AUTOMATED DETECTION OF FREQUENCY SPECIFIC FLUCTUATIONS IN ECG MORPHOLOGY

by

Lance Elliot Jackson

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL
ENGINEERING AND COMPUTER SCIENCE IN
PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

BACHELOR OF SCIENCE IN ELECTRICAL SCIENCE AND ENGINEERING

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

December 1990

Signature of Author ______________________________
Department of Electrical Engineering and Computer Science
December 17, 1990

Certified by ______________________________
Richard J. Cohen
Thesis Supervisor

Certified by ______________________________
David S. Rosenbaum
Thesis Supervisor

Accepted by ______________________________
Leonard A. Gould
Chairman, Department Committee on Undergraduate Theses

AUTOMATED DETECTION OF FREQUENCY SPECIFIC FLUCTUATIONS IN ECG MORPHOLOGY

by

Lance Elliot Jackson

Submitted to the Department of Electrical Engineering and Computer Science on December 14, 1990 in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical Science and Engineering.

Abstract

Electrical alternans (EA) is defined as a beat-to-beat alternation in electrocardiogram (ECG) morphology and is thought to be a possible marker of vulnerability to cardiac arrythmias. An automated computer system was developed to measure low level alternating-type ECG oscillations. The system utilizes a display program for visual checking of the ECG recordings. It then links the display program to the spectral analysis programs, which are provided the proper parameters. Finally, output files and hardcopies of the results are created which allow rapid and easy evaluation of the validity of the results and provide statistical measures of EA. In addition to measuring EA from ECG recordings, the automated system can be expanded to perform spectral analysis for variability on other types of waveforms.

Thesis Supervisor:	Richard J. Cohen
Title:	Hermann Von Helmholtz Associate Professor of Health Sciences and Technology
Thesis Supervisor:	David S. Rosenbaum
Title:	Instructor in Medicine, Harvard Medical School / Visiting Scientist, Massachusetts Institute of Technology

Table of Contents

Abstract 2
Table of Contents 3
List of Figures 4
List of Tables 5

1. Introduction 6
   1.1 Aims of Project 6
   1.2 Background 8
      1.2.1 History of Electrical Alternans 8
      1.2.2 Experimental Results Correlating Electrical Alternans with Cardiac Instability 9

2. Procedures and Techniques of Analysis 10
   2.1 Data Acquisition 10
   2.2 Analysis Procedures 11
      2.2.1 Creation and Verification of Data, Annotation, and Header Files 11
      2.2.2 Fiducial Point Refinement 13
      2.2.3 Optimal Data Segment Determination 14
      2.2.4 Spectral Analysis 15
      2.2.5 Creation of Final Results 16

3. Manual 17
   3.1 Flow 17
   3.2 Program Descriptions 18

4. Results of Analysis 31
   4.1 Results Provided as Output 31
      4.1.1 Numerical Results 35
      4.1.2 Plots for Validation of Analysis Process 36
      4.1.3 Spectral Results 38
   4.2 Evaluation of Quality of Results 39
      4.2.1 Effects of Replacing Erroneous Beats 39
      4.2.2 Effects of Not Replacing Erroneous Beats 40

5. Discussion and Conclusions 44
   5.1 Significance of Research 44
   5.2 Future Potential of Automated System 45
   5.3 Future Technical Improvements 46
   5.4 Fulfillment of Goals 47

Acknowledgements 49
References 50

Appendix A. Commented Code 52

List of Figures

Figure 1-1: Electrical Alternans of the T-wave 6
Figure 2-1: Acquisition of Data 11
Figure 2-2: Essential Flow in Alternans Analysis 12
Figure 3-1: Display Used in Assigning Shifts 23
Figure 3-2: Proper Assignment of Intervals 29
Figure 4-1: Numerical Results (Page 1 of Output) 32
Figure 4-2: Plots for Validation of Analysis Process (Page 2 of Output) 33
Figure 4-3: Spectral Results (Page 3 of Output) 34
Figure 4-4: Statistical Measures of Electrical Alternans 37
Figure 4-5: First Iteration of Example 42
Figure 4-6: Second Iteration of Example 43

List of Tables

Table 4-I: Effects of Replacing Measured Beats with Template Beats 40

Chapter 1

Introduction

1.1 Aims of Project

Electrical altemans (EA) is defined as a beat-to-beat altemation of electrocardiogram (ECG) morphology. This electrical manifestation of cardiac variability occurs on an every-other-beat basis following an [ABABAB...] pattem. Morphology altemation can be observed as changes in the amplitude, width, and/or shape of the ECG wavefom. Additionally, EA can involve any individual component (e.g. QRS, ST-segment, or T-wave) of the ECG complex, or can simultaneously effect the entire ECG wavefonn. Figure 1-1 illustrates the A-B-A-B pattem of EA, with altemating morphology apparent in the T-wave of the ECG. Consequently, we would like to selectively measure EA in individual components of the ECG complex by measuring ECG morphology fluctuations occurring at half of the frequency of the heart rate.

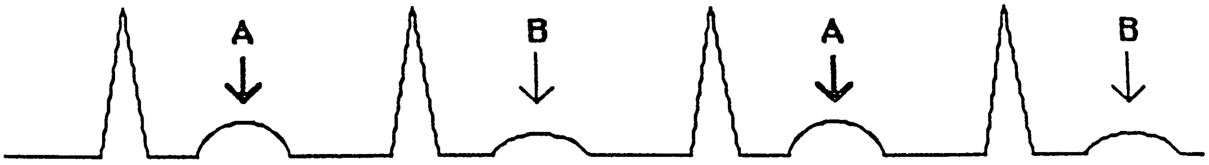


Figure 1-1: Electrical Altemans of the T-wave

Although EA has been observed in clinical ECG recordings since the first part of this century, altemans has only recently been associated with the occurrence of sudden cardiac death. Many reports have indicated that altemating ECG morphology precedes ventricular fibrillation in certain clinical situations. However, the exact role of EA in the pathogenesis of arrythmias remains unclear.

Recent work has suggested that EA is a body surface manifestation of temporal dispersion of repolarization, which in turn may arise from spatial dispersion of repolarization and consequently be coupled to the genesis of reentrant arrythmias [Rosenbaum, et al 90, Smith 86, Smith, et al 84a, Smith, et al 84b]. Additionally, sensitive measures of EA have correlated with increased susceptibility to ventricular arrythmias in dogs subjected to arrythmogenic interventions [Smith, et al 88]. Finally, preliminary data indicates that EA may in fact be present in patients who are at increased risk for ventricular arrythmias [Smith, et al 88].

These findings have been under-appreciated in the past because EA is often not apparent upon casual visual inspection of ECG waveforms. Preliminary work in our laboratory has shown that the majority of noise from body surface recordings (which has obscured detection of alternans) is due to respiratory motion of the chest wall. However, EA of the surface ECG of dogs may be isolated from respiration and other frequency specific noise sources using a multi-dimensional spectral technique [Smith, et al 88]. The purpose of our current study is to implement such a technique for detection of EA in the human ECG and to ascertain whether low level beat-to-beat cardiac alternations is a marker of underlying electrical instability in man.

The primary goal of this thesis was to design a system that would perform EA analysis of ECG data with minimal user interaction, since it must be applied to clinical settings. We believed that by making the process highly automated, we could furnish a system relatively easy to learn how to use for an inexperienced person without in-depth knowledge of cardiac electrophysiology or signal processing. Additionally, we felt automation would provide maximal consistency between data sets by reducing the possibility for user induced variability and inconsistency. The second goal of the thesis was to design the software package to be generally applicable to many types of cardiac potentials (e.g. action potentials, extracellular electrograms, etc.) By making the system easily adaptable, alternans analysis performed by the system need not be restricted to clinical ECG data.

1.2 Background

1.2.1 History of Electrical Alternans

Electrical alternans was first reported in 1908 [Hering 08]. In 1936, Hamburger [Hamburger, et al 36] reported occurrences of EA in only one of 10,000 ECGs. Finally in 1950, when Hellerstein and Leibow [Hellerstein, et al 50] elicited ST and T-wave alternans after acute coronary occlusion in the dog, the prescence of EA began to be associated with increased susceptibility to arrythmias. Currently, there are more than 500 reported cases of EA in the world's literature.

EA can generally result from either mechanical or electrical origins. Mechanical alternans is caused by the motion of the heart within the chest cavity and can involve any component of the ECG waveform. Our analysis system is currently not being used to study alternans of mechanical origin; rather, it is focusing on EA resulting from oscillating electrical properties typically involving repolarization phases of the cardiac cycle.

Mines [Mines 13] was the first to propose an electrical basis for EA, and it is now apparent that electrogenic ST and T wave EA exists in a variety of pathologic conditions. Increased susceptibility to ventricular arrythmia is also observed in the majority of these conditions; however, the relationship between EA and arrythmia susceptibility has not been systematically studied in man. Herein lies the rationale for our current study.

There are presently two predominant theories regarding the mechanism of EA. The first hypothesis describes EA on a macroscopic level by maintaining that subpopulations of cells with extended refractory periods may respond to every other stimulus, leading to alternating patterns of conduction in the heart. Thus, the mechanism of EA is attributed to alternating wavefront fractionation and is consequently linked to reentrant arrythmogenesis [Smith, et al 84a]. The second theory describes the mechanism of EA on a cellular level, maintaining that EA is a result of fluctuation in action potential morphology. This situation would lead to global EA, even if the macroscopic conduction sequence remained intact. Supporting this hypothesized mechanism, the magnitude of action potential alternation was shown to strongly correlate with simultaneous EA of the total ECG [Nakashima, et al 78], indicating a causal relationship between action potential and ECG alternation. Moreover, we have shown that spatial dispersion of recovery (a condition thought to presage reentry) can result from non-uniform action potential oscillations [Rosenbaum, et al 90], suggesting that alternation could indeed be linked with risk to arrythmia.

1.2.2 Experimental Results Correlating Electrical Alternans with Cardiac Instability

There has been a variety of experiments in animals relating EA and cardiac electrical instability. For example, Russel et al [Russel, et al 79] reported EA preceding ventricular fibrillation (VF) due to ischemia in 95% of the experiments performed, suggesting alternans as a precursor of VF. Other studies have demonstrated that EA can be very subtle and require more sensitive detection methods than visual analysis. Thus, digital signal processing techniques were developed (the multi-dimensional spectral technique) and utilized in an effort to attain maximal sensitivity.

Initial studies performed on dogs in our laboratory demonstrated that a strong inverse relation exists between measures of EA and VF threshold. The greatest ECG alternation was measured within the T wave, suggesting that repolarization is effected most. Thus, these data indicate that there is indeed a strong relationship between EA and electrical instability. The hypothesis that surface ECG recordings of beat-to-beat oscillations can provide a non-invasive probe of vulnerability to arrythmia is supported by these findings.

Chapter 2

Procedures and Techniques of Analysis

This chapter discusses the general procedures used in the analysis of data. It focuses on the basis for the techniques used and discusses their advantages and disadvantages. For details of the software and its use, see Chapter 3.

2.1 Data Acquisition

For our current project, the data was collected in the clinical Electrophysiology Laboratory of the Massachusetts General Hospital, Boston. Patients were referred for electrophysiological testing for a variety of indications, which included documented ventricular tachycardia and ventricular fibrillation. Each patient underwent programmed cardiac stimulation to ascertain susceptibility to ventricular arrythmias [Ruskin, et al 83]. Prior to programmed cardiac stimulation, ECGs were recorded during atrial pacing for approximately three minutes. ECG electrodes were positioned in an orthogonal configuration (X,Y,Z) and recorded onto a Hewlett Packard ECG recorder. Initially, the data was recorded to FM magnetic tape, from which it was later digitized onto a laboratory microcomputer (500 Hz with 12 bit precision) with appropriate low pass anti-alias filtering. In later studies, a real time data acquisition system was utilized which digitally recorded the waveforms onto floppy disk. This real time acquisition system is used as the current mode of analysis. Figure 2-1 on page 11 illustrates the acquisition of the data.

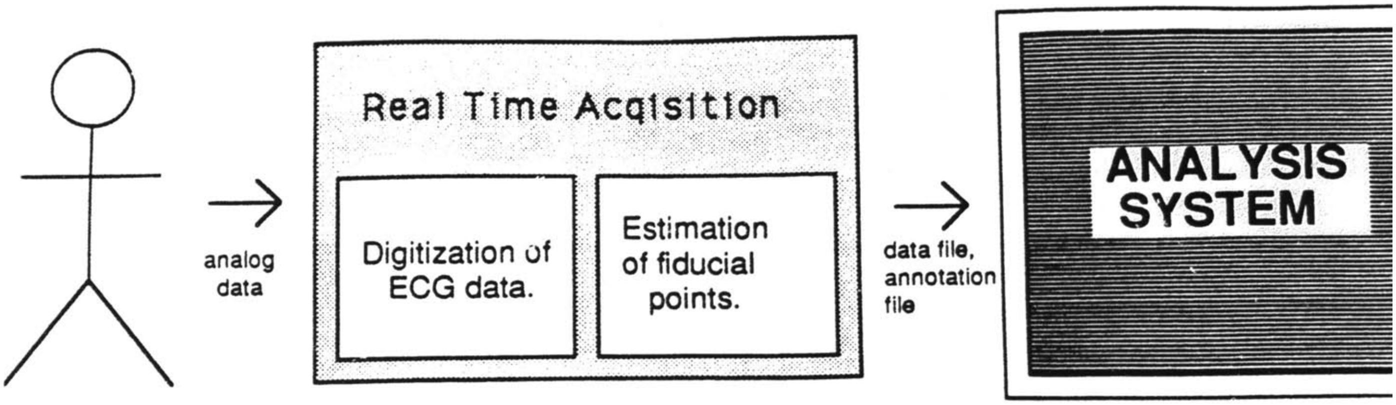

analog data
Real Time Acqisition
Digitization of ECG data.
Estimation of fiducial points.
data file, annotation file
ANALYSIS SYSTEM

Figure 2-1: Acquisition of Data

2.2 Analysis Procedures

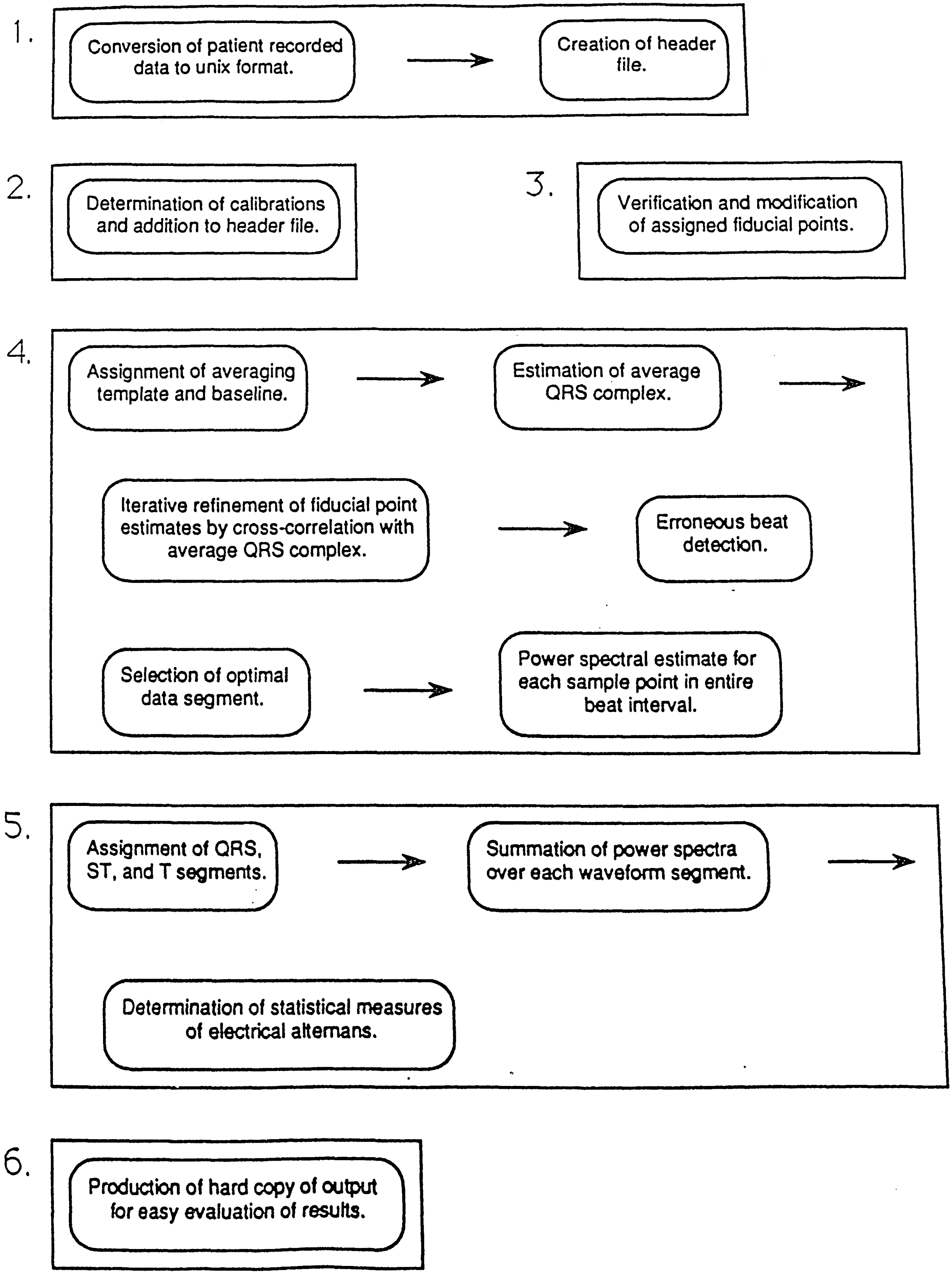
A flow chart showing the major stages essential for a complete analysis is found in figure 2-2. The analysis system designed for this thesis performs these steps.

