35.27604	0.45947	0.43430	-0.02517
25	-75.46847	35.59157	0.46393	0.48950	0.02557
26	-75.46847	35.59157	0.47400	0.48447	0.01047
27	-75.46847	35.59157	0.50500	0.46897	-0.03603
28	-75.46199	35.34019	0.49720	0.43513	-0.06207
29	-75.46149	35.44291	0.43880	0.46833	0.02953
30	-75.46149	35.44291	0.46933	0.43980	-0.02953
31	-75.45802	35.49640	0.39487	0.45258	0.05771
32	-75.45529	35.36663	0.42000	0.47428	0.05428
33	-75.45529	35.36663	0.44490	0.46806	0.02316
34	-75.45529	35.36663	0.47340	0.46093	-0.01247
35	-75.45529	35.36663	0.47733	0.45995	-0.01738
36	-75.45529	35.36663	0.50150	0.45391	-0.04759
37	-75.44394	35.45205	0.50983	0.44516	-0.06465
38	-75.44007	35.33047	0.42400	0.44461	0.02061
39	-75.43887	35.48304	0.36783	0.47700	0.10917
40	-75.43887	35.48304	0.44600	0.45746	0.01146
41	-75.43887	35.48304	0.48850	0.46183	-0.01667
42	-75.43887	35.48304	0.48900	0.44696	-0.04104
43	-75.43887	35.48304	0.50550	0.44256	-0.06292
44	-76.43375	36.64382	0.39907	0.46277	0.06371

Figure A-4 Validation table and graphs for the surface interpolated from D/L Leucine values of AAR determined Pleistocene *Mercenaria*. Interpolation method parameters are also included. The surface can be viewed in \AAR_Database\MyGIS\3DSiteMap.sxd found in Appendix IV.

REFERENCES

Adam, N.R. and Gangopadhyay, A., 1997, Database Issues in Geographic Information Systems: Boston, Kluwer Academic Publishers, 136 p.

Allen, J.H., 1988, The World Data Center System, international data exchange and global change. *in* Mounesy, H. and Tomlinson, R., editors, Building Databases for Global Science: New York, Proceedings of the first meeting of the International Geographical Union Global Database Planning Project, held at Tylney Hall, Hampshire, UK, 9-13 May 1988, p. 138-152.

Andrews, J.T., Miller, G.H., Davies, D.C., Davies, K.H., 1985, Generic identification of fragmentary Quaternary molluscs by amino acid chromatography: a tool for Quaternary and Palaeontological research. Geological Journal, v. 20, p. 1-20.

Codd, E.F., 1970, A relational model of data for large shared data banks. Communications of the ACM, v. 13, n. 6, p. 377-387.

Coffey, J.R., Marsden, A.E., Reekie, C.J., 1982, A flexible computer based system for the reconstruction of subsurface geological conditions for ground engineering purposes. *in* Horder, M.F. and Howarth, R.J., editors, Computer Applications in Geology I & II; Miscellaneous Paper No 14: London, Geological Society of London, p. 185-197.

Corrado, J.C., Weems, R.E., Hare, P.E. and Bambach, R.K., 1986, Capabilities and limitations of applied aminostratigraphy, as illustrated by analyses of *Mulinia Lateralis* from the late Cenozoic marine beds near Charleston, South Carolina. South Carolina Geology, v. 30, no. 1, p. 19-46.

Davis, J.C., 1986, Statistics and Data Analysis in Geology, 2nd Edition: New York, John Wiley and Sons, 638 p.

Dittert, N., World Data Center for Marine Environmental Sciences provides lessons in marine geosciences data management: EOS, Transactions, American Geophysical Union, v. 84, n. 16, 22 April 2003.

Elmasri, R. and Navathe, S.B., 2000, Fundamentals of Database Systems, Third Edition: Reading, MA., Addison-Wesley, 955 p.

Fletcher, C.H., 1987, Stratigraphy and Reconstruction of the Holocene Transgression: A computer aided study of the Delaware Bay and inner Atlantic shelf [Ph.D Dissertation]: Newark, University of Delaware, Department of Geology, 468 p.

Goodfriend, G., 1991, Patterns of racemization and epimerization of amino acids in land snail shells over the course of the Holocene. Geochimica et Cosmochimica Acta, v.55, no.1, p.293-302.

Gunderson, K.D., editor, 1994, International inventory of automated databases in the geosciences: U.S. Geological Survey Open-File Report 94-299, 321 p.

Harris, M.S., 2000, Influence of complex geologic framework on Quaternary coastal evolution: An example from Charleston, South Carolina [Ph.D Dissertation]: Newark, University of Delaware, Department of Geology, 331 p.

Hill, G.W. and Walton, R.J., 1988, Global database planning efforts by the U.S. Geological Survey. *in* Mounesy, H. and Tomlinson, R., editors, Building Databases for Global Science: New York, Proceedings of the first meeting of the Internationa Geographical Union Global Database Planning Project, held at Tylney Hall, Hampshire, UK, 9-13 May 1988, p. 296-306.

Hoffman, D.R., 2003, Effective Database Design for Geoscience Professionals: Tulsa, OK, PennWell Corporation, 263 p.

Kimber, R.W.L. and Griffin, C.V. and Milnes, A.R., 1986, Amino acid racemization dating: Evidence of apparent reversal in Aspartic Acid racemization with time in shells of *Ostrea*. Geochimica et Cosmochimica Acta, v.50, p. 1159-1161.

Kimber, R.W.L. and Griffin, C.V., 1987, Further evidence of the complexity of the racemization process in fossil shells with implications for amino acid racemization dating. Geochimica et Cosmochimica Acta, v.51, p. 839-846.

