A2HBC Tutorial

Mariano Llamedo Soria

Version 0.1 beta

1 What is it ?

The *Argentino-Aragonés heartbeat classifier* (A2HBC) is a Matlab script developed for research purposes during my PhD studies [1]. As you can guess, it is just a heartbeat classifier. The main objective of this software is to ease the performance comparison against other (hope better) heartbeat classifiers. You can also use it for classifying unlabeled ECG recordings.

2 Features

The main features are:

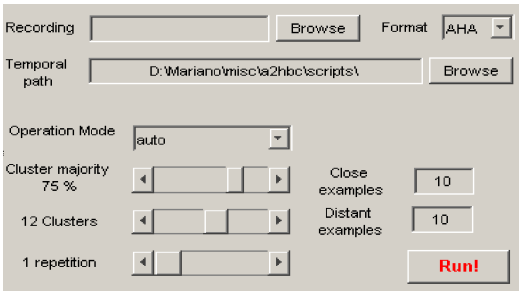
- *Validated performance*. The performance was thoroughly evaluated in [2].
- *Several formats accepted* (MIT, ISHNE, AHA and HES)
- *Open source*. Documented and easily customizable.
- *Multiprocessing ready*. It is ready to run in both a desktop PC or a high performance cluster.
- *user interface*. The algorithm have a simple graphical user interface (GUI) to ease the labeling of heartbeat clusters.

3 Usage

Several examples are included in the *examples.m* script. In this tutorial we will show some examples that any user can run to understand how to use A2HBC, with some ECG recordings included. It can be executed simply by its name:

```
>>a2hbc
```

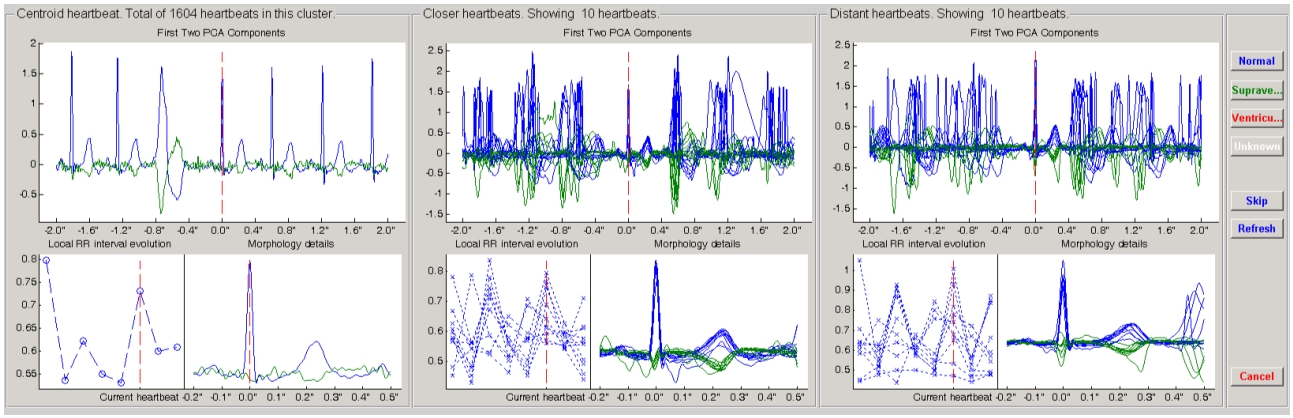
In this case the *control panel* is displayed to gather some information from the user:



You must enter the recording file name and its format, the other controls you can leave with its default values by the moment. You can select the *208.dat* recording, included in the *example recordings* folder, and set the *MIT* format. Then press *Run!,* the script will start working. You can follow the evolution in the progress bar, and after a while, it ends and display the classification results

```
Configuration
-------------
+ Recording: ... \example recordings\208.dat (MIT)
+ Mode: auto (12 clusters, 1 iterations, 75% cluster-presence)

  True             | Estimated Labels
  Labels           | Normal  Suprav  Ventri  Unknow| Totals
 ------------------|-------------------------------|-------
  Normal           | 1567       6      13       0  | 1586
  Supraventricular|    2       0       0       0  |    2
  Ventricular      |  255       8    1102       0  | 1365
  Unknown          |    2       0       0       0  |    2
 ------------------|-------------------------------|-------
  Totals           | 1826      14    1115       0  | 2955

Balanced Results for
--------------------
| Normal     || Supravent || Ventricul ||            TOTALS             |
|  Se    +P  ||  Se    +P  ||  Se    +P  ||    Acc    |    Se     |    +P     |
|  99%   45% ||   0%    0% ||  81%   99% ||    60%    |    60%    |    48%    |

Unbalanced Results for
----------------------
| Normal     || Supravent || Ventricul ||            TOTALS             |
|  Se    +P  ||  Se    +P  ||  Se    +P  ||    Acc    |    Se     |    +P     |
|  99%   86% ||   0%    0% ||  81%   99% ||    90%    |    60%    |    62%    |
```