2.2.1 Creation and Verification of Data, Annotation, and Header Files

The first stage of the analysis procedure involves preparing the initial data file (which contains the digitized ECG data) and annotation file (which contains a fiducial point estimate corresponding to the QRS complex of each beat). A file containing the basic description of the data file (known as a header file) is also created. Secondly, the gains used in recording the data are determined from the calibrations and added to the header file for later reference.

A third stage involves verification and modification of the fiducial point estimates previously assigned during the recording of the data. The goal of the user at this point is the elimination of spurious annotations and the addition of necessary annotations that are absent. This action is accomplished by viewing the waveforms in a display program which

Figure 2-2: Essential Flow in Altemans Analysis


1.
Conversion of patient recorded data to unix format.
Creation of header file.
2.
Determination of calibrations and addition to header file.
3.
Verification and modification of assigned fiducial points.
4.
Assignment of averaging template and baseline.
Estimation of average QRS complex.
Iterative refinement of fiducial point estimates by cross-correlation with average QRS complex.
Erroneous beat detection.
Selection of optimal data segment.
Power spectral estimate for each sample point in entire beat interval.
5.
Assignment of QRS, ST, and T segments.
Summation of power spectra over each waveform segment.
Determination of statistical measures of electrical altemans.
6.
Production of hard copy of output for easy evaluation of results.

allows modification of the annotation file. Additionally, the user is expected to note the quality of the data while viewing it and determine if the data (or specific channels) should be eliminated from the study due to poor quality. False and missing annotations can be previously pinpointed by plotting the assigned beat-to-beat intervals versus beat number (a positive spike on this graph indicates a missed annotation whereas a negative spike indicates an extra annotation).

This third stage is very important because it is imperative that there be no missed annotations or incorrectly assigned additional annotations. The reason for this is that proper beat-to-beat phase must be maintained; odd beats must continually fall on the odd cycle while even beats must fall on the even cycle. However, if an improper annotation assignment is located in the middle of the data segment so that the observed phase is altered by 180°, the detected beat-to-beat altemation is impaired because the distinction between even and odd beats is lost. Thus, destructive interference of the spectrum (and consequently of the measure of EA) will occur if the respective phases of every beat is not correctly interpreted by the system.

2.2.2 Fiducial Point Refinement

The fourth stage is the major computation stage of the analysis and has many substeps. The first step requires the user to assign an appropriate baseline and an averaging template. The baseline is important because it is the zero voltage reference point for the analysis of the data which is maintained throughout the duration of the analysis. Improper assignment of the baseline can cause false amplification or reduction in the measured degree of EA.

An average QRS complex is determined by first creating a vector magnitude ECG for the entire waveform by calculating the square root of the sum of squares of each channel's baseline corrected data. Then an average vector magnitude QRS complex is created by aligning each beat according to the fiducial point estimates and averaging the waveforms about the fiducial points. The region used in averaging is specified by the user.

The subsequent step is iterative refinement of the the fiducial point estimates. Using the average QRS complex as a template for comparison, the fiducial points are adjusted to obtain maximum cross-correlation between each individual beat and the template. Then the average QRS complex is recalculated using the new fiducial points estimates for alignment, and the cross-correlation step is iteratively executed. We have found that two iterations are almost universally adequate in our analysis to obtain convergence of the fiducial point estimates.

2.2.3 Optimal Data Segment Determination

It is important to identify ECG complexes exhibiting differing morphology for later elimination from the spectral analysis. These beats with markedly different ECG morphology and timing are typically premature ventricular complexes. If included in the spectral analysis, these waveforms can have destructive effects on the beat-to-beat alternation measured by our spectral method because the large abnormal differences present in these beats tend to mask the degree of lower amplitude EA present. The criteria used for assignment of erroneous beats are low correlations and abnormal peak-to-peak (RR) intervals. For example, any ECG complex, when cross-correlated with the average template ECG and found to have a cross-correlation coefficient less than 0.95, was considered an erroneous complex. Also, any ECG R-wave occurring 50 milliseconds early or late was considered distinctive of an erroneous complex. Both of these gauges combine to be a moderately effective identifier of ectopic beats and waveforms of differing morphology. It must be noted, however, that waveforms of obviously different visual morphology often have high cross-correlations and can escape detection by this method.

The next step in the analysis requires determination of the optimal data segment consisting of the desired number of consecutive beats (128 beats in our current project). It is imperative that the 128 analyzed beats be consecutive so that proper beat-to-beat phase is maintained. The best segment is defined to be the 128 beat segment of data exhibiting the fewest number of previously determined erroneous beats. Often, it is not possible to identify a segment of data containing 128 consecutive beats with no erroneous beats. Consequently, any erroneous beats present in the segment are eliminated and replaced with the average ECG waveform of the data during the spectral analysis. By choosing an optimal segment and reducing the number of ECG complexes that need to be eliminated from the analysis, the accuracy of the measure of EA is increased.

2.2.4 Spectral Analysis

All of the previous steps serve as preparation for analysis of the data by our multi-dimensional spectral estimation technique. This analysis is performed on the vector magnitude ECG waveform of the optimal data segment with the erroneous beats replaced by the average beat waveform. First, each of the consecutive 128 beats are aligned according to fiducial points and superimposed. Then a data matrix is constructed which contains 128 rows (each corresponding to a successive beat) and *N* columns (each corresponding to one sample point of the beat). Next, each column of the data matrix undergoes power spectral estimation, which is performed by discrete Fourier transform. This procedure produces an alternate dimensional spectrum for each sample point of the aligned waveforms. These individual spectra allow summation over a certain prescribed segment of data points to yield a total spectrum for that segment. A measure of EA can then be determined by evaluating the spectrum at half the frequency of the heart rate.

There are several reasons why this multi-dimensional spectral algorithm is utilized by the analysis system for measuring EA in ECG morphology. First, the magnitude of ECG alternation was shown by Smith [Smith, et al 88] to be on the order of one to 50 microvolts, which demonstrates the necessity of a sensitive method for detecting EA. We feel that frequency domain techniques are the most appropriate because they allow selective isolation and evaluation of the magnitude of specific frequencies of beat-to-beat waveform fluctuations. Since EA is a frequency specific event (i.e. occurs at half of the frequency of the heart rate), spectral analysis is particularly advantageous for its detection. Simultaneously, the effects of respiratory modulation of ECG morphology, muscle artifact, 60 cycle interference, and other noise sources which tend to obscure the amount of alternation observable in the time domain can be eliminated.

2.2.5 Creation of Final Results

The fifth stage of the analysis involves the creation of the final results. Initially, the user is asked to denote the QRS, ST, and T intervals of the average vector magnitude ECG waveform. The user is also expected to exclude from the analysis regions containing unwanted noise which can contaminate the results, such as pacing artifacts. Given these intervals, the power spectra of the points in each segment are algebraically summed over the interval to yield a total spectrum for the interval. Next, a series of statistical measures of EA are calculated and final displays are created. Finally, in the sixth and last stage, hardcopies of the aggregate results and displays are produced, allowing the user to assess the results of the analysis. Section 4.1 beginning on page 31 gives a more detailed description of the displays and statistical techniques used in the final analysis of the data.

Chapter 3

Manual

This chapter of the thesis is designed to serve as a concise manual for the automated cardiac variability measuring system. This manual (as well as the rest of the thesis) can serve as an introduction to the analysis for those who have never used the system. For those who have used the system, this chapter can serve as a reference manual for greater familiarization with the system and can answer specific questions about modifying the default parameters.

3.1 Flow

The following is the flow of programs invoked by the user in the analysis of a data file:

1. convert
2. cals
3. display
4. alternans
5. final
6. results

These programs are listed in the sequence in which they would generally be called. Each one of these programs may call additional programs, which are also described in this manual. This flow and the purpose of each stage is pictorially represented in figure 2-2 on page 12. The stage numbers given above correspond to the stage numbers appearing in the figure.

3.2 Program Descriptions

Program: convert

Function: Converts data file and annotation file acquired in real-time to unix format for analysis. Also creates a descriptive file called *header*.[*data_file*].

Invocation: User-invoked in step 1 of flow.

Arguments: Interactive.

Comments: This program swaps the byte order of the data file obtained in real time and puts the data file in *$dpath*/[*data_file*]. The *$dpath* is the directory where the data file is stored, and must be set in the shell if the default path is not desired. This *$dpath* must be maintained until alternans has been run.

The program also creates the annotation file of desired format, which is named *RR*.[*intervention_code*].[*data_file*], and puts it into a subdirectory named *rec*.[*data_file*] located in the directory *$path*. *$path* is set inside of convert. *$path* contains all of the *rec*.[*data_file*] directories, which subsequently contain all of the output files for each particular data file.

Finally, this program creates the header file, which is by convention given the name *header*.[*data_file*]. The header file contains important parameters of the data file, such as the file number, the sampling rate, the number of channels in the data recording, and the gains of each channel. The header file is put in the directory *$DB*, which contains a header file for each data file. *$DB* is assigned inside of convert

Program: cals

Function: Determines calibrations used in data recording and subsequently appends the gains to the file *header*.[*data_file*].

Invocation: User-invoked in step 2 of flow.

Arguments: Interactive.

Comments: This program invokes the display program show for annotation editing of data files. Once in the editor, the user is asked to mark the baseline and the peak of the calibration step. The user then exits from the program and the gains are calculated and appended to *header*.[*data_file*]. Note that if a particular channel appears to be of bad quality, the user should change the gain for that channel to 0.0 in the header file.

....

Program: display

Function: Simultaneously displays the data with the file *RR*.[*intervention_code*].[*data_file*] for verification and editing of the estimated fiducial points. Also for determining quality of data obtained.

Invocation: User-invoked in step 3 of flow.

Arguments: Interactive.

Comments: This programs displays the file *RR*.[*intervention_code*].[*data_file*] (file of annotations determined in real time) with the data file. This allows the user to delete spurious annotation assignments and assign annotations that were missed. At the same time, the user is expected to examine the quality of the data and determine if the data is usable.

A good way for the user to check beforehand if the annotation file needs editing is to plot the RR intervals over time and see if there are any outlying points. A far outlying point can indicate a missed annotation or a spurious annotation. The plotting can easily be done with the following command:

plt *RR*.[*intervention_code*].[*data_file*] 2

The 2 refers to the second column of the annotation file, which contains the RR intervals. Subsequently, the user can invoke the program **outlier** to identify the sample point numbers of the outlying points. Then the user can search for the sample point numbers within **display**.

....

Program: outlier

Function: Determines location of RR intervals which deviate more than the maximum allowable threshold from the median RR interval.

Invocation: Optional invocation by user (only if necessary).

Arguments: Interactive.

Comments: This program finds the sample point numbers of annotations following long or short RR intervals. Thus, if such annotations exist, the user can search for the annotations within display and perform the necessary corrections.

....

Program: show

Function: Simultaneously displays data file with annotations from an annotations file.

Invocation: Invoked by cals and display.

Arguments:

[*data_file*] (e.g. *$dpath/*[*data_file*])

{[*#_of_bytes*],[*#_of_channels*])} (e.g. {2,3}).

-an [*annotation_changes_file*] (e.g. *calnotes*)

-gain [*gain*] (e.g. 0.05)

-n [*#_of_windows_to_display*] (e.g. 3)

-rate [*sampling_rate*] (e.g. 500/sec)

Comments: This is the all-purpose program used for displaying data.

....

Program: alternans

Function: This program creates the file *in.*[*data_file*] which contains the programs and their arguments necessary to perform the spectral analysis.

Invocation: User-invoked in step 4 of flow.

Arguments: Interactive.

Comments: This program sets up the file *in.*[*data_file*] so that the majority of the computation intensive analysis can be performed. When **alternans** is invoked, is uses the program **plot.window** to create a graphic display of the second beat of each channel in the data file. A default template window is marked that extends 35 milliseconds on either side of the assigned peak.

After displaying the waveform and window, the user is prompted to assign various shifts for use in the analysis. The first shift asked for is the *baseshift* in milliseconds from the left template marker to the baseline in the right direction (0 millisecond default). It is important that a steady baseline is chosen because it is used as the zero amplitude reference throughout the remainder of the analysis. It is advisable that the user choose a point in the middle of a long section of baseline to avoid error.

Next, the user is asked to verify the window assignment. The window should roughly contain the entire QRS complex. .It is important that this window be properly assigned because it is used as the template in the cross-correlation stage of the analysis, where it is used to refine the fiducial point estimations. In verifying the window assignment, the user is asked to enter the needed *peakshift* in the left direction to center the template on the QRS complex (0 millisecond default). Finally, the user is asked to enter the template size *WINDOW* (70 millisecond default). Figure 3-1 on page 23 presents a sample display used in assigning shifts, and illustrates the associated variables assigned by the user.

Once this user interaction has taken place, the program has all the parameters necessary to create the file *in*.[*data_file*]. It writes calls to the analysis programs (along with their necessary arguments) to *in*.[*data_file*] so that they can be called in sequence. The order of the major programs within *in*.[*data_file*] are as follows:

1. xcnew
2. badbeat
3. segment
4. mkavnew
5. segs
6. spitnew

The user has the option of immediately starting the analysis by running *in*.[*data_file*], or writing it to a batch file for later invocation.

....

Program: xcnew

Function: Performs iterative cross-correlation of assigned template with localized region of each fiducial point. Creates a file of refined fiducial points called *pks*.[*intervention_code*].[*data_file*].

Invocation: Called by *in*.[*data_file*].