Lajoie, K.R., Wehmiller, J.F., and Kennedy, G.L., 1980, Inter- and intrageneric trends in apparent racemization kinetics of amino acids in Quaternary Mollusks, *in* Hare, P.E., Hoering, T.C. and King, K. Jr., editors, Biogeochemistry of Amino Acids: New York, John Wiley and Sons, p. 305-340.

Miller, G.H. and Brigham-Grette, J., 1989, Amino acid geochronology: Resolution and precision in carbonate fossils. Quaternary International, v. 1, p. 111-128.

National Research Council, 2001, National Spatial Data Infrastructure Partnership Programs, Rethinking the Focus: Washington, D.C., National Academy Press, 82 p.

Riggs, S.R., York, L.L., Wehmiller, J.F. and Snyder, Stephen W., 1992, Depositional patterns resulting from high frequency Quaternary sea-level fluctuations in northeastern North Carolina. *in* Fletcher, C.H. and Wehmiller, J.F., editors, Quaternary Coasts of the United States. SEPM (Soc. Sediment. Geol.) Special Publication, v. 28, p. 141-154.

Riggs, Stanley R., Cleary, William J., Snyder, Stephen W., 1995, Influence of inherited geologic framework on barrier shoreface morphology and dynamics. Marine Geology. v. 126, p. 213-234.

Suckow, A., and Ingolf, D., 2001, A database system for geochemical, isotope hydrological, and geochronological laboratories. Radiocarbon, v. 43, p. 325-337.

Szabo, B.J., 1985, Uranium-series dating of fossil corals from marine sediments of southeastern United States Atlantic Coastal Plain. Geological Society of America Bulletin, v. 96, p. 398-406.

Thieler, E. R., Riggs, S. R , Hoffman, C. W., Wehmiller, J. F., Mallinson, D. J., Foster, D. S., Culver, S. J.,Farrell, K. M., Bratton, J. F., McNinch, J. E., 2003, The record of Quaternary sea-level change, North Carolina Coastal Plain, Mid-Atlantic US. Abstracts, 2003 International Quaternary Association (INQUA) Meeting, Reno NV.

USGS, 2001, Drilling Quaternary Sediments of the Barrier Island – Estuarine System: http://woodshole.er.usgs.gov/project-pages/northcarolina/html/ncgs.htm, (July 6, 2001).

Wehmiller, J.F., 1980, Intergeneric Differences in Apparent Racemization Kinetics in Mollusks and Foraminifera: Implications for Models of Diagenetic Racemization. *in* P.E. Hare, T.C. Hoering, and K. King, Jr. (eds.), Biogeochemistry of Amino Acids: John Wiley and Sons, New York, pp. 341-355.

Wehmiller, J.F., 1984, Relative and absolute dating of Quaternary mollusks with amino acid racemization: Evaluation, applications and questions. *in* Mahaney, W.C., editor, Quaternary dating methods: Developments in palaeontology and stratigraphy, Elsevier, New York, p. 171-193.

Wehmiller, J.F., 1986, Amino acid racemization geochronology. *in* Hurford, A. J., Jager, E. and Ten Cate, J.A.M., editors, Dating Young Sediments, Proceedings of 1985 Beijing Workshop on Dating of Young Sediments, United Nations CCOP, Bangkok, pp. 139-158.

Wehmiller, J.F., 1993, Applications of Organic Geochemistry for Quaternary Research: aminostratigraphy and aminochronology. *in* Engel, M. and Macko, S., editors, Organic Geochemistry, Plenum Publishing Company, New York, p. 755-783.

Wehmiller, J.F. and Miller, G.H., 2000, Aminostratigraphic Dating Methods in Quaternary Geology. *in* Noller, J.S., Sowers, J.M., Lettis, W.R., editors, Quaternary Geochronology, Methods and Applications: Washington, D.C., American Geophysical Union, p.187-222.

Wehmiller, J.F., York, Linda L., Bart, Michelle L., 1995, Amino acid racemization geochronology of reworked Quaternary mollusks on U.S. Atlantic coast beaches: implications for chronostratigraphy, taphonomy, and coastal sediment transport. Marine Geology. v.124, p. 303-337.

Wehmiller, J. F., Thieler, E. R., York, L. L., Pellerito, V., 2002, Aminostratigraphy of Subsurface Units, Eastern Albemarle Sound and Northern Outer Banks, North Carolina. Fall 2002 AGU meeting, San Francisco.

York, L.L., 1984, Aminostratigraphy of Stetson Pit and Ponzer Areas of North Carolina By Pleistocene Mollusk Analysis: unpublished Master's Thesis: Newark, University of Delaware, Department of Geology 188 p.

York, L.L., 1990, Aminostratigraphy of U.S. Atlantic coast Pleistocene deposits: Maryland continental shelf and North and South Carolina coastal plain [Ph.D Dissertation]: Newark, University of Delaware, Department of Geology, 580 p.

York, L.L., Wehmiller, J.F., Cronin, T.M. and Ager, T.A., 1989, Stetson Pit, Dare County, North Carolina: An integrated chronologic, faunal, and floral record of subsurface coastal Quaternary sediments. Palaeogeography, Palaeoclimatology, Palaeoecology, v. 72, p. 115-132.

York, L.L. and Wehmiller, J.F., 1992, Aminostratigraphic results from Cape Lookout, N.C., and their relation to the preserved Quaternary marine record of SE North Carolina. Sedimentary Geology, v. 80, p. 279-291.

Young, R.P., 1982, A computer based system for the storage, retrieval and analysis of geophysical and geotechnical data: *in* Horder, M.F. and Howarth, R.J., editors, Computer Applications in Geology I & II; Miscellaneous Paper No 14: London, Geological Society of London, p. 32-55.