This is possible because this recording include the expert annotations, or *ground truth*, for each heartbeat. The manual annotations in MIT format are typically included in *.atr* files (in this case *208.atr*). Now you can check other operation modes, as the *slightly-assisted*. Click on *Run!* and then, eventually, the algorithm may ask you for help. In case of needing help, a window like this will appear:



In this window the algorithm is asking you to label the centroid of the cluster, that is showed in the left panel. In the top of each panel some information is showed, as the amount of heartbeats in the current cluster. In the middle panel, you have some examples of heartbeats close to the centroid in a likelihood sense. The same is repeated in the right panel, but with examples far from the centroid. This manner you can have an idea of the dispersion of heartbeats within a cluster. Large differences across the panels indicates large cluster dispersion. If you decide to label the cluster, you can use one of the 4 buttons on your right. The unknown class is reserved for the cases where you can not make a confident decision. At the same time, in the command window, a suggestion appears:

```
Configuration
-------------
+ Recording: .\example recordings\208.dat (MIT)
+ Mode: assisted (3 clusters, 1 iterations, 75% cluster-presence)
Suggestion: Normal
```

This means that the centroid heartbeat in the *.atr* file is labeled as *Normal*. You will see this suggestion for each cluster analyzed, if there are annotations previously available. You are informed about the percentage of heartbeats already labeled with a progress bar, in the bottom of the control panel window.

In case you believe that a cluster includes several classes of heartbeats, you can decide to *skip* the classification, and try to re-cluster those heartbeats in the next iteration. You are free to perform as many iterations as you decide, by skipping clusters. The refresh button resamples heartbeats close and far from the centroid, and then redraw the middle and right panels. This feature is useful for large clusters.

There are two possible ways of using A2HBC, in a single desktop PC or in a high performance cluster of computers.

3.1 The power of the command-line

You can control all the features described up to the moment (and more) from the command-line of Matlab. This is particularly useful for integrating *a2hbc* in your scripts. You can find several examples in the script *examples.m*, this is probably the best way of getting familiar with it. Here I reproduce some worked examples included in this script. First let's start executing

```
a2hbc( ...
    'recording_name', [ '.' filesep 'example recordings' filesep '208.dat'], ...
    'recording_format', 'MIT', ...
    'op_mode', 'auto');
```

As you can see, the parameter interface of *a2hbc* is by *name-value* parameters. In the previous example, the name of the parameters are self-explanatory, the only comment is for the third, which is the operating mode. In Table 1 you can check the complete list of parameters. As a result, you will get similar results to the obtained in the first example using the GUI.

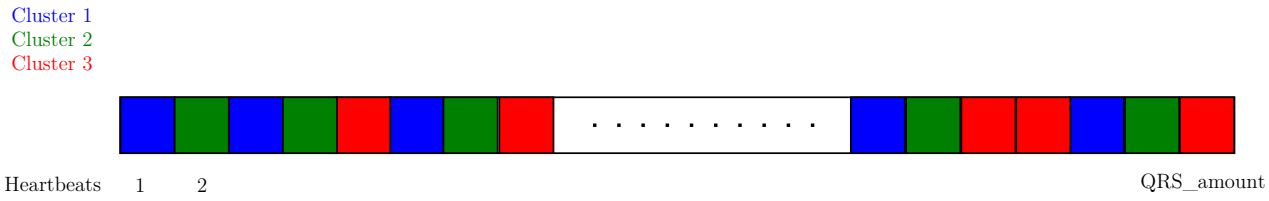
```
Configuration
-------------
+ Recording: .\example recordings\208.dat (MIT)
+ Mode: auto (12 clusters, 1 iterations, 75% cluster-presence)

  True             | Estimated Labels
  Labels           | Normal  Suprav  Ventri  Unknow| Totals
 ------------------|-------------------------------|-------
  Normal           | 1575       3       8       0  | 1586
  Supraventricular|    2       0       0       0  |    2
  Ventricular      |  250       7    1108       0  | 1365
  Unknown          |    1       0       1       0  |    2
 ------------------|-------------------------------|-------
  Totals           | 1828      10    1117       0  | 2955

Balanced Results for
--------------------
| Normal     || Supravent || Ventricul ||            TOTALS             |
|   Se    +P  ||   Se    +P  ||   Se    +P  ||    Acc    |    Se     |    +P     |
|   99%   46% ||    0%    0% ||   81%   99% ||    60%    |    60%    |    48%    |

Unbalanced Results for
----------------------
| Normal     || Supravent || Ventricul ||            TOTALS             |
|   Se    +P  ||   Se    +P  ||   Se    +P  ||    Acc    |    Se     |    +P     |
|   99%   86% ||    0%    0% ||   81%   99% ||    91%    |    60%    |    62%    |
```