Arguments:

-a [*avg_template_file*] (e.g. *avg*.[*intervention_code*].[*data_file*])

-c [*channel_#*] (e.g. -c0 -c1 -c2)

-d [*data_file*] (e.g. *$dpath*/[*data_file*])

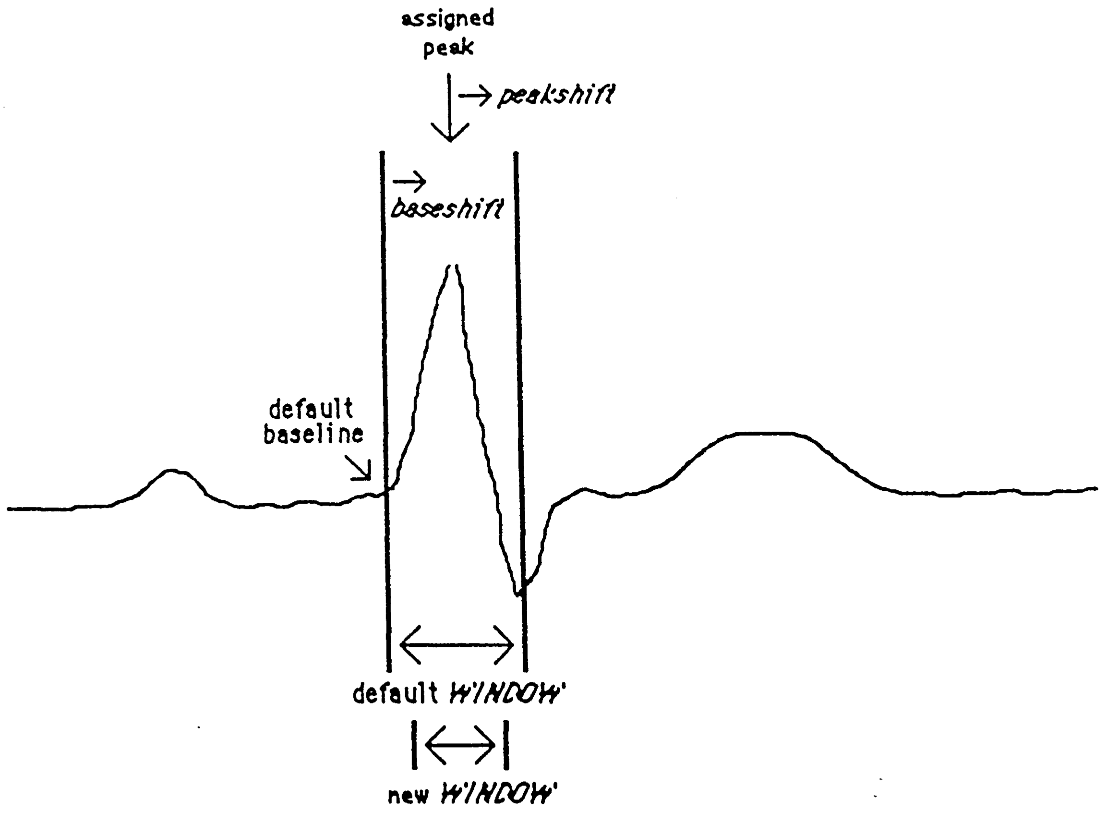


Figure 3-1: Display Used in Assigning Shifts

-g [*gains*] (e.g. -g 1.0 -g 1.0 -g 1.0)

-l [*buffer_shift_left*] (e.g. 0 milliseconds)

-N [*#_of_channels*] (e.g. 3)

-n [*#_of_iterations*] (e.g. 2)

-o [*baseline_shift*] (e.g. 0 milliseconds)

-p [*annots_file*] (e.g. *RR*.[*intervention_code*].[*data_file*])

-s [*max_allow_shift*] (e.g. 35 milliseconds)

-S [*samp_rate*] (e.g. 500/sec)

-w [*window_size*] (e.g. 70 milliseconds)

-x do not attempt to center window

Comments: This program initially creates an averaged template from all of the beats in the data file. It then iteratively looks for maximal cross-correlation between the template and each beat, assigns the best fiducial point, and writes the new refined fiducial points to the file *pks.*[*intervention_code*].[*data_file*]. The refined fiducial points located in this file are used throughout the remainder of the analysis. It also writes to *pks.*[*intervention_code*].[*data_file*] a correlation coefficient for each beat, or a measure of the similarity of each beat to the average waveform. The correlation coefficient provides a means of detecting erroneous beats based on morphology.

....

Program: badbeat

Function: Determines erroneous beats based on RR intervals and morphology.

Invocation: Called by *in.*[*data_file*].

Arguments:
-c [*min_allow_corr*] (e.g. 0.95)

-i [*use_RR_int_criterion?*] (e.g. 1 for yes)

-m [*use_morph_criterion?*] (e.g. 1 for yes)

-p [*annots_file*] (e.g. *pks.*[*intervention_code*].[*data_file*])

-t [*thresh_for_dev_from_typ_RR_int*] (e.g. 25 sample points)

-w [*typ_RR_int*] (e.g. 300 sample points)

Comments: A beat is considered erroneous if its RR interval exceeds the typical RR interval by more than the threshold. A beat is also considered erroneous if its correlation coefficient is less than the minimum allowable correlation. Either or both of these criteria can be used. Be warned that the correlation coefficient criterion is not a very reliable method of determining erroneous beats. Beats of very bad visual morphology (even premature ventricular contractions) often have correlations greater than the present cutoff used of 0.95.

The output of this program is put into *bad.*[*intervention_code*].[*data_file*] which has the following format: column 1 = beat number; column 2 = bad RR interval (0=no, 1=yes); column 3 = bad morphology (0=no, 1=yes); column 4 = bad RR interval and/or bad morphology (0=no, 1=yes).

....

Program: segment

Function: Determines optimal segment of consecutive beats for spectral analysis.

Invocation: Called by *in.*[*data_file*].

Arguments:

-b [*bad_beat_file*] (e.g. *bad.*[*intervention_code*].[*data_file*])

-c [*column_containing_bad_beats*] (e.g. 3)

-l [*segment_length*] (e.g. 128)

Comments: This program is very helpful in identifying the optimal data segment of consecutive beats. It chooses the segment with the least number of erroneous beats (according to the provided criteria) and furnishes the starting beat of that segment. This stage is important because the fewer the number of erroneous beats, the more accurate the final measure of electrical alternans.

....

Program: mkavnew

Function: Creates an average ECG waveform for each of the individual channels and writes it to *int.*[*intervention_code*].[*data_file*].

Invocation: Called by *in.*[*data_file*].

Arguments:

-a [*out_file*] (e.g. *int.*[*intervention_code*].[*data_file*])

-c [*channel_#*] (e.g. -c0 -c1 -c2)

-d [*data_file*] (e.g. *$dpath*/[*data_file*])

-g [*gains*] (e.g. -g 1.0 -g 1.0 -g 1.0)

-l [*buffer_shift_left*] (e.g. 0 milliseconds)

-n [*#_of_beats_to_avg*] (e.g. 10)

-N [*#_of_channels*] (e.g. 3)

-p [*annots_file*] (e.g. *pks.*[*intervention_code*].[*data_file*])

-S [*samp_rate*] (e.g. 500/sec)

-w [*window_size*] (e.g. 70 milliseconds)

Comments: The average ECG complex calculated by this program is not utilized in the quantitative analysis. It is simply used for purposes of displaying and checking the data. The file *int.*[*intervention_code*].[*data_file*] is used by segs to display the average ECG waveform of each channel.

....

Program: segs

Function: Creates a display page (put in the file *seg.*[*intervention_code*].[*data_file*]) provided as output at the end of the analysis.

Invocation: Called by *in.*[*data_file*].

Arguments:

[*intervention_code*].[*data_file*]

[*start_beat_of_opt_segment*] (e.g. 0)

[*end_beat_of_opt_segment*] (e.g. 128)

Comments: This program creates a display page with the following plots: a plot of the average ECG waveform of each channel; a plot of the iterative templates used in xcnew; a plot of the RR intervals with erroneous intervals marked; and a plot of the correlation coefficients with erroneous morphology marked. Note that there is also a command segs_screen which writes the same display to the screen. See section 4.1 beginning on page 31 for further description of the final output.

....

Program: spitnew

Function: Creates matrix file containing all of the beats aligned according to fiducial points, which it puts in *mat*.[*intervention_code*].[*data_file*]. Performs the spectral analysis of the data and puts the calculated spectrum of each point in *big*.[*intervention_code*].[*data_file*].

Invocation: Called by *in*.[*data_file*].

Arguments:

-b [*excluded_beats*] (e.g. -b 10 -b 33 -b 107)

-B [*spc_file*] (e.g. *big*.[*intervention_code*].[*data_file*])

-c [*channel_#*] (e.g. -c0 -c1 -c2)

-d [*data_file*] (e.g. *$dpath/*[*data_file*])

-g [*gains*] (e.g. -g 1.0 -g 1.0 -g 1.0)

-l [*buffer_shift_left*] (e.g. 0 milliseconds)

-m [*matrix_file*] (e.g. *mat*.[*intervention_code*].[*data_file*])

-n [*#_of_beats*] (e.g. 128)

-N [*#_of_channels*] (e.g. 3)

-p [*annots_file*] (e.g. *pks*.[*intervention_code*].[*data_file*])

-S [*samp_rate*] (e.g. 500/sec)

-w [*window_size*] (e.g. 70 milliseconds)

Comments: This program performs the actual spectral analysis. The spectrum of each data point of the entire beat interval is put into *big*.[*intervention_code*].[*data_file*] and summed later during the final stage of the analysis. The matrix file is used during the final stage of the analysis to create the vector average ECG for assignment of the QRS, ST, and T segment intervals.

....

Program: final

Function: Prompts user to assign QRS, ST, and T segment intervals of the vector magnitude ECG. Sums spectra for each segment and puts numerical values of spectra in *spc*.[*intervention_code*].[*data_file*], puts various measures of EA in *res*.[*intervention_code*].[*data_file*], and puts graphic output in *p*.[*intervention_code*].[*data_file*] for later plotting.

Invocation: User-invoked in step 5 of flow.

Arguments: Interactive.

Comments: Essentially, this program collects the current results of the analysis and puts the results in specified files. When invoked, the program displays the vector magnitude ECG and prompts the user to assign intervals over which to sum the spectra. It then uses the program sumnew to sum the spectra. An example of appropriate intervals is shown in figure 3-2 on page 29. For further description of the output files created by this program, see section 4.1 beginning on page 31.

....

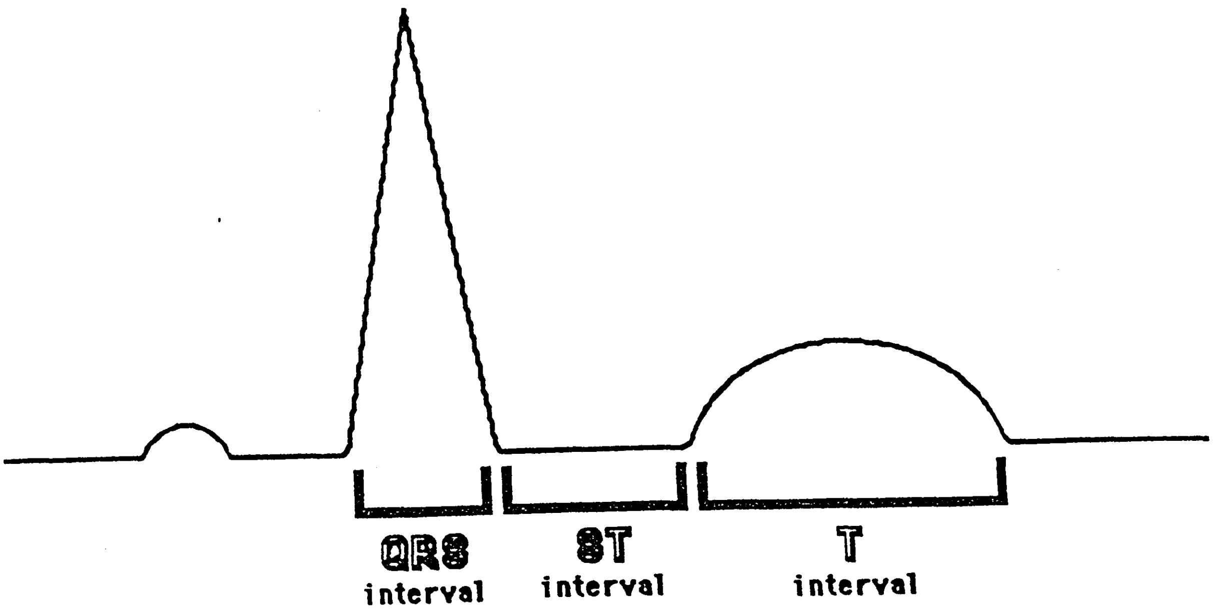


Figure 3-2: Proper Assignment of Intervals

Program: sumnew

Function: Sums the spectra of each complex.

Invocation: Invoked by final.

Arguments:

-b [*start_point*] (e.g. [*start_QRS*])

-B [*blanking_region*] (e.g. blank out pacing artifact)

-e [*end_point*] (e.g. [*end_QRS*])

-f [*spc_file*] (e.g. *big*.[*intervention_code*].[*data_file*])

-l [*#_beats_in_spc*] (e.g. 128)

-o [*out_file*] (e.g. *spc*.[*intervention_code*].[*data_file*])

-s [*sum_last_?_points*] (e.g. we usually only use last point)

Comments: final uses this to sum the spectra for the QRS, ST, and T complexes.

Program: results

Function: Sends output files *res*.[*intervention_code*].[*data_file*], *seg*.[*intervention_code*].[*data_file*], and *p*.[*intervention_code*].[*data_file*] to the laser printer for plotting.

Invocation: User-invoked in step 6 of flow.

Arguments: [*data_file*]

Comments: See section 4.1 beginning on page 31 for a total description of the results provided at the end of the analysis.

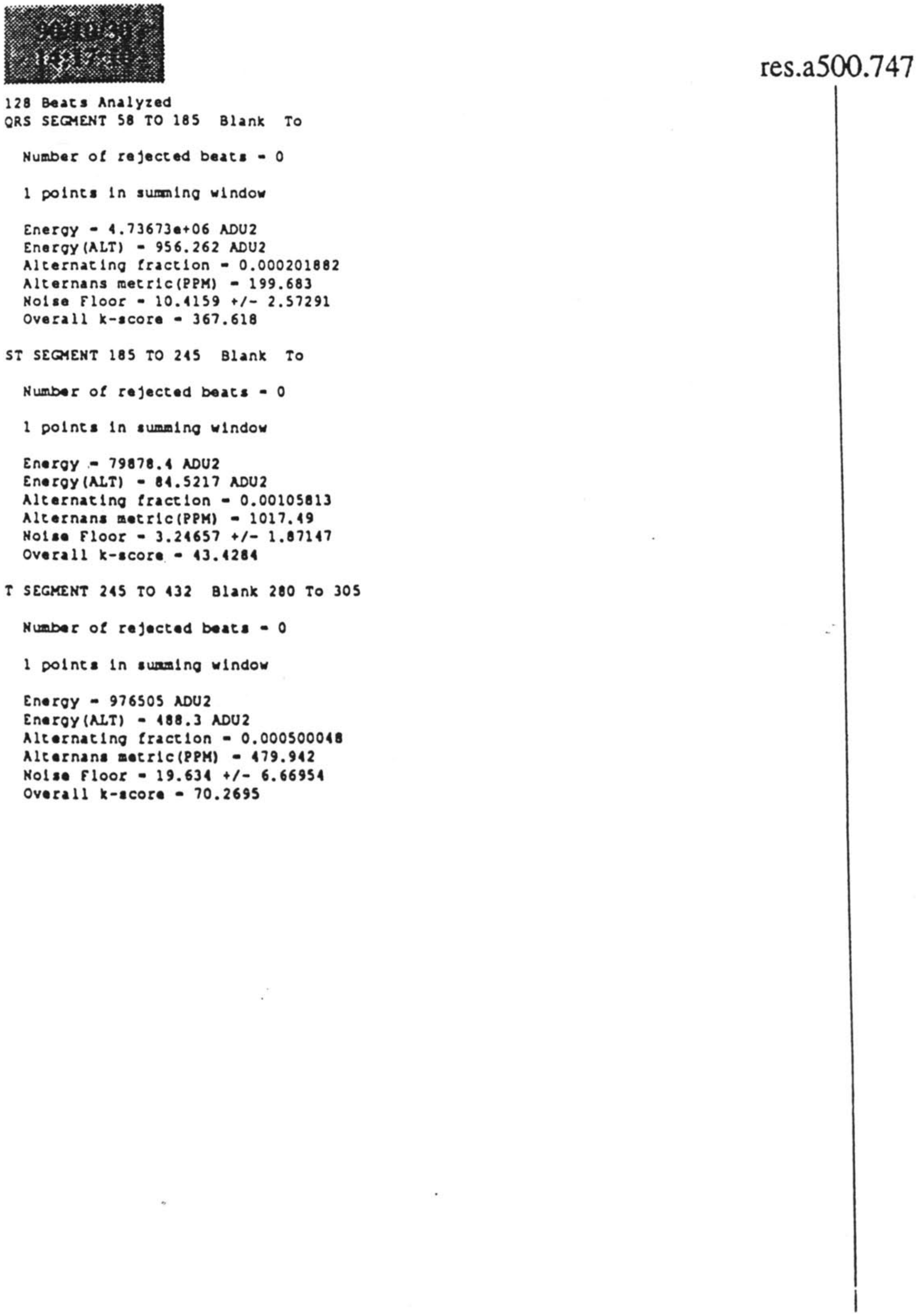
Chapter 4

Results of Analysis

4.1 Results Provided as Output

As described in the last chapter, there are several output files created by the analysis system during the course of the analysis. The program results is invoked to produce a hardcopy of the output files which aid the user in evaluating the results and their validity. Figures 4-1, 4-2, and 4-3 on the following pages show the output pages produced by the analysis system for a sample data file.

