# **ECE-497/BME-491:** Applied Biomedical Signal Processing
# Laptop Day #8
# Due at the end of class, January 23, 2007

Today we have the following goals

- trigger on ECG signals
- select a template from an ECG signal
- signal average the template
- perform template matching to discriminate SR and VT
- create and ROC curve to examine results

*At the end of class you should turn in five plots.*

1) Go to the class website and download *laptop8_files.rar*, then uncompress them.

2) There are a number of both data files and m-files in *laptop8_files.rar*. Most of these are files you have either used before or are modifications of files you created in previous laptop assignments. The one we are most interested in at this point is the script file *do_all_pt2.m* which runs most of the other programs. Look this over and try and understand what it is doing. It is important that the correlation coefficients be saved at the end of the program, and each patient should have his/her data written to a different file.

- The first thing the program does is use the Murthy and Rangaraj method for detecting the beats. This algorithm was origianlly designed for detecting the QRS on the body surface with a much lower sampling rate, but it has been slightly modified for our use. The only argument you need to change for now is the last argument in the call to *M_and_R_QRS.m.* This parameter sets the triggering threshold at this fraction of the maximum of the $g$ parameter. Every time you run this code look at the plots to be sure the threshold is set acceptably. The upper figure shows the $g$ parameter and the location of the threshold, while the lower figure shows the intracardiac signal and the location of the beats. *Modify the threshold as necessary to be sure you detect all of the beats (except maybe the first and the last one.)* If the threshold is not good enough, hit (control) c in the Matlab window to stop the script, delete the figures present, change the threshold, and start again.
- The next thing that will happen is you are shown a picture of a beat and where the triggering point is located. *The initial template is not acceptable!* Now we need to make a template, but this is a bit tricky, since you and I may not have the same triggering points. However, we will assume we are not that far off. Use the *left* and *right* points in Table 1 to determine the template.

- The next thing the script will do is run a program to do some signal averaging. In this case the template is aligned at both the original trigger point and at each point within ±10 points of the trigger point. The number of good fits (with a correlation coefficient larger than 0.95) is determined. *Use all the beats you can for creating the signal averaged template.*

- Finally, the script completes running, matching the template to each beat detected and computing the correlation coefficient for each beat. At this point, the template is aligned only at the triggering point, not at the best fit near the triggering (this is why the parameter *delta* is zero). These are stored in arrays which are saved. The last thing the program does is to show the values of the correlation coefficients between the template and the SR beats with a '+'. The correlation coefficients between the template and the VT beats are shown with a variety of different symbols. We are hoping for no overlap, but that may not be the case. *Print out this plot for each patient and turn it in*

3) Now save *do_all_pt2.m* with appropriate names, modify the file as necessary, and process the data from patients 4, 5, and 9. Note that not all of these patients have the same number of VT data files, and the triggering thresholds may need to be changed for each patient. The template size will need to be changed for each patient, and finally be sure to save the data in appropriate names (different for each patient). *Print out the final plot for each patient to turn in.*

4) The program *make_ROC.m* has been partly written for you, but you need to finish it.

- Set the parameter $\gamma$ to be 1000 points between 0 and 1. Using a *for loop* (or an equivalent) fill in the arrays for the probability of detection $P_d$ and probability of false alarm $P_f$. (*Hint: if t = [0 1 2 3] and you type sum(t>1.5) you will get 2*)

- Using a *for loop* (or an equivalent), estimate the area under the ROC curve. A simple approximation for the area under one small piece would be

$$\Delta area \quad = \quad [P_f(i) - P_f(i-1)]\,\frac{1}{2}\,[P_d(i) + P_d(i-1)]$$

My ROC curve for this data look like that in Figure 1, though yours may be somewhat different since your triggering points may have been different. *Plot your ROC curve and turn it in.*

Turn in your code for your *make_ROC.m.*

| Patient | Left | Right |
|---|---|---|
| 1 | 60 | 20 |
| 2 | 60 | 10 |
| 3 | 70 | 10 |
| 4 | 90 | 15 |
| 5 | 40 | 60 |
| 6 | 70 | 10 |
| 7 | 60 | 0 |
| 8 | 100 | 0 |
| 9 | 50 | 0 |
| 10 | 60 | 10 |

Table 1: Left and Right (of trigger point) for template selection

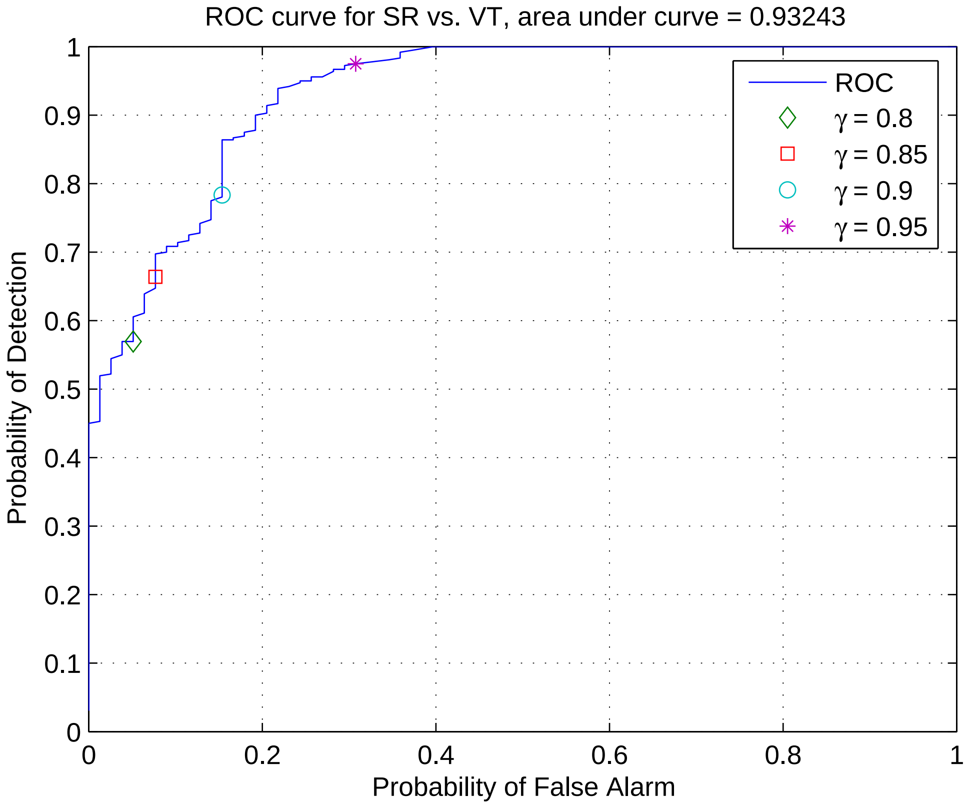

Figure 1: Example ROC curve for patients 2, 4, 5, and 9.