In the case that you would like to integrate a2hbc to your software, or you have a proprietary ECG format not allowed by a2hbc, the best choice is that you pass the ECG samples directly. For doing this, you will have follow the requirements in Table 1 indicated with a [2], and

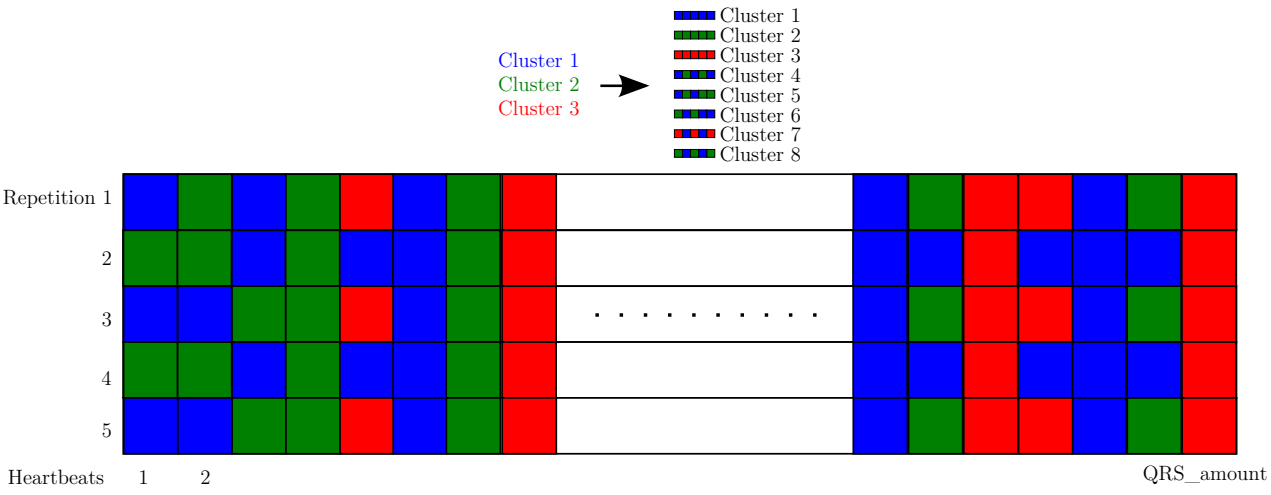
- *ECG* is the ECG signal matrix of *nsamp*× *nsig*, in ADC samples.
- *ECG_header* is a *struct* with the ECG properties, with fields:
  - *freq* is the sampling rate of the ECG
  - *nsamp* is the number of samples
  - *nsig* is the amount of leads.
  - *gain* is a vector of *nsig*×1 with the gain of each lead ($^{\text{ADC samples}}/_{\mu V}$).
  - *adczero* is a vector of *nsig*×1 with the offset of each lead in ADC samples.
- *QRS_annotations* is a *struct* with the location of the QRS complexes, with fields:
  - *time* is a vector of *QRS_amount*×1, with the sample value where the QRS complexes are.
  - *ann_type* [optional] is a *char* vector of QRS_amount×1, with each heartbeat label. This field is for evaluating the performance of a classifier, as a result *a2hbc* generates the confusion matrix seen in the examples above.