Figure 4-1: Numerical Results (Page 1 of Output)



[illegible]

res.a500.747

```
128 Beats Analyzed
QRS SEGMENT 58 TO 185  Blank  To

  Number of rejected beats - 0

  1 points in summing window

  Energy - 4.73673e+06 ADU2
  Energy(ALT) - 956.262 ADU2
  Alternating fraction - 0.000201882
  Alternans metric(PPM) - 199.683
  Noise Floor - 10.4159 +/- 2.57291
  Overall k-score - 367.618

ST SEGMENT 185 TO 245  Blank  To

  Number of rejected beats - 0

  1 points in summing window

  Energy - 79878.4 ADU2
  Energy(ALT) - 84.5217 ADU2
  Alternating fraction - 0.00105813
  Alternans metric(PPM) - 1017.49
  Noise Floor - 3.24657 +/- 1.87147
  Overall k-score - 43.4284

T SEGMENT 245 TO 432  Blank 280 To 305

  Number of rejected beats - 0

  1 points in summing window

  Energy - 976505 ADU2
  Energy(ALT) - 488.3 ADU2
  Alternating fraction - 0.000500048
  Alternans metric(PPM) - 479.942
  Noise Floor - 19.634 +/- 6.66954
  Overall k-score - 70.2695
```

Figure 4-2: Plots for Validation of Analysis Process (Page 2 of Output)

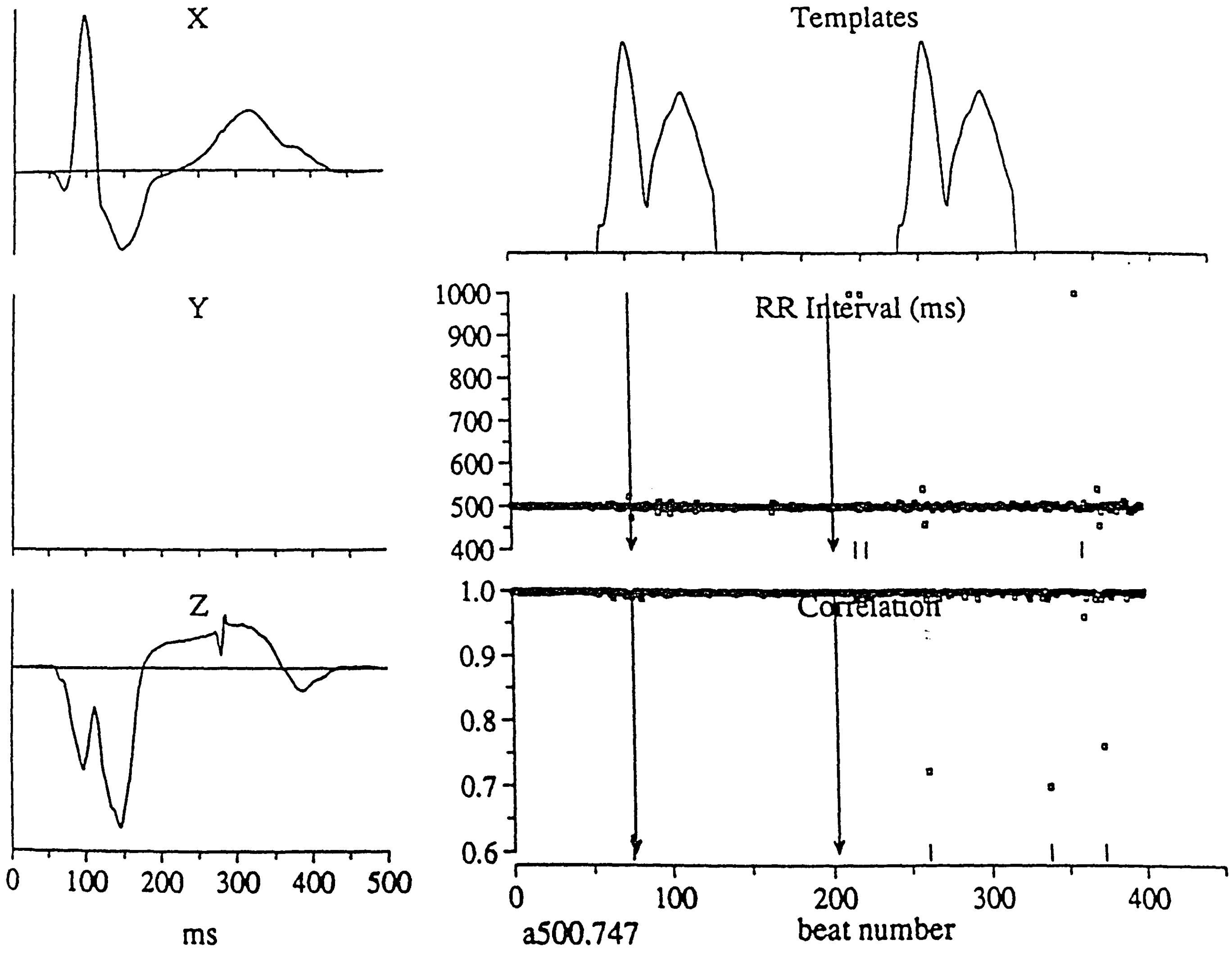
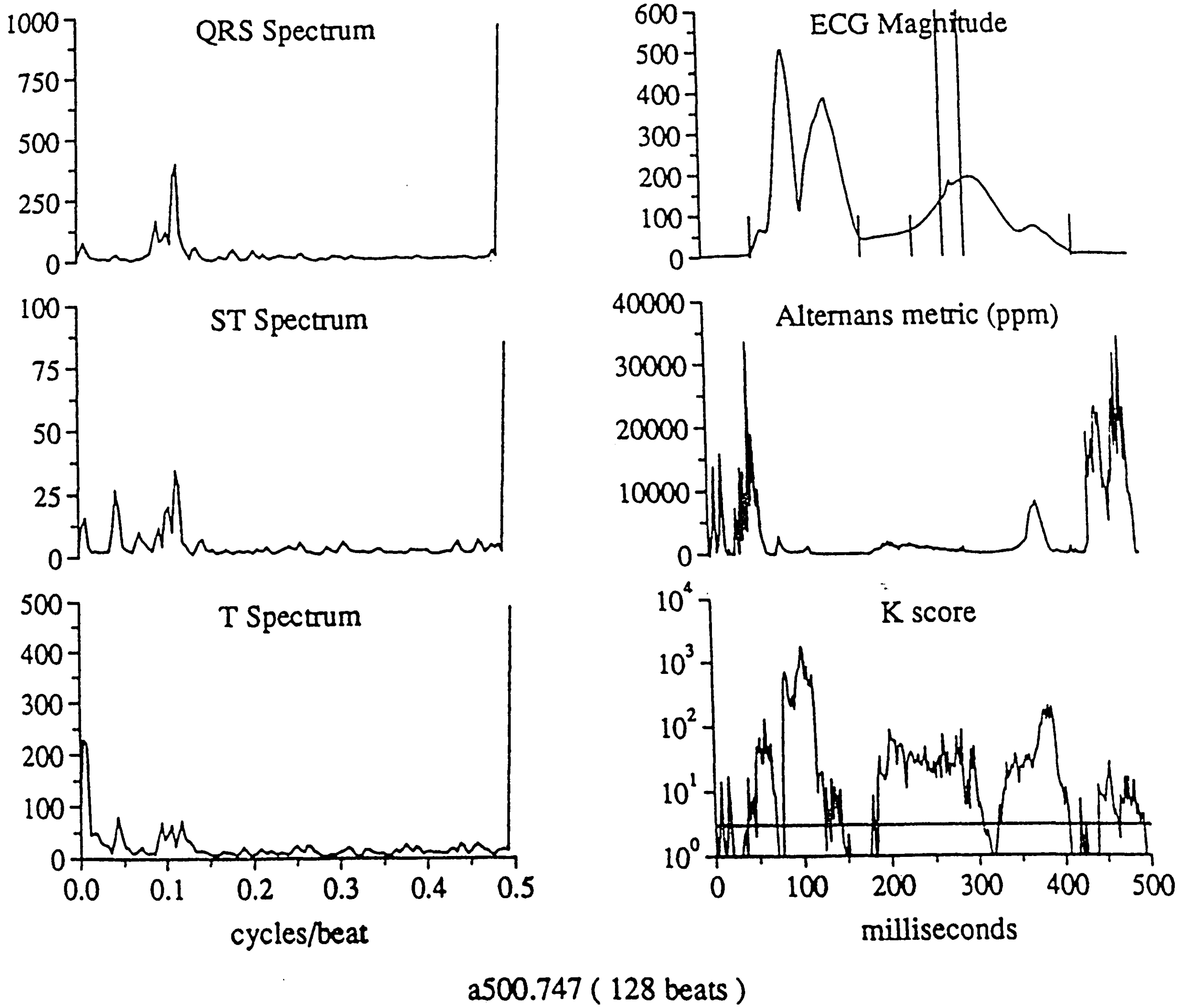


Figure 4-3: Spectral Results (Page 3 of Output)



4.1.1 Numerical Results

The first page of the output provides specific parameters used in the analysis of the data file and statistical measures of EA. First, it states the number of beats used in the spectral analysis (typically 128). It also shows the number of beats rejected (i.e. erroneous beats) of the 128 beats used and states the number of points used in the summing window. The summing window is the number of points at the end of the spectrum which are considered representative of beat-to-beat alternation. Typically, the last point (the alternans point) representing the exact frequency of 0.5 cycles/beat is used. However, it is possible to use the last two or more points in the summing window in case phase resetting has occurred.

The first page also provides individual statistics for the QRS, ST, and T intervals. It gives the segment (in milliseconds) of the vector average ECG which was chosen by the user to represent each interval. It then shows the corresponding blanking interval, if it exists. (Note that the segment intervals minus the blanking intervals are used in calculating the alternans statistics of each interval.) Finally, the first output page provides the statistics important for assessing the degree of electrical alternans present in each segment.

The first value provided is the energy of the interval (given in squared analog-to-digital units, or ADU^2). This energy is calculated by summing the squares of the amplitudes of each point in the vector magnitude ECG interval. The second value is the alternans energy (also given in ADU^2), which represents the amplitude of the alternans point. The alternating fraction, which appears third, is merely the alternans energy normalized by the total energy of the segment. While the alternans energy relates the "total alternans energy," the alternating fraction shows the degree of alternans present in relation to the total energy of the signal.

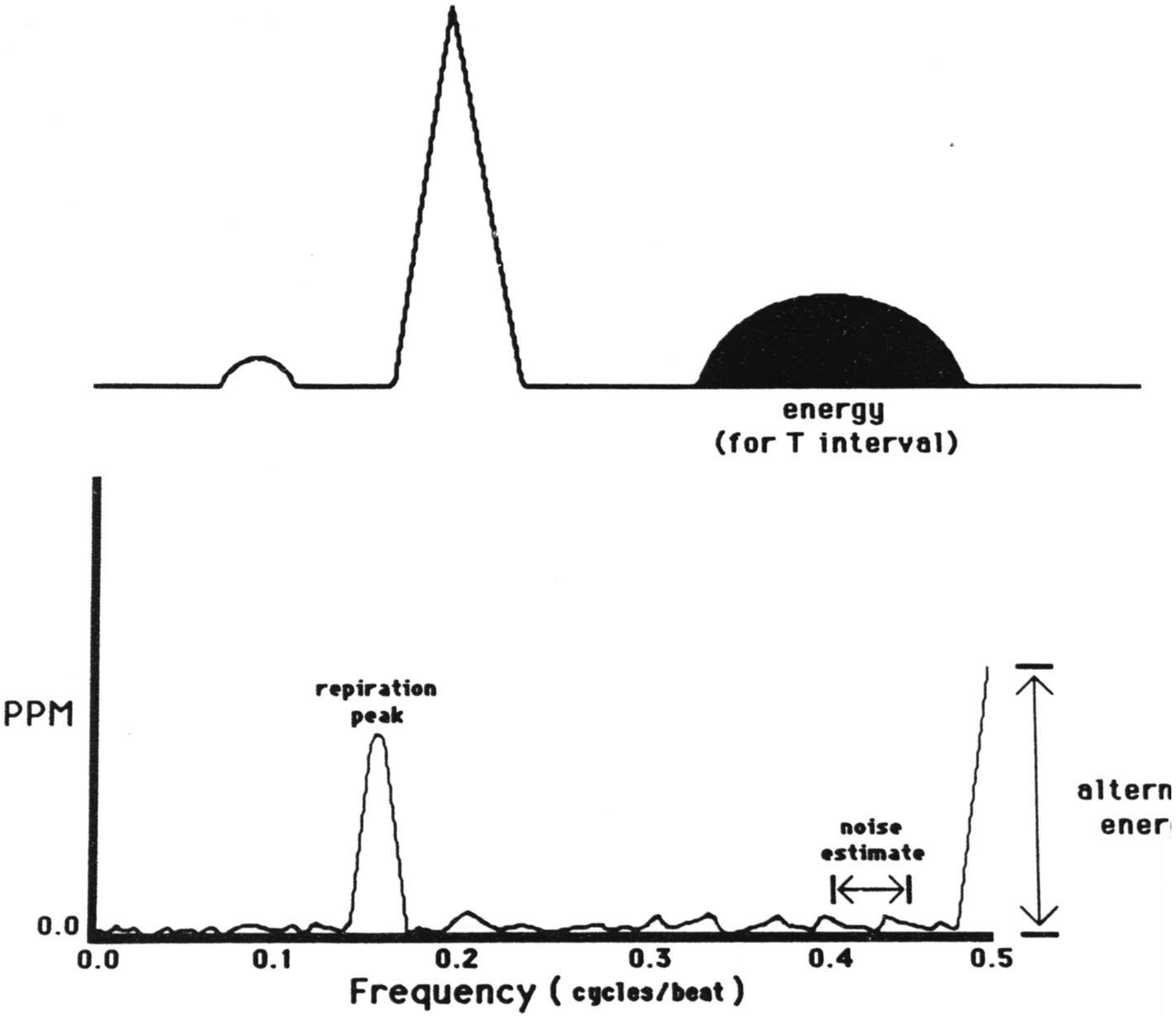
The last three statistics compare the alternans energy to an estimate of noise. The noise of an adjacent spectral band is calculated by a construction of the mean and standard deviation of eight frequency samples in that band. The adjacent band used is located eight points to the left of the alternans point, and extends four points in either direction. The noise mean and standard deviation are related in the fifth statistic, which is labeled the noise floor (given in ADU^2). The alternans metric (the fourth statistic, given in parts per million) represents the amount by which the alternans energy exceeds the mean noise power estimate, normalized by the energy of the average waveform segment. Thus, the alternans metric relates the amount of alternation minus the effects of noise. The significance of the alternation is determined by the last statistic, the K score. The K score is calculated by dividing the difference of the alternans energy and noise mean by the noise standard deviation. The alternation in morphology is judged significant if the K score exceeds a value of three. Please refer to figure 4-4 on page 37 for a pictorial representation of the statistics found on the first page of the output.

While these numerical results are provided in ADU, the results can be converted to microvolts. In order to do this, multiply the results given in ADU by (1000 microvolts)/(400 ADU), or 2.5 microvolts/ADU. To convert ADU^2 to microvolts^2, multiply by $2.5^2 = 6.25$ (microvolts/ADU)^2.

4.1.2 Plots for Validation of Analysis Process

The second page of the output gives several plots which are useful in tracing the course of the analysis. The three plots on the left side of the page represent the average ECG waveforms (averaged over all of the beats) of the individual X, Y, and Z leads versus time in milliseconds. The top plot on the right side shows the iterative templates used in the cross-correlation and fiducial point refinement stage of the analysis. The templates should contain the major events of the QRS complex. Also, the differences between the templates after two iterations should be essentially imperceptible, showing that two

Figure 4-4: Statistical Measures of Electrical Alternans



$$\text{alternating fraction} = \frac{\text{alternans energy}}{\text{energy}}$$

$$\text{alternans metric} = \frac{\text{alternans energy} - \text{noise mean}}{\text{energy}}$$

$$\text{K score} = \frac{\text{alternans energy} - \text{noise mean}}{\text{noise standard deviation}}$$

iterations through the cross-correlation stage are sufficient. The last two plots on the right side show the RR intervals (in milliseconds) versus beat number and the cross-correlation coefficients versus beat number. Each erroneous RR interval and each erroneous correlation (as determined by the analysis system) is denoted by a small line marker. Additionally, the first beat and the last beat of the optimal segment chosen by the system are marked with arrows. These two graphs allow the user to verify that the segment chosen by the system is reasonable, or that modification is necessary.

4.1.3 Spectral Results

The third and final page of the output gives several plots which graphically relate the results of the spectral analysis. The three plots on the left side of the page show the determined total spectra for each segment (given in ADU^2 versus frequency in cycles per beat). The final point (or points) of the spectra represent alternans. The top plot on the right side of the page relates the average vector magnitude ECG (given in ADU versus milliseconds). The QRS, ST, T, and blanking interval boundaries are denoted by line markers, allowing the user to verify that the assignments are reasonable. The second plot on the right side shows the alternans metric of each sample point in the entire beat interval (given in parts per million versus milliseconds). Finally, the last plot relates the K score (versus milliseconds) of each sample point in the entire beat interval. This plot is logarithmic and has a line marker at a K score of three, indicating significance. These last two plots allow the user to examine the trend of alternation found over an average beat of the data segment being analyzed.

4.2 Evaluation of Quality of Results

There are two types of commonly encountered situations which make the user question the quality of the results. These two questionable situations occur when (1) erroneous beats are detected in the optimal segment by the system and (2) erroneous beats that should have been detected by the system are not recognized as erroneous. This section of the thesis will explain the best course of action for the user to take in these two situations.

4.2.1 Effects of Replacing Erroneous Beats

Although the analysis system utilizes an algorithm to determine the optimal segment (the segment with the least number of erroneous beats), it may not be possible to select a segment of the proper length that contains no erroneous beats. Thus, it is to the user's advantage to know the ultimate effects on the results of the analysis of replacing erroneous beats with the average waveform. This will allow the user to decide if the data is of good enough quality to be included in the study.

To determine the ramifications of various numbers of erroneous beats, a study was conducted. First, a control data set was chosen which had no erroneous beats and high levels of alternans. Then, various numbers of erroneous beats were interjected into the file *bad*.[*intervention_code*].[*data_file*] to simulate flawed optimal segments, and the analysis was completed in each case. The results of the study appear in table 4-I on page 40. The table presents the alternans metric and K score of the QRS, ST, and T complexes resulting from differing numbers of total erroneous beats.

From the table, it is apparent that differing numbers of erroneous beats do alter the measured alternans metric and K score. However, the effects are only gradual as the number of erroneous beats is increased. Additionally, the changes in the two measures do not appear to follow a predictable trend. One would conclude from this study that it is not

Table 4-I: Effects of Replacing Measured Beats with Template Beats

# of Bad Beats	Complex	Alternans Metric	K Score
0	QRS	3.63	4.62
	ST	1012	29.77
	T	1218	432
1	QRS	3.88	4.34
	ST	950	34.7
	T	1169	289
3	QRS	2.74	4.35
	ST	908	21.2
	T	1144	235
5	QRS	3.24	5.34
	ST	977	25.9
	T	1168	199
7	QRS	3.91	5.33
	ST	885	31.4
	T	1120	161
10	QRS	2.19	6.11
	ST	843	18.8
	T	1020	342

significantly detrimental to replace an erroneous beat with the average waveform, but it is best to obtain the segment with fewest erroneous beats for greatest accuracy of the final results. It appears that there is no obvious cutoff for the number of erroneous beats that would indicate a data set that should be rejected.

4.2.2 Effects of Not Replacing Erroneous Beats

Sometimes, erroneous beats which are obviously flawed upon visual inspection are not rejected according to the system's algorithm. This situation is readily detectable on the RR Interval and Correlation versus beat number plots. Erroneous beats missing detection are indicated by far outlying points which are unmarked as erroneous beats. When a user encounters this type of situation, he should examine these trouble points with the display program. (Recall that the sample point numbers of outlying points can be determined with the program **outlier** for easy location of erroneous beats within the display program.) If the beat containing the trouble fiducial point obviously looks flawed, the user should then alter the file *in.[intervention_code].[data_file]* so that the erroneous beat will be recognized as erroneous by the system and consequently replaced by the average waveform during spectral analysis. There are several simple ways to change the file *in.[intervention_code].[data_file]* in order to accomplish this goal. The two most obvious ways to do this are (1) change the threshold value for the deviation from the typical RR interval (the -t switch to **badbeat**) or (2) change the minimum allowable correlation (the -c switch to **badbeat**).

An example of an occurrence where an erroneous beat was not rejected according to the default parameters of the system can be found in figure 4-5 on page 42. In this situation, a beat of obviously undesirable visual morphology passed as an acceptable beat and was included in the optimal 128 beat segment (notice the one beat of low correlation within the segment). With this erroneous beat included in the analysis, a spectrum with a large amount of noise and little alternation resulted. In order to correct this, the minimum allowable correlation was increased from 0.95 to 0.98 in the file *in.[intervention_code].[data_file]* so that the beat was recognized as erroneous by the system. The results after this second iteration (with all of the other parameters left the same) can be found in figure 4-6 on page 43. As can be seen, the beat of undesirable morphology was not included in the spectral analysis, and a much nicer spectrum resulted. Much of the noise was eliminated from the spectrum and a higher degree of alternation became apparent.

It can be concluded from the two types of situations described in this section that it is not very harmful to the results if erroneous beats are replaced by the average beat, but it is

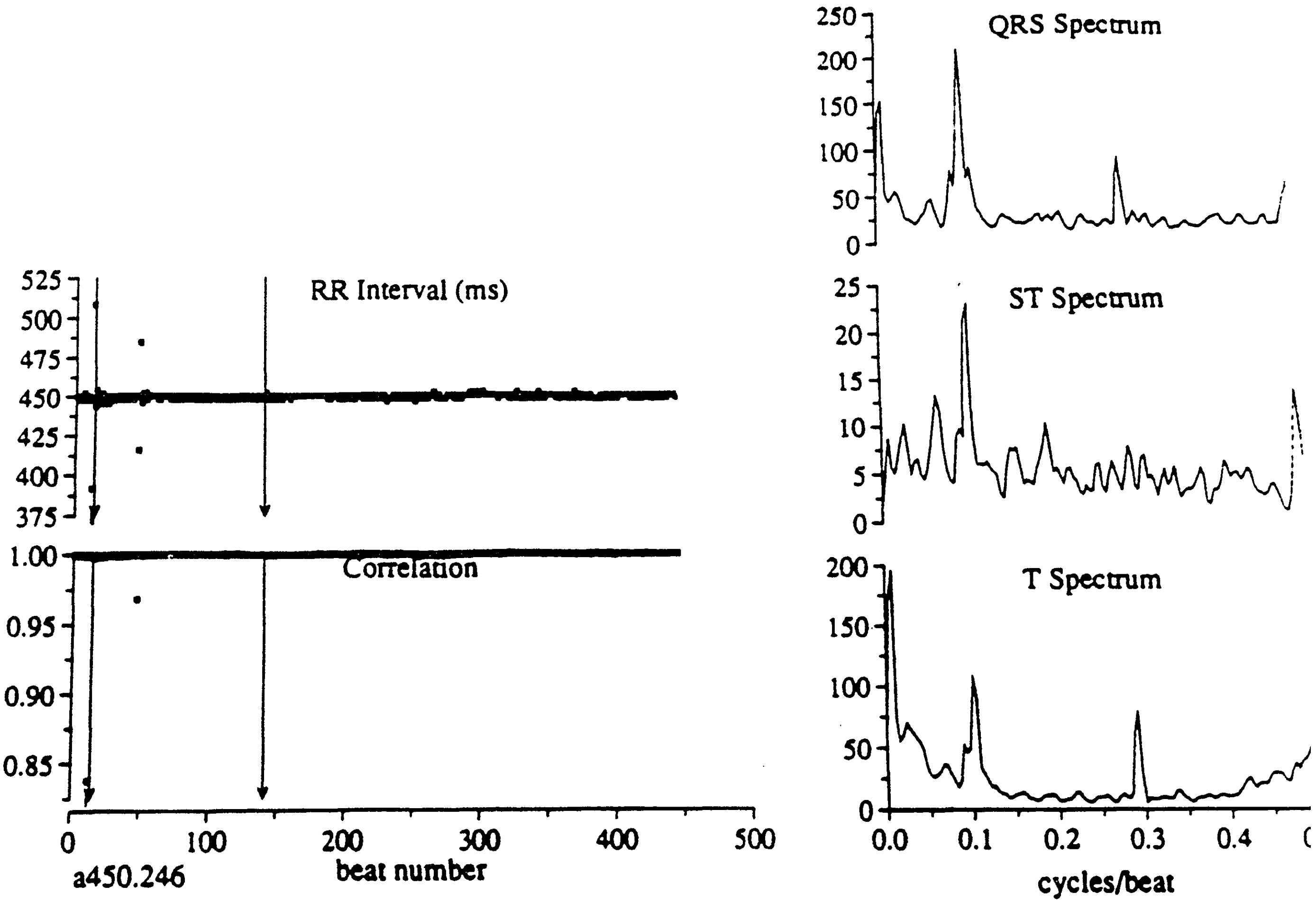


Figure 4-5: First Iteration of Example

extremely damaging if an obvious erroneous beat is not detected and replaced. The situations in this section also show the advantages of various features of the analysis system, including: the need of having a program to choose the optimal segment so that the fewest possible erroneous beats are included in the analysis; the utility of producing hardcopy plots for the user to view the course of the analysis and determine if iteration is necessary; and the benefit of generating the batch file *in.*[*intervention_code*].[*data_file*] so that changes for iteration can be easily made. Note that there are also other situations

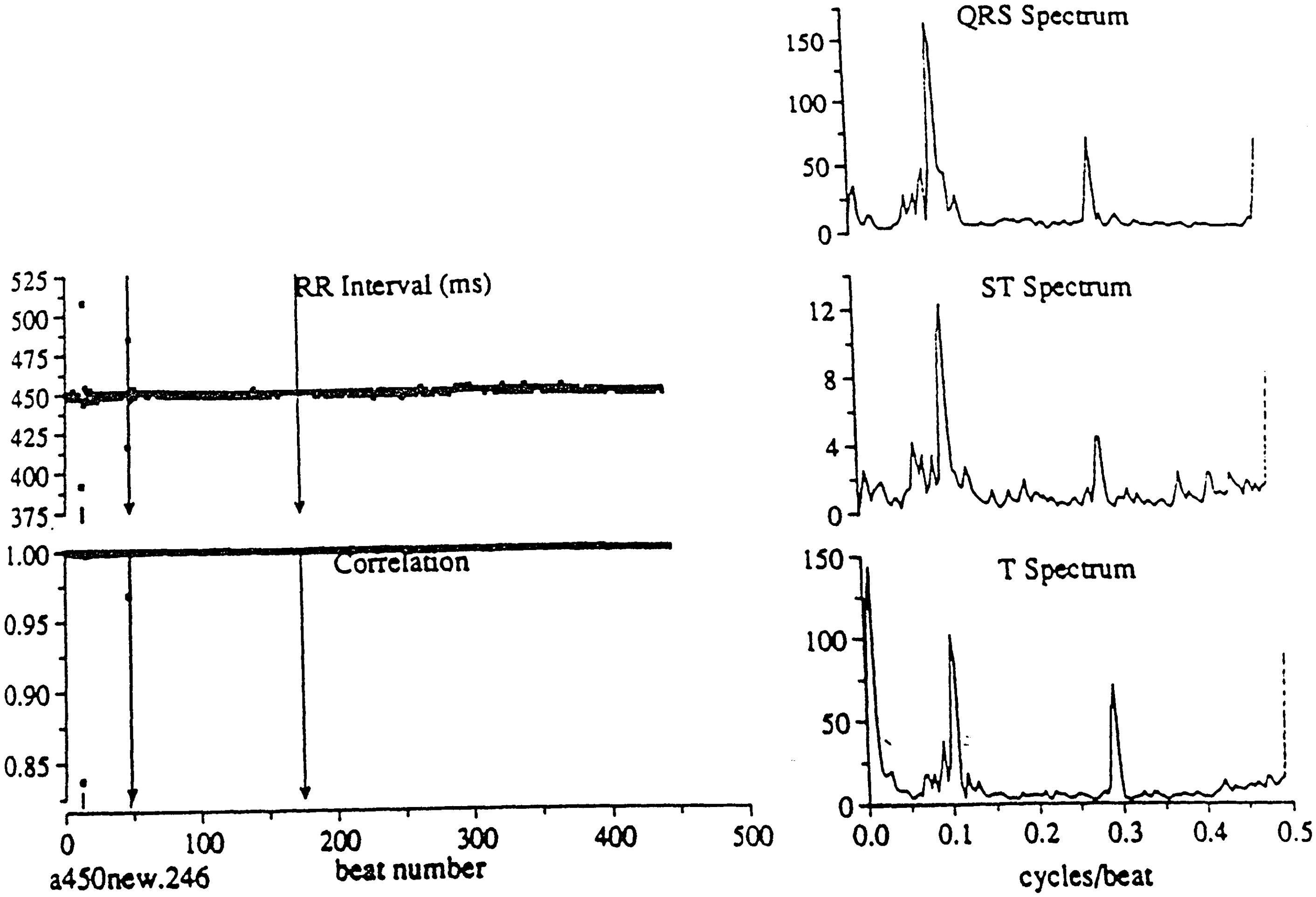


Figure 4-6: Second Iteration of Example

indicating the necessity for iteration, all of which can be determined from the hardcopy plots. These instances include: noticeably misrepresented templates that do not include the major features of the QRS complex, or templates that are too large; an obviously misrepresented baseline; and improperly chosen QRS, ST, T, and blanking intervals.