The parameters 'cant_pids' and 'this_pid' are explained in the next section, since were designed for partitioning and multiprocessing of recordings. The parameter *SimulateExpert* was designed to simulate the expert input by using the expert annotations provided in the annotations files, or via the *QRS_annotations* parameter. *ClusterPresence* is a threshold for evaluating the qualified majority in operating modes *auto* or *slightly-assisted*. The lower this threshold the more confident the algorithm in labeling all heartbeats in a cluster as the centroid. *Repetitions* was designed to evaluate multiple times the algorithm performance in a particular recording. As a result, the computed confusion matrix is a 3-D cube with the amount of repetitions as the third coordinate. With this kind of confusion matrix it is possible to estimate the dispersion of the results presented in [1, 2]. Finally, the *ClusteringRepetitions* parameter requires a special explanation, since it was not described in the bibliography. In few words, it is a trick for increasing the clustering resolution for complex or long-term recordings. The higher this parameter, the higher the amount of cluster found and, at the end, the assistance required by the algorithm.

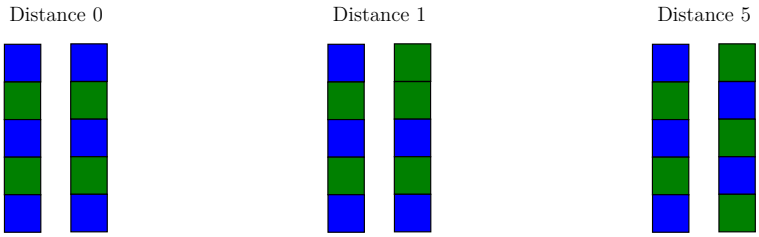
From the user point of view, you should be satisfied with this clue, however if you are interested in the trick, I will explain it with a toy example. Consider the following clustering problem, where we are interested in finding 3 clusters.



Now if we repeat the same process two times, our clustering algorithm does not guarantee to find the same partition. However our intuition tells us that in case where the classes are well separated, the partition is likely to remain very similar through the repetitions. In the opposite case, the partition can change. Then after N repetitions, the *QRS_amount* heartbeats were assigned N cluster labels.



So the number of clusters increase according to N. In order to keep this number in the order of tens, it was used a merging criterion for *similar clusters*. The similarity is measure in terms of labeling differences across the repetitions. For example:



Then we can group together some clusters based on this labeling distance. The *a2hbc* group together clusters within a distance of $0.2 \cdot N$.

3.2 The power of a high performance computing cluster

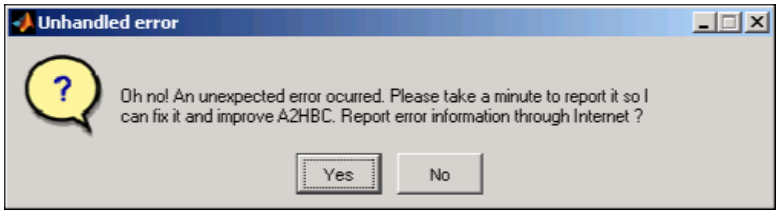
Maybe one of the most useful features of *a2hbc* is that was developed for being used in a high performance computing cluster. The parameters *cant_pids* and *this_pid* controls the partitioning of the work for each recording.

```
% Computer 1

lab1 = a2hbc( ...
    'recording_name', [ '.' filesep 'example recordings' filesep '208.dat'], ...
    'recording_format', 'MIT', ...
    'this_pid', 1, ...
    'cant_pid', 2, ...
    'op_mode', 'auto');

% Computer 2

lab2 = a2hbc( ...
    'recording_name', [ '.' filesep 'example recordings' filesep '208.dat'], ...
    'recording_format', 'MIT', ...
    'this_pid', 2, ...
    'cant_pid', 2, ...
    'op_mode', 'auto');


% Somewhere: results collection and processing
lab = [lab1; lab2];
...
```

It is recommended to adapt this features to the batch manager available in your computing facilities. In the case of our University, the batch manager used is Condor [3]. You can ask me for the condor implementation for multiprocessing, but the details of this are outside of the scope of this tutorial.

4 If something unexpected happens ...

Don't panic, *a2hbc* "should handle" nicely the exceptions. In case something unexpected happens you will see a message like this:



You can debug the problem by yourself or let *a2hbc* send a report to me by the Internet.