Chapter 5

Discussion and Conclusions

5.1 Significance of Research

Ventricular arrythmias are one of the most devastating types of organic heart disease. Since it is important to have a means of identifying those at greatest risk of suffering such an event, it would be extremely useful to have a non-invasive method for screening large populations and ascertaining individual risk. Currently, the automated system of this thesis is specifically tailored to determine the degree of EA in ECG recordings of human subjects. Consequently, this system can be used to ascertain the significance of EA in clinical arrythmias.

There is extensive clinical literature supporting the notion that EA is a precursor to ventricular arrythmia. Previous work suggests that EA is closely associated with dispersion of recovery and quantitatively varies with vulnerability to arrythmia [Rosenbaum, et al 90]. This led us to hypothesize that EA measurements may provide a way to describe ventricular susceptibility to reentrant dysrythmias, and set a theoretical basis for our clinical project. If our project indicates that there is indeed a predictable correlation between EA and susceptibility to arrythmias, it could potentially result in the development of a non-invasive technique for quantifying a patient's cardiac electrical stability. The analysis system developed for this thesis will provide a semi-automated and sensitive technique for measuring EA in a clinical environment.

Additionally, this particular project will provide a way to assess the universality of alternans type dynamics and the relation of EA to chaos theory. Recent study of the chaotic behavior of nonlinear systems has described a "universal" period-doubling route to chaos, suggesting that these systems approach chaos through a common pathway. In other words, these systems respond to continually increasing subharmonics of the driving frequency until they lapse into chaos. EA could represent the first step in this route to chaos, since it occurs at the first subharmonic. If EA is proven by our research to be a reliable precursor to arrythmia, period-doubling behavior would be supported as a property of the pre-arrythmic ventricle, and the heart would be indicated as obeying the universal laws of non-linear system dynamics. The analysis system will also be important in determining the validity of this hypothesis.

5.2 Future Potential of Automated System

Currently we have several additional specific future plans for the system. First of all, we will customize the analysis system to examine the dynamic origins of the signal average late potential. Late potentials manifest on the signal average QRS complex have been linked to elevated risk to ventricular arrythmias [Kuchar, et al 86, Kuchar, et al 90]. These signals originate from regions of slow conduction and create a situation for reentry. We hypothesize that there exists both a dynamic (temporal dispersion of repolarization) and a static (delayed conduction) basis of late potentials, each of which may contribute to the genesis of arrythmia. We wish to measure the waveform oscillations which comprise the "averaged" late potential and determine if alternans type periodicities contribute to the existence of late potentials.

We will also utilize the system to try to gain insight into the underlying cellular mechanism of electrical alternans and its relation to the occurrence of arrythmia. We have shown experimentally that the occurrence of EA may be due to alternating action potential durations within injured zones of heart tissue that promote spatial dispersion of recovery [Rosenbaum, et.al 90]. This spatial dispersion may suggest a mechanism by which alternans are associated with arrythmia vulnerability. Specialized percutaneous endocardial catheters now permit the recording of monophasic action potentials in humans. Using the analysis system, we plan to measure periodic fluctuations of action potentials recorded by these instruments. This may improve our understanding of the cellular basis for EA as well as factors which predispose to lethal arrythmias.

5.3 Future Technical Improvements

There are several future improvements planned for the analysis system that will make it more automated and complete. The present algorithm utilizing cross-correlation techniques is not entirely reliable at detecting ectopic beats and other beats that should be excluded from the spectral analysis. Thus, we plan to incorporate a more robust algorithm for identification of ectopic ECG complexes. Additionally, we hope to improve the present fiducial point detection algorithm, since it is not completely flawless at identifying all fiducial points. Ideally, improvements of our current algorithms will eventually eliminate the need for displaying the data or iterating through the analysis more than once. They should also increase the accuracy of our measures of EA by more reliably eliminating erroneous beats.

Another improvement we wish to incorporate is an automatic baseline detector. This will increase the automation of our analysis by further decreasing the need for user interaction. We feel that these improvements are worthwhile modifications. While these changes to the system are already planned, more ideas to improve the system will probably be conceived. Over time, this analysis system should develop into a further automated and efficient system.

5.4 Fulfillment of Goals

The automated analysis system developed for this thesis has proved to be very useful in our current analysis. It has provided a means of efficiently analyzing the large amount of data that has been collected. The analysis process is now relatively easy to learn and perform such that persons with little previous knowledge of the details of EA and cardiac electrophysiology can quickly become proficient at analyzing data for the presence of EA. The manual included in this thesis is designed to facilitate familiarization with and use of the analysis system.

Presently, we feel that the results obtained with the system are very accurate measures of EA. We have designed the system to accept a variable number of channels of ECG data and choose the optimal data segment of 128 consecutive beats according to our defined algorithm. The parameters to this algorithm can be tailored by the user if, based on the provided output, the default parameters do not appear to provide the best results. Additionally, a multi-dimensional spectral technique is utilized by our system which is very sensitive and attempts to eliminate unwanted noise generated from extraneous sources. It is our belief that the spectral technique provides a very accurate measure of beat-to-beat variation. Finally, the results are furnished graphically so that the steps taken by the automated system can be easily verified.

The system has already been utilized to analyze the data collected for our first project in an attempt to determine the reliability of EA as a marker of vulnerability to arrythmia. The system proved helpful and efficient in performing the analysis. We currently have additional plans to use the system to (1) study the dynamic origins of signal averaged late potentials and (2) research the mechanism of electrical alternans in man. It appears that the system is adequately versatile and can readily be made to accommodate new modes of analysis. In conclusion, we feel that we have accomplished both (1) our first goal of developing an efficient and accurate automated system for analyzing ECG data for EA and (2) our second goal of designing a universal system that can be adapted to perform various types of analysis.

Acknowledgements

This work was supported by NASA grant NAGW-988 and NIH grant #1R01HL39291-02, and was performed in Richard Cohen's laboratory of the Harvard - MIT Division of Health Sciences and Technology.

I have many people whom I would like to thank for their contributions to making this thesis materialize. I owe thanks to Danny Kaplan, Joe Smith, Paul Albrecht, Mike Perrott, Zhihoa Yin, and Dave Rosenbaum for the use of their software in the analysis system. Also, much thanks to Professor Richard Cohen for acting as my thesis supervisor and providing a laboratory for me to work in for three years.

In addition, I would like to thank Dave Rosenbaum for acting as a close advisor, teacher, and friend during my introduction to medical research. I also appreciate him allowing me to use his research proposal as an aid in writing my thesis. I owe a great deal of gratitude to him for helping me plan, perform, and write my thesis, as well as for sparking a lifetime interest in cardiology.

Finally, I would like to thank my parents for all of the support that they have given me.

References

[Hamburger, et al 36]
Hamburger WW, Katz LN, Saphir L.
Electrical alternans, Clinical study with report of 2 necropsies.
J.A.M.A. (106:902), 1936.

[Hellerstein, et al 50]
Hellerstein JK, Leibow IM.
Electrical alternation in experimental coronary arteryocclusion.
Am J Physiol (160:366), 1950.

[Hering 08] Hering HE.
Munch. med.
Wschr (55:1417), 1908.

[Kuchar, et al 86] Kuchar DL, Thorburn CW, Sammel NL.
Late potentials after myocardial infarction: Natural history and prognostic significance.
Circulation (74:1280), 1986.

[Kuchar, et al 90] Kuchar D, Rosenbaum DS, Smith JM, Albrecht P, Cohen RJ, Ruskin JN.
Late potentials in a canine myocardial infarction model: Correlation with ventricular arrythmias.
1990.
In Press.

[Mines 13] Mines GR.
On functional analysis by the action of electrolytes.
J Physiol (46:188), 1913.

[Nakashima, et al 78]
Nakashima M, Hashimoto H, Kanamaru M, Nagaya H, Hashizume M, Oshi H.
Experimental studies and clinical report on the electrical alternans of ST segment during myocardial ischemia.
Jpn Heart J (19:396-408), 1978.

[Rosenbaum, et al 90]
Rosenbaum DS, Ruskin JN, Jackson L, Cohen RJ, Salama G.
Electrical alternans as a precursor to ventricular arrythmias: A cellular mechanism.
Journal of American College of Cardiology , 1990.

[Ruskin, et al 83] Ruskin JN, Schoenfeld MR, Garan H.
Role of electrophysiologic techniques in the selection of anti-arrthymic drug regimens for ventricular arrythmias.
Am J Cardiol (52:41C), 1983.

[Russel, et al 79] Russel DC, Smith HJ, Oliver MD.
Transmembrane potential changes and ventricular fibrillation during repetitive myocardial ischemia in the dog.
Br Heart J (42:88), 1979.

[Smith 86] Smith JM.
The Stochastic nature of cardiac electrical instability: theory and experiment.
PhD thesis, Massachusetts Institute of Technology, 1986.

[Smith, et al 84a] Smith JM, Ritzenberg AL, Cohen RJ.
Percolation theory and cardiac conduction.
IEEE Comp Soc. , 1984.

[Smith, et al 84b] Smith JM, Cohen RJ.
Simple finite-element model accounts for wide range of ventricular dysrhythmias.
Proc Nat Acad Sci USA (81:233), 1984.

[Smith, et al 88] Smith JM, Clancy EA, Valeri R, Ruskin JN, Cohen RJ.
Electrical alternans and cardiac electrical instability.
Circulation (77:110-121), 1988.

Appendix A
Commented Code

The original commented code comprising the software portion of this thesis follows, and includes:

1. convert
2. cals
3. display
4. outlier
5. alternans
6. plot.window
7. badbeat
8. segment
9. segs
10. final
11. results