Table 1: List of parameters of *a2hbc*

Name	default	Value validation	Description
'recording_name'	–	*char*	The file name of the ECG recording[1]
'recording_format'	–	{MIT, 'ISHNE', 'AHA', 'HES', 'MAT'}	The format of the ECG recording[1]
'ECG'	–	*numeric*	The ECG samples[2]
'ECG_header'	–	*struct*	Struct with ECG features[2,3]
'QRS_annotations'	–	*numeric* && $x_i \geq 1$, $\forall i$	Sample occurrence of QRS complexes[2]
'op_mode'	'auto'	{'auto', 'slightly-assisted', 'assisted'} || $1 \leq x \leq 3$	
'cant_pids'	1	$x \geq 1$	How many processes in total to compute this recording[3]
'this_pid'	1	$x \geq 1$ && $x \leq$'cant_pids'	Which of the processes is this.[3]
'CacheData'	true	*logical*	Save intermediate results to speed-up re-processing ?
'InteractiveMode'	false	*logical*	Show the control panel after processing the recording.
'SimulateExpert'	false	*logical*	Use expert annotations to simulate expert interaction[3]
'tmp_path'	–	*char*	Path to store intermediate results.
'NumOfClusters'	12	$x > 1$	Number of cluster to search.
'ClusteringRepetitions'	1	$1 \leq x \leq 10$	Repetitions of the clustering process.[3]
'ClusterPresence'	75	$0 \leq x \leq 100$	Threshold for the qualified majority.[3]
'Repetitions'	1	$x \geq 1$	Repetitions to evaluate this recordings.[3]

[1] and [2] indicates groups of parameters that can not be mixed. You can specify file name and format, or pass the ECG samples, QRS annotations, etc.
[3] See a complete explanation in the text, or in the references [1, 2].

5 Acknowledgments

This software was supported by the University of Zaragoza, Spain and the National Technical University of Buenos Aires, Argentina.

A Cluster scripts

Master script: run_a2hbc.condor

```
#!/bin/bash

#            0   1      2      3         4              5          6         7         8      9
databases=( AHA ESTTDB LTSTDB MITBIH-AR MITBIH-AR-DS2 MITBIH-SUP MITBIH-ST INCART Biosigna  MITBIH-LT)
db_fmt=(    AHA MIT    MIT    MIT       MIT            MIT        MIT       MIT    HES       MIT)
pidsXrec=(  3   10      30     3         3              3          3         3      3         30
)
db_sizes=(  154  90     86     44        22             78         18        75     56        7
)
db_ext=(    ecg dat    dat    dat       dat            dat        dat       dat    hes       dat
)
db_paths=(  "/extra/database/bio/ecg/thew/aha/" \
        "/extra/database/bio/ecg/thew/European␣ST-T␣Database/" \
        "/extra/database/bio/ecg/thew/Long-Term␣ST␣Database/"  \
        "/extra/database/bio/ecg/thew/mitbih-ar/"  \
        "/extra/database/bio/ecg/thew/ds2/"  \
        "/extra/database/bio/ecg/thew/MIT-BIH␣Supraventricular␣Arrhythmia␣Database/"  \
        "/extra/database/bio/ecg/thew/MIT-BIH␣ST␣Change␣Database/" \
        "/extra/database/bio/ecg/thew/St␣Petersburg␣Institute␣of␣Cardiological␣Technics␣12-lead␣Arrhythmia
\
        "/extra/database/bio/ecg/thew/biosigna/"  \
        "/extra/database/bio/ecg/thew/The␣MIT-BIH␣Long␣Term␣Database/"  \
        )

#Typical configuration
 op_modes=(      auto slightly-assisted  assisted assisted  )
 NumOfClusters=(     9    9          9    12        )
 ClusteringRepetitions=( 1    1          1    1     )
 ClusterPresence=(   50   75         50   50        )

#Long-term configuration
#op_modes=(         assisted     )
#NumOfClusters=(        15         )
#ClusteringRepetitions=(    5          )
#ClusterPresence=(      50         )



#Esto sirve para que no haya dependencias de hacer algunos JOBS antes que otros.
#finish_preproc_first=1
finish_preproc_first=0

#Esto sirve para limitar la cantidad de procesos que corren en CONDOR.
#limited=1
limited=0

#numero de veces que reintenta un registro cuando ocurre un error. Esto esta
#pensado para cuando falla al acceder a un archivo en el filesystem en red.
retries=1

submit_delay=0
processes_running_total=1200
processes_idle_total=700 #esto es para no encolar mucho trabajo en Condor
databases_running=4
iterations=60


if [ "$limited" -eq "0" ]
then
```

```
    databases_running=${#databases_idx[@]}
fi

aux_str="*."
tmp_path=/extra/scratch/bio/mllamedo/tmp/
condor_output_path=/home/bio/mllamedo/ecg_classification/tmp/condor_output/

#do_cleanup=1 #SI limpia temporales anteriores
do_cleanup=0 #NO limpia temporales anteriores

#echo "Cleaning previous $condor_output_path_this_work/global.dag files ..."
#rm ./global.*
#echo "done."

global_parent_string="PARENT"

aux=$(echo "scale=0; ${#databases[@]}-1" | bc)
#databases_idx=($(seq 0 $aux) )
#databases_idx=($(seq 0 8) )
databases_idx=(3 4)



 [... lines omitted ...]



if [ "$limited" -eq "1" ]
then
    condor_submit_dag -f -maxjobs $databases_running $condor_output_path_this_work/global.dag
else
    condor_submit_dag -f $condor_output_path_this_work/global.dag
fi

echo "Logs en $condor_output_path_this_work"
```

Run the job

```
-bash-3.2$ . run_a2hbc.condor

MITBIH-AR limitara a  registros idle.
Generating dag files MITBIH-AR work ...

Renaming rescue DAGs newer than number 0
------------------------------------------------------------------------
File for submitting this DAG to Condor           : /home/bio/mllamedo/ecg_classification/tmp/condor_output
Log of DAGMan debugging messages                 : /home/bio/mllamedo/ecg_classification/tmp/condor_output
Log of Condor library output                     : /home/bio/mllamedo/ecg_classification/tmp/condor_output
Log of Condor library error messages             : /home/bio/mllamedo/ecg_classification/tmp/condor_output
Log of the life of condor_dagman itself          : /home/bio/mllamedo/ecg_classification/tmp/condor_output

-no_submit given, not submitting DAG to Condor.  You can do this with:
"condor_submit /home/bio/mllamedo/ecg_classification/tmp/condor_output/2012_03_01-13_02_01/MITBIH-AR.dag.c
------------------------------------------------------------------------
Runtime error (func=(main), adr=12): Divide by zero
Runtime error (func=(main), adr=11): Divide by zero
MITBIH-AR-DS2 limitara a  registros idle.
Generating dag files MITBIH-AR-DS2 work ...

Renaming rescue DAGs newer than number 0
------------------------------------------------------------------------
File for submitting this DAG to Condor           : /home/bio/mllamedo/ecg_classification/tmp/condor_output
Log of DAGMan debugging messages                 : /home/bio/mllamedo/ecg_classification/tmp/condor_output
Log of Condor library output                     : /home/bio/mllamedo/ecg_classification/tmp/condor_output
Log of Condor library error messages             : /home/bio/mllamedo/ecg_classification/tmp/condor_output
Log of the life of condor_dagman itself          : /home/bio/mllamedo/ecg_classification/tmp/condor_output

-no_submit given, not submitting DAG to Condor.  You can do this with:
"condor_submit /home/bio/mllamedo/ecg_classification/tmp/condor_output/2012_03_01-13_02_01/MITBIH-AR-DS2.d
------------------------------------------------------------------------

Renaming rescue DAGs newer than number 0
------------------------------------------------------------------------
File for submitting this DAG to Condor           : /home/bio/mllamedo/ecg_classification/tmp/condor_output
```

```
Log of DAGMan debugging messages                 : /home/bio/mllamedo/ecg_classification/tmp/condor_output
Log of Condor library output                     : /home/bio/mllamedo/ecg_classification/tmp/condor_output
Log of Condor library error messages             : /home/bio/mllamedo/ecg_classification/tmp/condor_output
Log of the life of condor_dagman itself          : /home/bio/mllamedo/ecg_classification/tmp/condor_output

Submitting job(s).
1 job(s) submitted to cluster 3255737.
-----------------------------------------------------------------------
Logs en /home/bio/mllamedo/ecg_classification/tmp/condor_output/2012_03_01-13_02_01
```