badbeat and segment were written in C programming language, while the rest of the programs were written in Shell programming language.

```
#*************************************************************
#                                                            *
#                          convert                           *
#                                                            *
#*************************************************************
#
# This file converts the IBM data to the needed format for
# analysis and creates a header file.

CHAN="3"
SAMP="500"
DB="/db/yin/twa/header"
if [ -z "$dpath" ]
then
  echo "ERROR  dpath does not exist."
  exit 1
fi
echo "dpath = $dpath"
path="/db/yin/twa/record"

if [ -z "$1" ]
then
  echo -n "Enter number of data file to convert (e.g. 246). "
  read file1
else
  file1=$1
fi

if [ -z "$2" ]
then
  echo -n "Enter name of peaks file to convert to an RR file \
(e.g. pk246). "
  read file2
else
  file2=$2
fi

if [ -z "$3" ]
then
  echo -n "Enter intervention code (e.g. a600.246). "
  read file3
else
  file3=$3
fi

if [ -z "$4" ]
then
  echo -n "Enter number of channels. "
  read CHAN
else
  CHAN=$4
fi


cat >> $DB/header.$file1 <<!
$file1
1
$dpath/$file1
```

```
0
$SAMP
16      3       12      0
0
channels $CHAN
!

dd conv=swab  < $file1 > tmp.$$
mv tmp.$$ $file1
mv $file1 $dpath/$file1

mkdir $path/rec.$file1
/home/pa/bin/math $file2 -o’0,c1,c1-oc1,0,0,0’ -P | tail +2 > \
  $path/rec.$file1/RR.$file3
rm $file2
```

```
#****************************************************************
#                                                               *
#                               cals                            *
#                                                               *
#****************************************************************
#
# This program determines the calibrations of a data file
# to be put into its header file.
# The program requires that the user enter two annotations to data.
# The first corresponds to the cal pulse base and the second
# to the cal pulse peak.

DB="/db/yin/twa/header"

if [ -z "$dpath" ]
then
  echo "ERROR  dpath does not exist."
  exit 1
fi
echo "dpath = $dpath"

if [ -z "$1" ]
then
  echo -n "Enter data file for which to find cals. "
  read data
else
  data=$1
fi

cp $dpath/$data data.$$
CHAN=`awk '/channels/ {print $2}' $DB/header.$data`

echo "You must enter 1 annotation for base of cal and one"
echo "annotation for peak of cal."
#echo "Hit return to continue.."

show $dpath/$data {2,$CHAN} -gain 0.05 -an calnotes

dget calnotes % -P > tmp.$$

tbase=`awk 'NR==1 {print $1}' tmp.$$`
tpeak=`awk 'NR==2 {print $1}' tmp.$$`

a=0
oldcal=""
while [ "$a" -lt "$CHAN" ]
do

        peak=`dget :$tpeak,$tpeak data.$$ {2,$CHAN} $a -P`
        base=`dget :$tbase,$tbase data.$$ {2,$CHAN} $a -P`

        cal=`calc "400/abs($peak - $base)" `

        G="$oldcal $cal"

        a=`calc "$a + 1"`
        oldcal="$G"
```

```
done

echo "gains: $G" >> $DB/header.$data
rm *.$$
rm calnotes
```

```
#*****************************************************************
#                                                               *
#                          display                              *
#                                                               *
#*****************************************************************
#
#In current form does not allow one to delete annotations from file
#define variables.
#$1 = data file to use
#$2 = peak file to use (and potentially edit)
data="$1"
peaks="$2"

if [ -z "$1" ]
then
echo "Enter data file (e.g. 586)"
read data
echo "Enter peaks annotation file (e.g. pks.a600.586)"
read peaks
fi

DB="/db/yin/twa/header"
CHAN=`awk '/channels/ {print $2}' $DB/header.$data`

if [ -z "$dpath" ]
then
  echo "ERROR  dpath does not exist."
  exit 1
fi
echo "dpath = $dpath"
cp $dpath/$data tmp.$$
chmod +w tmp.$$

if [ -n "$peaks" ]
then
#convert RR peak file to annotation file format
awk ' {print $2,"2","65","0" }' $peaks  > oldnotes.$$
#display data with annotations
show tmp.$$ {2,$CHAN}  -gain .05  -an newnotes.$$ / oldnotes.$$
mergeann oldnotes.$$ newnotes.$$

else

#display data without annotations
show tmp.$$ {2,$CHAN}  -gain 0.05
rm *.$$
exit 1
fi

#convert annotation file format back to RR peak file format
echo "Do you want to modify $peaks file ? n or [y] "
read foo
if [  "$foo" = "n" ]
then
rm *.$$
exit 1
fi
```

```
awk ' {
         if (NR > 1) {
             print "0",$1,$1-lastR,"0","0","0"
         }
         lastR=$1
      }' oldnotes.$$ >  $peaks
rm  *.$$
```

```
#****************************************************************
#                                                               *
#                          outlier                              *
#                                                               *
#****************************************************************
#
#Displays bad RR intervals.
#Finds bad RR intervals from annotations file in RR or pks format.
#$1 = annotation file
#$2 = allowable slop (in sample points)
peaks="$1"
slop="$2"

if [ -z "$2" ]
then
echo "Enter annotation file with peaks (e.g. pks.a600.123)"
read peaks
echo "Enter allowable slop (in sample points)"
read slop
fi


M=`RRstats $peaks | awk ' { print $2 }'`
high=`calc "$M + $slop" `
low=`calc "$M - $slop" `
awk  '$3 > '$high' || $3 < '$low' { print NR - 1,$2,$3 }' $peaks   | more
```

```
#********************************************************************
#                                                                   *
#                        alternans                                  *
#                                                                   *
#********************************************************************
#
#DESCRIPTION:
#This version of xcor uses "xcnew" which is a computationally faster
#version of xcnew which uses interpolation to .2 ms (not .1ms)
#and permits data files that contain more than 3 channels.
#This script will run "xcnew" program in batch mode after prompting the
#user for all necessary data.  The procedure creates a separate
#shell script called "$wpath/in.$file"

#warning : can not run more than one in.$file program simultaneously
#so this program allows one to call multiple xcnew runs is series

#NOTE: When assigning parameters for the xcnew program "WINDOW",
#"SHIFT","peakshift", it is important to keep the following guidelines
#in mind:
#The xcnew program uses a default of 35 ms for the baseline (i.e. 35 ms
#to the left of the RAW peak). The user can modify both the basline and
#the peak (which becomes the refined peak location) estimate
#independently using the -o baseshift and -l peakshift
#switches, respectively. Since the mkavnew and spitnew programs both, by
#default, use a point 75 ms to the left of the refined peak as
#a baseline, then the shift required to maintain the peak-to-baseline
#relationship specified by the user is given by the equation;
#BSHIFT=-( (75 - offset) + peakshift + baseshift).
#In this way, the baseline used by the spitnew program will be identical
#to the refined baseline specified by the user prior to running xcnew.
#
#baseshift= distance in ms from unmodified raw peak to ECG baseline.
#peakshift=distance in ms from unmodified raw peak to modified raw peak.
#WINDOW=width in ms of template window
#SHIFT=maximal allowable shift of window (taken as 0.5 * WINDOW)
#BSHIFT=additional offset required (in ms) for mkavg and spitnew programs
#       to insure that the baseline selection is identical to that chosen
#       by user on raw data (when he used peakshift & baseshift).
#offset=default baseline point used by xcnew taken at the start of
#template.
#
#Note that the ECG analysis resulting from mkavnew and spitnew start
#at the point of the baseline.  Thus one could check the correctness
#of the baseline selection just by examining the first point of
#columns 0,2,4,6 of int.$ID or column zero of mat.$ID.
#Also note that the programs expect all shifts to be
# expressed in ms (not points).
#
#CALIBRATIONS: Note that cal program has divided atd/ 1 mv from cal
#signal into 400 atd/mv.  Therefore, to convert calculated data (in atd)
#back to microvolts you must multiply the result by 1000/400 = 2.5
#To convert squared atd units to microvolts squared, you must multiply
#by 6.25.
#
#
#EXPECT AS INPUT     1. A RAW PEAK FILE IN RR.$ID FORMAT
#                    2. RAW DATA FILE
```

```
#PROGRAMS CALLED:
#                  mkavnew
#                  xcnew
#                  plt
#                  segs
#
#INPUT FILES:      datafile (raw data file)
#                  RR.intervention.idcode (raw peaks)
#
#OUTPUT FILES:     int.intervention.idcode (signal averaged ecg
#                  x,y,z vector)
#                  in.$file (shell created and executed to run
#                  multiple xcnew as batch)
#                  out.$file (standard output of in.$file)
#
#                  bad.$ID contains bad beats
#
#
#*****************************************************************
#-----------------------------------------------------------------
#Checking for data file to be analyzed
#-----------------------------------------------------------------

#Define shell variables
DB="/db/yin/twa/header"
BEATS="128"
#Number of iterations for cross-correlation
ITERS="2"
rev=`expr $ITERS - 1 `
if [ -z "$dpath" ]
then
  echo "ERROR  dpath does not exist."
  exit 1
fi
echo "dpath = $dpath"

export DB dpath

wpath=`pwd`
if [ -z "$1" ]
then
  echo -n "Enter data file to analyze "
  read file
else
  file="$1"
fi

until [  -f "$dpath/$file" ]
do
  echo "Can't find $file in $dpath "
  echo "Your choices of data files are "
  ls $dpath
  echo -n "Enter data file to analyze "
  read file
done
export file
```

```
#_____________________________________________________________
#Sampling rate
#_____________________________________________________________
SAMP=`awk 'NR==5 {print $0}' $DB/header.$file`

#_____________________________________________________________
#Number of channels to analyze
#_____________________________________________________________
channels=`awk '/channels/ {print $2}' $DB/header.$file`
if [ -z "$channels" ]
then
  echo "Can't find channel # in header file "
  echo -n "Enter number of data channels [3] "
  read channels
  if [ -z "$channels" ]
  then
    channels="3"
  fi
fi

#set channel calls to variable C
a=1
C="-c0"
while [ "$a" -lt "$channels" ]
do
        C="$C -c$a"
        a=`calc "$a + 1" `
done

#_____________________________________________________________
#SET GAINS FOR STUDY
#_____________________________________________________________
OLDGAIN=""
G=""
G1=`awk '/gains/ {print $2}' $DB/header.$file`
G2=`awk '/gains/ {print $3}' $DB/header.$file`
G3=`awk '/gains/ {print $4}' $DB/header.$file`
for GAIN in $G1 $G2 $G3
do
        G="$OLDGAIN -g $GAIN"
        OLDGAIN="$G"
done

#-------------------------------------------------------------------
#
echo -n "Enter type of raw peak annotation (e.g. a600 ) "
read intervention
#
#----------
#Assign ID code

ID="$intervention.$file"
#----------
#Allign raw peak and baseline
#----------
plot.window RR.$ID $file

echo -n "Shift for baseline estimate (+ = right , - = left) [0 ms] "
```

```
read baseshift

[ -n "$baseshift" ] || baseshift="0";

echo -n "Shift raw peak for centering template ( + = left , - = \
  right) [0 ms] "
read peakshift

[ -n "$peakshift" ] || peakshift="0";

echo -n 'Window size[70 ms]? '
read WINDOW
[ -n "$WINDOW" ] || WINDOW="70";

#Maximum allowable shift for template to cross-correlate
SHIFT=`calc "int( $WINDOW/2 )" `

#Default baseline (offset) used by xcnew program is at start of
#template window.  i.e. WINDOW/2 to the left of the peak.
offset=`calc "int( $WINDOW/2 )" `

#note -x switch in xcnew will prevent a reloction of raw peak
#estimate to the highest point within the template window.

#IF YOU WANT ALTERNANS METRIC AND K SCORE TO BE CALCULATED FROM LAST
#4 POINTS OF SPECTRUM (RATHER THAN LAST POINT E.G. NYQUEST FREQUENCY
#INSERT -s4 as switch to spit program.
#THIS OPTION IS USEFULL WHEN THE ALTERNANS ENERGY IS SHIFTED OVER
#TO ADJACENT FREQUENCY BANDS SECONDARY TO PHASE RESETTING OF BEATS

chmod 777 in.$file

cat >> in.$file <<!

xcnew -aavg.$ID  -d$dpath/$file $G  -n$ITERS -pRR.$ID -s$SHIFT \
  -w$WINDOW -l $peakshift -o $baseshift -x $C  -N $channels -S $SAMP
get :1, pks.rev$rev % -P > pks.$ID
rm pks.rev*
WIDTH=\`RRstats pks.$ID | awk '{print \$2}' \`
badbeat -w \$WIDTH -c 0.95 -i 1 -m 1 -p pks.$ID -t 25   > bad.$ID
start=\` segment  -b bad.$ID -c 3 -l $BEATS \`
end=\`calc "\$start + $BEATS - 1" \`
awk '(NR - 1) >= '\$start' { print \$0 }' pks.$ID > bestpks.$$
beatout=""
for F in \`awk '\$1 >= '\$start' && \$1 <= '\$end' && \$4 == 1 \
  {print ( \$1 - '\$start' ) }'  bad.$ID  \`
do
beatout="\$beatout -b \$F "
done
WIDTH=\`calc "int( (\$WIDTH/$SAMP) * 1000 )" \`
BSHIFT=\`calc "-( (75 - $offset) + $baseshift + $peakshift)"\`
mkavnew -aint.$ID -d$dpath/$file $G -n 10 -pbestpks.$$ -l \$BSHIFT \
  -w\$WIDTH $C -N $channels -S $SAMP
PTERM=lw
export PTERM
segs  $ID \$start \$end > seg.$ID
spitnew -n$BEATS -l\$BSHIFT $G   -d$dpath/$file -pbestpks.$$ \
  -mmat.$ID $C  -N $channels -Bbig.$ID -w \$WIDTH \$beatout -S $SAMP
```

```
rm *.$$

!

#-----------------------------------------------------------------
#user can either run shell at this point or have the file in.$file
#saved for a batch shell run from /data/dave/batch
chmod 555 in.$file

echo -n "Do you want to run now or abort for later? [run]/abort "
read foo
if [ "$foo" = "abort" ]
then
  echo "cd $wpath" >> /db/yin/twa/batch
  echo "in.$file" >> /db/yin/twa/batch
  chmod +x /db/yin/twa/batch
  exit 1
fi

#------------------------------------------------------------------
#execute the shell just created called in.$file
#
(  in.$file 2> ERROR
   chmod  777  in.$file
   echo "alternans analysis finished...."
   bell ) &
```

```
#********************************************************************
#                                                                   *
#                          PLOT.WINDOW                              *
#                                                                   *
#********************************************************************
#
#This script will input RR file of raw RR intervals ($1) and a raw
#data file ($2) and generate a plot of three beats (the first three
#RR peaks) shown with the default window displayed (+/- 35 ms from
#raw peak).  This can be used then to determine if the baseline
#estimate used to subtract the DC componant from each signal is OK.
#OUTPUT FILES:
#               PLT.tmp = temporary file containing raw data
#
#SAMPLING RATE
SAMP=`awk 'NR==5 {print $0}' $DB/header.$2`
#

dir=`pwd`
peak1=`get :1,1 $dir/$1 1 -P `
peak2=`get :2,2 $dir/$1 1 -P `
peak3=`get :3,3 $dir/$1 1 -P `

peak1m=`calc "($peak1-(20 * $SAMP/1000))/(60 * $SAMP)"`
peak3m=`calc "($peak3+(20 * $SAMP/1000))/(60 * $SAMP)"`

winstart=`calc "($peak2 - 35*($SAMP/1000))/$SAMP"`
#winend=`calc "($peak2 + 35*($SAMP/1000))/$SAMP"`

chkpath $2

/home/pa/bin/filter -t $2  -c0 -c1 -c2 -b $peak1m  -e $peak3m -d5 -r5 \
  -s$SAMP  -T  | /home/pa/bin/math % -o"(c0 - $winstart),c1,c2,c3" -D \
   > $dir/plt.tmp

plt $dir/plt.tmp  0 1 -W .1 .05   .9 .2833 -sty  -c 0 0 0 -  \
  -c .07 0 .07 -
plt $dir/plt.tmp  0 2  -W .1 .3833 .9 .6166 -steyn  -c 0 0 0 - \
  -c .07 0 .07 -
plt $dir/plt.tmp  0 3   -W .1 .7166 .9 .9499 -steyn -c 0 0 0 - \
  -c .07 0 .07 -

/bin/rm $dir/plt.tmp
```

```
/************************************************************/
/*                                                          */
/*                       badbeat.c                          */
/*                                                          */
/************************************************************/

#include "matrix_functions.h"
#include "/home/pa/util/utils.c"

  main(int argc, char *argv[])
{

  char ctmp, *pksname;
  matrix *BAD_BEATS, *A, *TEMP;
  double mean_rr, max_dev, cmin;
  int interval=1, morph=1, i;

  while( (ctmp = option(&argc, argv)) != 0)
    {
      switch(ctmp)
        {

        case 'c': /* minimum allowable correlation */
          cmin = atof(opt_argv());
          break;

        case 'i': /* use RR intervals to determine bad beats */
          interval = atoi(opt_argv());
          break;

        case 'm': /* use morphology to determine bad beats */
          morph = atoi(opt_argv());
          break;

        case 'p': /* name of peaks file to be analyzed */
          pksname = opt_argv();
          break;

        case 't': /* maximum allowable threshold for deviation */
                  /* from typical RR interval */
          max_dev = atof(opt_argv());
          break;

        case 'w': /* width of typical RR interval */
          mean_rr = atof(opt_argv());
          break;

        case 'h': /* help */
          printf("\n\n badbeat: determines bad beats\n");