Query the job status

```
-bash-3.2$ condor_q -dag mllamedo


-- Submitter: hermes@ : <172.16.4.100:37992> : hermes.cps.unizar.es
 ID      OWNER/NODENAME   SUBMITTED     RUN_TIME ST PRI SIZE CMD
3255737.0   mllamedo        3/1  13:02   0+00:00:42 R  0   7.3  condor_dagman -f -
3255738.0    |-MITBIH-AR    3/1  13:02   0+00:00:27 R  0   7.3  condor_dagman -f -
3255739.0    |-MITBIH-AR-D  3/1  13:02   0+00:00:22 R  0   7.3  condor_dagman -f -
3255740.0    |-100_auto0    3/1  13:02   0+00:00:00 I  0   0.0  condor_exec.sh a2h
3255740.1    |-100_auto0    3/1  13:02   0+00:00:00 I  0   0.0  condor_exec.sh a2h
3255740.2    |-100_auto0    3/1  13:02   0+00:00:00 I  0   0.0  condor_exec.sh a2h
3255741.0    |-101_auto0    3/1  13:02   0+00:00:00 I  0   0.0  condor_exec.sh a2h
3255741.1    |-101_auto0    3/1  13:02   0+00:00:00 I  0   0.0  condor_exec.sh a2h
3255741.2    |-101_auto0    3/1  13:02   0+00:00:00 I  0   0.0  condor_exec.sh a2h
3255742.0    |-103_auto0    3/1  13:02   0+00:00:00 I  0   0.0  condor_exec.sh a2h
3255742.1    |-103_auto0    3/1  13:02   0+00:00:00 I  0   0.0  condor_exec.sh a2h
3255742.2    |-103_auto0    3/1  13:02   0+00:00:00 I  0   0.0  condor_exec.sh a2h
3255743.0    |-105_auto0    3/1  13:02   0+00:00:00 I  0   0.0  condor_exec.sh a2h
3255743.1    |-105_auto0    3/1  13:02   0+00:00:00 I  0   0.0  condor_exec.sh a2h
3255743.2    |-105_auto0    3/1  13:02   0+00:00:00 I  0   0.0  condor_exec.sh a2h
```

Check for errors

```
-bash-3.2$ ./my_cat /home/bio/mllamedo/ecg_classification/tmp/condor_output/2012_02_29-18_10_55/



/home/bio/mllamedo/ecg_classification/tmp/condor_output/2012_02_29-18_10_55/MITBIH-AR_slightly-assisted1.e
??? Undefined function or method 'CollectResutls' for input arguments of type
'char'.









/home/bio/mllamedo/ecg_classification/tmp/condor_output/2012_02_29-18_10_55/MITBIH-AR_assisted2.err
??? Undefined function or method 'CollectResutls' for input arguments of type
'char'.




/home/bio/mllamedo/ecg_classification/tmp/condor_output/2012_02_29-18_10_55/global.dag.lib.err
Renaming rescue DAGs newer than number 0
Renaming rescue DAGs newer than number 0



/home/bio/mllamedo/ecg_classification/tmp/condor_output/2012_02_29-18_10_55/MITBIH-AR-DS2_assisted2.err
??? Undefined function or method 'CollectResutls' for input arguments of type
'char'.




/home/bio/mllamedo/ecg_classification/tmp/condor_output/2012_02_29-18_10_55/MITBIH-AR_assisted3.err
??? Undefined function or method 'CollectResutls' for input arguments of type
```

```
'char'.



/home/bio/mllamedo/ecg_classification/tmp/condor_output/2012_02_29-18_10_55/MITBIH-AR_auto0.err
??? Undefined function or method 'CollectResutls' for input arguments of type
'char'.



/home/bio/mllamedo/ecg_classification/tmp/condor_output/2012_02_29-18_10_55/MITBIH-AR-DS2_slightly-assiste
??? Undefined function or method 'CollectResutls' for input arguments of type
'char'.



/home/bio/mllamedo/ecg_classification/tmp/condor_output/2012_02_29-18_10_55/MITBIH-AR-DS2_assisted3.err
??? Undefined function or method 'CollectResutls' for input arguments of type
'char'.

Encontramos 804 archivos ...
de los cuales 10 con errores a reportar ...
```