          printf("\n\t-c (minimum allowable correlation)");
          printf("\n\t-i (use RR intervals to determine bad beats
            (1=yes, 0=no: 1 is default))");
          printf("\n\t-m (use morphology to determine bad beats
            (1=yes, 0=no: 1 is default))");
          printf("\n\t-p (name of peaks file to be analyzed)");
          printf("\n\t-t (maximum allowable threshold for deviation
            from typical RR interval)");
          printf("\n\t-w (width of typical RR interval)");
        printf("\n\t *Note that all operations and variables are
          in points!!!");
        printf("\n\t *A beat's RR interval is considered bad if
          it deviates from the typical");
        printf("\n\t\t interval by more than the threshold
          value.");
        printf("\n\t *A beat's morphology is considered bad if it
          correlates < 95 percent.");
        printf("\n\t\t away from the mean.\n");
        exit(0);

      default:
        printf("\n ILLEGAL OPTION %c\n", ctmp);
        exit(0);
      }
  }


/* Matrix BAD_BEATS: col 0 will contain beat number; col 1 will */
/* indicate bad RR intervals; col 2 will indicate bad */
/* morphology; col 3 will indicate total bad beat (0 for good */
/* beat, a 1 for bad beat). */
/* Matrix A: used for vector operations. */

BAD_BEATS = init("BAD_BEATS");
A = init("A");
TEMP = init("TEMP");

/* Makes A matrix out of column 2 (RR intervals) of pks file. */
fill_cols(2,pksname,A);

s_to_m(0,A->rows,4,BAD_BEATS);

/* Determines bad intervals */
if (interval)
  {

    /* Determines difference between each point and the mean */
    /* and puts into A. */
    sub(A,s_to_m(mean_rr,A->rows,A->cols,TEMP),A);

    /* Assigns a 1 to col 1 if RR int is bad. */
    /* Assigns a 0 to col 1 if RR int is good. */
    for (i = 0; i < A->rows; i++)
      {
        if (fabs(A->element[i][0]) >= max_dev)
          BAD_BEATS->element[i][1] = 1.0;
      }
  }

/* Determines bad morphology. */
if (morph)
  {

    /* Makes A matrix out of column 4 (correlation coefficients) */
    /* of pks file. */
    fill_cols(4,pksname,A);
```

```
      /* Determines minimum correlation coefficient within 3 x */
      /* standard deviation of coeff. */
      /*      cmin = mean(A) - 3*sqrt(var(A)); */

      /* Assigns a 1 to col 2 if RR int is bad. */
      /* Assigns a 0 to col 2 if RR int is good. */
      for (i = 0; i < A->rows; i++)
        {
          if (A->element[i][0] < cmin)
            BAD_BEATS->element[i][2] = 1.0;
        }
    }

  /* Combines desired criterea (RR int and/or morphology) to */
  /* obtain total bad beats. */

  /* Assigns beat number to col 0 of BAD_BEATS. */
  /* Assigns a 1 to col 3 if beat is bad. */
  /* Assigns a 0 to col 3 if beat is good. */
  for (i = 0; i < A->rows; i++)
    {
      BAD_BEATS->element[i][0] = i;
      if ((BAD_BEATS->element[i][1] == 1.0) ||
         (BAD_BEATS->element[i][2] == 1.0))
      BAD_BEATS->element[i][3] = 1.0;
    }

  /* Output the BAD_BEATS file. */

  print(BAD_BEATS);

}
```

```
/*******************************************************/
/*                                                     */
/*                     segment.c                       */
/*                                                     */
/*******************************************************/

#include "matrix_functions.h"
#include "/home/pa/util/utils.c"

  main(int argc, char *argv[])
{
  FILE *fp;
  char ctmp, *bad_name;
  matrix *A, *BAD_PER_SECTION;
  double length, min;
  int i, j, column=3, best_section_num;

  while((ctmp = option(&argc, argv)) != 0)
    {
      switch(ctmp)
        {
        case 'b': /* file containg beat numbers and bad beat */
                  /* markers */
          bad_name = opt_argv();
          break;

        case 'c': /* column of input file containing bad beat */
                  /* markers */
          column = atoi(opt_argv());
          break;

        case 'l': /* length of segment (in beats) desired */
          length = atoi(opt_argv());
          break;

        case 'h': /* help */
          printf("\n\n segment: determines best section of beats to
            use\n");
          printf("\n\t-b (file containing bad beats markers where:");
          printf("\n\t      0=good_beat and 1=bad_beat)");
          printf("\n\t-c (column number of input file containing
            bad beat markers (3 is default))");
          printf("\n\t-l (length of segment desired in beats)\n");
          exit(0);

        default:
          printf("\n ILLEGAL OPTION %c\n",ctmp);
          exit(0);
        }
    }

  /* Matrix BAD_PER_SECTION: will contain number of bad beats in */
  /* section starting count at current beat number. */
  /* Matrix A: will contain a 0 for bad beat, a 1 for good beat. */
  /* It is made out of column 3 of the input file. */

  A = init("A");
  BAD_PER_SECTION = init("BAD_PER_SECTION");
  /* Makes A matrix out of specified column of the bad beats */
  /* file. */
  fill_cols(column,bad_name,A);

  conform_matrix(A->rows-length+1,A->cols,BAD_PER_SECTION);

  /* Make sure there are at least the required number of beats. */
  if (A->rows < length)
    {
      printf("There are only %d beats.\n", A->rows);
      exit(0);
    }

  /* Determines number of bad beats in segment length starting at */
  /* current beat. */
  for (i = 0; (i + length) <= A->rows; i++)
    {
      BAD_PER_SECTION->element[i][0] = 0.0;
      for (j = i; j < i + length; j++)
        BAD_PER_SECTION->element[i][0] += A->element[j][0];
    }

  /* Determines best section. */
  min = BAD_PER_SECTION->element[0][0];
  best_section_num = 0;
  for (i = 1; i < BAD_PER_SECTION->rows; i++)
    {
      if (min == 0.0)
        break;
      if (BAD_PER_SECTION->element[i][0] < min)
        {
          min = BAD_PER_SECTION->element[i][0];
          best_section_num = i;
        }
    }

  /* Output start beat of best section. */

  printf("%d\n", best_section_num);

}
```

```
#************************************************************
#                                                          *
#                        segs                              *
#                                                          *
#************************************************************
#
# This program creates plots for verificatrion of the analysis
# process.

datum="$1"
start="$2"
end="$3"
dir=`pwd`

  if [ -z "$1" ]
  then
  echo -n "enter datum to plot  e.g. v600.123  "
  read datum
  fi

  if [ -z "$2" ]
  then
  echo -n "enter start beat of best segment  "
  read start
  fi

  if [ -z "$2" ]
  then
  echo -n "enter end beat of best segment  "
  read end
  fi

length=`rows int.$datum`

  if [ ! -f "int.$datum" ]
  then
  echo "segs can't find int.$datum in $dir"
  exit 1
  fi

  if [ ! -f "avg.$datum" ]
  then
  echo "segs can't find avg.$datum in $dir"
  exit 1
  fi

#sampling rate
DFNM=`echo $datum |cut -d. -f2`
SAMP=`awk 'NR==5 {print $0}' $DB/header.$DFNM`
#
length=`calc "$length*1000/$SAMP"`

/home/pa/bin/math int.$datum 0 -o"row*1000/$SAMP,c0" | plt 0 1 \
  -W .02 .68 .32 .93 -sytn -C 0 0 0 -  -c 0 0 - 0  -F"
hl .5 .95 c 1
X
xa 0 - - - - 0"
```

```
/home/pa/bin/math int.$datum 2 -o"row*1000/$SAMP,c0" |plt 0 1 \
  -W .02 .39 .32 .64 -styne -C 0 0 0 - -c 0 0 - 0  -F"
hl .5 .95 c 1
Y
xa 0 - - - - 0"

/home/pa/bin/math int.$datum 4 -o"row*1000/$SAMP,c0" |plt 0 1 \
  -W .02 .1 .32 .35 -stye -C 0 0  0 -  -c 0 0 - 0  -F"
hl .5 .95 c 1
Z
hl .5 -.3 c 1
ms
hl 1.5 -.3 c 1
$datum"

/home/pa/bin/math avg.$datum 0 -o"row*1000/$SAMP,c0" |plt 0 1 \
  -W .42 .68 .98 .93 -steny -F"
hl .5 .95 c 1
Templates"

/home/pa/bin/math pks.$datum 2 -o"c0*1000/$SAMP" | plt 0 \
  -W .42 .39 .98 .64 -stex -ps" " -F"
a $start - $start -
a $end - $end -
hl .5 .95 c 1
RR Interval (ms)"

awk '  {
         if ($2 > 0) { print  $1,"1" }
       else
      { print $1,"100" }   } ' bad.$datum |
 plt 0 1 -W .42 .39 .98 .64 -steyx -ps"|" -Y 1 2

plt pks.$datum 4   -W .42 .10 .98 .35 -ste -ps" "  -F"
a $start - $start -
a $end - $end -
xo 0.05
#xa 0 - 10 - 5 -
hl .5 .95 c 1
Correlation
hl .5 -.28 c 1
beat number"

awk '  {
         if ($3 > 0) { print  $1,"1" }
       else
      { print $1,"100" }   } ' bad.$datum |
plt 0 1 -W .42 .10 .98 .35 -steyx -ps"|" -Y 1 2
```

```
#******************************************************************
#                                                                 *
#                             final                               *
#                                                                 *
#******************************************************************
#
#This version will iteratively call for big.* files in working
#directory to be input to program

BEAT="128"

[ -f FINAL ] &&   rm FINAL

for BIGFILE in `ls big.*`
do
  echo -n "Do you want to analyze $BIGFILE [y] "
  read foo
  [ "$foo" = "n" ] && continue

ID1=`echo "$BIGFILE" | cut -d. -f2 `
file=`echo "$BIGFILE" | cut -d. -f3 `
ID=$ID1.$file
#BEAT=`awk ' /'$ID'.beats/ { print $2 } ' $DB/header.$file `
#echo BEAT is $BEAT

#if [ -z "$BEAT" ]
#then
#echo -n "Enter number of beats in $ID [256] "
#read BEAT
#[ -z "$BEAT" ] && BEAT="256"
#fi

echo "Using $BEAT beats "

#Sampling rate
SAMP=`awk 'NR==5 {print $0}' $DB/header.$file`
echo "Sampling rate is" $SAMP

#--------------------------------------------------------------------
#plot out ecg from matrix file to take segment measurements from

/home/pa/bin/math mat.$ID 0 -o"row*1000/$SAMP,c0" |plt 0 1 -st   -F"
hl .5 1.1 c 1
Vector Magnitude $ID
xa 0 - 10 -  5 -"
#--------------------------------------------------------------------
echo -n "Enter start of qrs (ms) "
read s_qrs
echo -n "Enter end of qrs (ms) "
read e_qrs

S_qrs=`calc "$s_qrs*$SAMP/1000"`
E_qrs=`calc "$e_qrs*$SAMP/1000"`

echo -n "Enter start of BLANKING interval for qrs (ms) "
read bstr_qrs
echo -n "Enter end of BLANKING interval for qrs (ms) "
read bend_qrs
```

```
blank_qrs=""
if [ "$bstr_qrs" != "" ]
then
  Bstr_qrs=`calc "$bstr_qrs*$SAMP/1000"`
  if [ "$bend_qrs" != "" ]
  then
    Bend_qrs=`calc "$bend_qrs*$SAMP/1000"`
    blank_qrs="-B $Bstr_qrs $Bend_qrs"
  fi
fi


echo -n "Enter start of st (ms) "
read s_st
echo -n "Enter end of st (ms) "
read e_st

S_st=`calc "$s_st*$SAMP/1000"`
E_st=`calc "$e_st*$SAMP/1000"`

echo -n "Enter start of BLANKING interval for st (ms) "
read bstr_st
echo -n "Enter end of BLANKING interval for st (ms) "
read bend_st

blank_st=""
if [ "$bstr_st" != "" ]
then
  Bstr_st=`calc "$bstr_st*$SAMP/1000"`
  if [ "$bend_st" != "" ]
  then
    Bend_st=`calc "$bend_st*$SAMP/1000"`
    blank_st="-B $Bstr_st $Bend_st"
  fi
fi


echo -n "Enter start of t (ms) "
read s_t
echo -n "Enter end of t (ms) "
read e_t

S_t=`calc "$s_t*$SAMP/1000"`
E_t=`calc "$e_t*$SAMP/1000"`

echo -n "Enter start of BLANKING interval for t (ms) "
read bstr_t
echo -n "Enter end of BLANKING interval for t (ms) "
read bend_t

blank_t=""
if [ "$bstr_t" != "" ]
then
  Bstr_t=`calc "$bstr_t*$SAMP/1000"`
  if [ "$bend_t" != "" ]
  then
    Bend_t=`calc "$bend_t*$SAMP/1000"`
    blank_t="-B $Bstr_t $Bend_t"
  fi
fi

S=""
echo -n "Do you want to use last 4 points of spectrum for stats \
y/[n] "
read foo
[ "$foo" = "y" ] && S="-s4"

#---------------------------------------------------------------
#Envoke summing program to integrate spectra over assinged intervals.

cat >>  FINAL  <<!

#ANALYSIS OF $ID

echo "$BEAT Beats Analyzed" > res.$ID
echo "QRS SEGMENT $s_qrs TO $e_qrs  Blank $bstr_qrs To $bend_qrs" \
  >> res.$ID
sumnew -b $S_qrs -e $E_qrs $blank_qrs -l $BEAT -f big.$ID $S \
  -ospc.qrs >>res.$ID
echo "ST SEGMENT $s_st TO $e_st  Blank $bstr_st To $bend_st" \
  >> res.$ID
sumnew -b $S_st -e $E_st $blank_st -l $BEAT -f big.$ID $S  -ospc.st \
  >> res.$ID
echo "T SEGMENT $s_t TO $e_t  Blank $bstr_t To $bend_t" >> res.$ID
sumnew -b $S_t -e $E_t $blank_t -l $BEAT -f big.$ID    $S  -ospc.t \
  >> res.$ID
/home/pa/bin/math spc.qrs 0 spc.st 0 spc.t 0 \
  -o"(row/($BEAT * 2)),c0,c1,c2" -D -P > spc.$ID
!
#--------------------------------------------------------------

#NOTE:this version will create spc file with 4 columns- adding the
#col 0 ( the x axis scaled to cycles/beat)
#convert horizontal axes of spc file  to cycles/beat
#commented out as will use BEAT as value for rows in spc file
#allrows='rows spc.$ID '

#--------------------------------------------------------------
#PLOTTING ROUTINE
#plotting programs to replace plotall

[ ! -f "mat.$ID" ]  && continue
#will plot one blanking interval in ecg, this bit chooses which
#segment (qrs,st,t) was selected for blanking period within it.
BLANK=""
[ -n "$blank_qrs" ] && BLANK="$blank_qrs"
[ -n "$blank_st" ]  && BLANK="$blank_st"
[ -n "$blank_t" ]  && BLANK="$blank_t"
[ -z "$BLANK" ]    && BLANK="junk 0 0"
echo "$BLANK" > tmp.BLANK
read junk  s_blank e_blank < tmp.BLANK
rm tmp.BLANK

#make plot of qrs st t spectra
cat >> FINAL <<!
```

```
plt spc.$ID 0 1 -T lw -W .1 .73 .45 .98 -stx -F"
hl .5 .95 c 1
QRS Spectrum" > p.$ID
plt spc.$ID 0 2 -T lw -W .1 .435 .45 .685 -stex  -F"
hl .5 .95 c 1
ST Spectrum" >> p.$ID
plt spc.$ID 0 3 -T lw -W .1 .14  .45 .39  -set -F"
hl .5 .95 c 1
T Spectrum
hl .5 -.28 c 1
cycles/beat
hl 1.2 -.5 c 1
$ID ( $BEAT beats )" >> p.$ID
!

#make plot of matrix file
#plot vector ecg with appropriate interval and blanking bars
cat >> FINAL <<!
plt mat.$ID 0 -T lw -W .61 .73 .96 .98 -stex -c $S_qrs 0 $S_qrs 100 \
                                        -c $E_qrs 0 $E_qrs 100 \
                                       -c $S_st  0 $S_st  100 \
                                       -c $E_st  0 $E_st  100 \
                                       -c $S_t   0 $S_t   100 \
                                       -c $E_t   0 $E_t   100 \
                                       -c $s_blank 0 $s_blank - \
                                       -c $e_blank 0 $e_blank -  -F"
hl .5 .95 c 1
ECG Magnitude" >> p.$ID
!
#plot alternans metric as function of time
#This version will plot alternans metric as PPM by multiplying
#value by 1 X 10 ** 6 and plots on linear scale
#Also forces all negative values to zero
#plot k score as function of time
#calculate cutoff k score
#Uses cutoff of 3 so on log scale cutoff is .47712
logcut=".4771212"
#will force all data with value < 1 to = 1
#then plots on log scale

cat >> FINAL <<!
/home/pa/bin/math mat.$ID 1 -b"c0 > 0"         \
                  -e"c0=0"           \
                  -o"c0*1000000" -D | plt 0  -T lw \
                  -W .61 .435 .96 .685 -stex -F"
hl .5 .95 c 1
Alternans metric (ppm)" >> p.$ID


/home/pa/bin/math mat.$ID 2 -b"(c0>1)" -e"c0=1" \
                -o"row*1000/$SAMP,lgt(c0)" -D |plt \
                0 1 -T lw -W .61 .14 .96 .39 -ste  -ly \
                -c 0 $logcut   -  $logcut   -F"
hl .5 .95 c 1
K score
hl .5 -.28 c 1
milliseconds"  >> p.$ID
!
```

```
done


chmod 555 FINAL
( wpath='pwd'
#nice -20  FINAL
FINAL
bell
echo "final finished in $wpath"
chmod 777 FINAL
rm spc.qrs spc.st spc.t FINAL
) &
```

```
#********************************************************************
#                                                                   *
#                           results                                 *
#                                                                   *
#********************************************************************
#
#DESCRIPTION:
#
#THIS VERSION DOESS NOT OUTPUT SIGNAL TO NOISE DATA
#
#This script will generate data output of data with 3 digit idcode
#designation. So if the input idcode is 123, all files in the
#directory appended with the suffix 123 will be used to produce
#a hard copy of the results tables (res.nsr.123 files), a laser
#plot of aggregate qrs,st,tw spectra (col 0,1,2 of spc.nsr.123
#files), and a laser plot of the alternans metric with k score
#(col 1,2 of mat.nsr.123 files).
#
#PROGRAMS:
#                   plt
#                   lprint
#
#INPUT FILES:       res.intervention.idcode

#                   seg.intervention.idcode
#                   spc.intervention.idcode
#                   mat.intervention.idcode
#                   int.intervention.idcode
#
#OUTPUT FILES:      laser plots
#
#
#
idcode=$1
dir=`pwd`

#until [ -f "res.*.$idcode" ]
#do
#echo "results can't find res.*.$idcode in $dir"
#echo "Have you entered the correct idcode? e.g. res.nsr.123 \
#idcode=123"
#echo -n "Enter idcode: "
#read idcode
#done
if [ -z "$1" ]
then
echo "results filename (e.g. 398)"
exit 1
fi

echo "Checking for input files...."

/bin/rm -f PLOTSERROR

for file in  `ls res.*.$idcode`
do
  lprint  $file
  ID=`echo "$file" | cut -c5- `

  if [ ! -f "seg.$ID" ]
  then
    echo "results can't find seg.$ID. Will not plot ECG data"
  else
    lwcat seg.$ID 2>PLOTSERROR
  fi

  if [  -f "p.$ID" ]
  then
    lwcat p.$ID 2>PLOTSERROR
  else
    echo "results can't find  p.$ID. Will not plot spectra for $ID"
  fi

done

echo "results finished"
bell
cp $file /db/yin/twa/result
/bin/rm -f PLOTSERROR
```