Process the results

```
% Collect results for each operation mode
>> CollectResutls( '/extra/scratch/bio/mllamedo/tmp/MITBIH-AR', 'auto' )
>> CollectResutls( '/extra/scratch/bio/mllamedo/tmp/MITBIH-AR', 'slightly-assisted' )
>> CollectResutls( '/extra/scratch/bio/mllamedo/tmp/MITBIH-AR', 'assisted' )

>> ResultsForAllDatabases( '/extra/scratch/bio/mllamedo/tmp/', ...
                           {'auto' 'slightly-assisted' 'assisted'} )
```

```
######################
## AHA auto_9_1_50 ##
######################
Encontramos 60.00 iteraciones.
Warning: Assumnig equal priors.
> In DisplayResults at 129
  In ResultsForAllDatabases at 83

  True             | Estimated Labels
  Labels           | Normal Suprav Ventri Unknow| Totals
 ------------------|----------------------------|-------
  Normal           | 296739(736)  10589(363)  10406(647)  0(0) | 317735(0)
 Supraventricular| 0(0)  0(0)  0(0)  0(0) | 0(0)
 Ventricular      | 1660(131)  6653(231)  25674(250)  0(0) | 33987(0)
 Unknown          | 201(23)  65(11)  306(24)  0(0) | 572(0)
------------------|----------------------------|-------
  Totals           | 298600(752)  17308(469)  36387(818)  0(0) | 352294(0)

Balanced Results for AHA
------------------------
| Normal               || Supraventricular    || Ventricular         || Unknown             ||
TOTALS                  |
|     Se       +P      ||     Se       +P      ||     Se       +P      ||     Se       +P      ||
Acc       |     Se        |       +P       |
|  93(  0)%  70(  2)% ||  --( --)%  --( --)% ||  75(  1)%  96(  0)% ||   0(  0)%   0(  0)% ||
56(  0)%   |   56(  0)%   |   32(  0)%   |

Unbalanced Results for AHA
--------------------------
| Normal               || Supraventricular    || Ventricular         || Unknown             ||
TOTALS                  |
|     Se       +P      ||     Se       +P      ||     Se       +P      ||     Se       +P      ||
Acc       |     Se        |       +P       |
|  93(  0)%  99(  0)% ||  --( --)%  --( --)% ||  75(  1)%  71(  1)% ||   0(  0)%   0(  0)% ||
92(  0)%   |   56(  0)%   |   42(  0)%   |
```

```
[... lot of results ...]


Resumen de performances globales
--------------------------------

AHA      auto    9_1_50  93XX0   99XX0   NaNXXNaN        NaNXXNaN        75XX1   71XX1   0XX0    0XX0    92
AHA      auto    9_1_100 91XX0   99XX0   NaNXXNaN        NaNXXNaN        75XX0   66XX0   0XX0    0XX0    89
AHA      slightly-assisted       9_1_75  96XX0   99XX0   NaNXXNaN        NaNXXNaN        86XX1   82XX0   49
AHA      assisted        12_1_50 100XX0  100XX0  NaNXXNaN        NaNXXNaN        97XX0   98XX0   63XX5   83
AHA      assisted        9_1_50  100XX0  100XX0  NaNXXNaN        NaNXXNaN        97XX0   98XX0   54XX3   81

[... lot of results ...]

MITBIH-AR        auto    9_1_50  96XX0   98XX0   74XX4   42XX2   84XX1   88XX1   25XX4   30XX4   7XX7    1X
MITBIH-AR        auto    9_1_100 95XX0   98XX0   76XX0   35XX0   76XX0   77XX0   0XX0    0XX0    93XX0   62
MITBIH-AR        slightly-assisted       9_1_75  98XX0   99XX0   78XX3   62XX3   86XX2   93XX1   47XX14  60
MITBIH-AR        assisted        12_1_50 100XX0  99XX0   89XX2   88XX3   91XX1   97XX1   54XX13  70XX5   0X
MITBIH-AR        assisted        9_1_50  99XX0   99XX0   86XX1   85XX2   89XX2   97XX1   49XX6   63XX4   0X
```

Bibliography

References

[1] M. Llamedo, "Signal processing for automatic heartbeat classification and patient adaptation in the electrocardiogram," Ph.D. dissertation, Universidad de Zaragoza, June 2012, Online.

[2] M. Llamedo and J. Martínez, "An automatic patient-adapted ecg heartbeat classifier allowing expert assistance," *Biomedical Engineering, IEEE Transactions on*, 2012.

[3] Condor, "Condor high throughput computing system," 2010. [Online]. Available: http://www.cs.wisc.edu/condor/