THE CTH CLUSTER PACKAGE V 1.3.4

The CTH clustering package contains software tools that provide a variety of cluster analysis algorithms using data from neuron recordings and methods of viewing the results. This document provides information on how to use the tools.

WHAT'S NEW FOR 1.3.4

Version 1.3.3 was a bug fix release that has no new functionality.

Version 1.3.4 adds the ability to process swallow experiments. These are different from the other input files because they do not use I and E pulses to determine how the bins are calculated. Instead, there are start-of-swallow and an end-of-swallow markers. By default, two files are output for these types of input files. The first contains just the swallow CTHs. The second contains the control period CTHs and the swallow period CTHs as pairs. There is not an immediate relationship between these because the control CTHs are determined by I and E markers and the swallow CTHs by their markers.

The clustering software now has the ability to use two Archetype files when clustering a control/swallow .cth file. A preliminary set of swallow Archetypes is included in the release.

WHAT'S NEW FOR 1.3.2

The cthgui program has a new feature that lets you find the CTHs that were in one cluster during the control period and in a different cluster in the stim period. The results are saved to a .cth file than can be loaded in the cthgui program. It also creates a .csv file that the brainstem program can use. Use the "Export Ctl/Stim Pairs" button to find the pairs and save to a file. The default file name is based on the current file name. This is only supported for clustering using Archetypes or dendrogram. There is no support for doing this for other clustering methods, such as K-Means. This is only done for .cth files that have both control and stim periods.

WHAT'S NEW FOR 1.3.1

New features and bug fixes include:

cth_cluster:

Added support for cough experiments.

The -f somefile.xls argument is now required. There is no default. This caused confusion and sometimes the default .xls file was not available.

The default number of bins is now 100. You only need to use the -b flag if you want a different number of bins.

There are several perl scripts that cth_cluster needs. It invokes one of them, cth_cells.pl. This can be anywhere on the user's executable path. This script uses two other scripts, cth_data.pl and cth_xls2csv.pl. These are assumed to be in the current directory. I modified cth_cells.pl to extract the path to itself and added this to the invocation of the other scripts. It is painful to have to keep copying the scripts to the current directory. These are in /usr/local/bin by default. The prefix cth_ has been added to these scripts to avoid collision with the scripts they were derived from. The cth_xls2csv.pl script was modified to not print blank rows after the data. Some of the spreadsheets have a lot of empty rows and users were confused to see a screen or two of just rows of commas.

Fixed a bug that was introduced when the initial support for control / stim was added. It was not obvious for the CO2 experiments, but the cough experiments made it obvious something was broken. The error was that each period was normalized relative to itself. This had the effect of placing the control and stim CTHs into different coordinate systems. The fix was to normal all of the CTHs from both periods using the same scale factor.

cthgui:

Added support for cough.

The concate_types.m function was added to merge together sets of archetype CTHs from clustering runs on subsets of the archetype source CTHs. There is a convenience script that calls this, merge_arch.m.

brainstem:

Added better support for archetypes. Instead of showing cluster numbers, if the input file was clustered using archetype, the archetype number is displayed.

cth-demo package:

Generating the demo files takes a long time, up to an hour or two, which really gets in the way of a fast turn-around to fix a killer bug. These are now in their own optional package. These are not installed by default.

WHAT'S NEW FOR 1.3.0

Version 1.2.9 was a bug fix release that corrected a defect in 1.2.8.

New features include:

The cth_cluster program now has the ability to create .cth files that include only vagotomized, non-vagotomized, or all experiments.

The cthgui program now includes archetype clustering. This uses a set of archetype CTHs that were extracted from several clustering operations. These were identified as a set of CTHs that were representative of unique firing patterns. They are saved in a .type file. The archetype clustering operation reads in CTHs from a .cth file and for each CTH finds the archetype it is closet to.

WHAT'S NEW FOR 1.2.8

Version 1.2.7 was a bug fix release that corrected a defect in 1.2.6.

Bug fixes. Added an option to draw large projection plot and dendrogram wins suitable for screen captures.

Instead of a fixed number of up to 12 CTH plot windows, this version calculates how many rows and columns it needs to tile the CTH monitor with the CTHs, up to a limit if 8 x 8. If you create more than 64 clusters, the windows are too small to be usable. In this case, use the scroll buttons to view a subset of the clusters.

Support for creating and using CTH cluster centroids as archetypes for clustering. The CTH Info button now shows the nearest and next-nearest clusters for the CTH.

Support for showing control and stim periods in CTH plots.

WHAT'S NEW FOR 1.2.6

Bug fixes for different numbers of monitors and resolutions.

Circular B-spline curves generated and included in the demo files.

The cth_env utility now creates symbolic links instead of making local copies of the demo files. The user cannot change these, so local copies are not required.

WHAT'S NEW FOR 1.2.5

A couple of versions were for internal release/testing and are not generally available.

The .cth file version has been changed from version 1.15 to 2.0. You will need to recreate your local .cth files using the cth_cluster program. The version 1.2.5 package needs additional information that does not exist earlier versions. The clustering program will check the version and will refuse to load earlier versions.

The cth_cluster program now supports multiple types of periods. These include the generally longer initial control period, and control/stimulation pairs. This version supports carotid CO2 and vertebral CO2 control/stim events. By default, up to three .cth files are produced, with ctl, cco2, and vco2 as part of the file names. It is expected that other types, such as cough and swallow will be added in a later version. Details about how this works is provided later in the document.

An Export Cluster Archetypes button and function has been added. The mean/centroid CTH for each of the current set of clusters, including dendrogram and optionally other clustering methods, are saved to files with a .type extension. The format of these is the same as a .cth file.

A pick-files control has been added to select .type files.

A Cluster Using Archetypes button and function has been added. This uses the selected .type file as the source of the cluster centers, and creates clusters using the CTHs in the current .cth file. Details about how this works is provided later in the document.

A Fuzzy C-Means type of clustering has been added. This produces a list of the probabilities of a CTH being in the nearest five clusters.

The brainstem program is a new tool that is both an atlas replacement and a means of exploring CTH firing behavior in a visual model of the brainstem.

The color palette has been reworked to enhance the difference between colors. In earlier versions, some of the colors were difficult to distinguish. Unfortunately, colors are a function of the monitor being used. What looks unique on one monitor may look very similar on another one. There is not a lot the software can do about this.

The current list of experiments has been adjusted and new experiments added.

A Help menu has been added that contains some links to documents and other information.

WHAT'S NEW FOR 1.2.2

The Export To Atlas button now reads "Export To Atlas And Database". Clicking on this creates a second CSV file that contains information that can be used to create queries for records in existing databases. One use of this is to investigate how manual clustering and automated clustering correspond. The naming convention is that the file described in the 1.2.1 release will be yourfilename.csv, and the second one will be named yourfilename.db.csv. There is no need to add an extension when typing in the filename, the program will do that for you.

WHAT'S NEW FOR 1.2.1

There is now an Export To Atlas button in the GUI. This exports a file that an atlas package utility can can read and produce a file that can be imported into the atlas program. There is also a program, currently called brainstem, that is under development that will be able to read this file. The exported information is intended to be used to draw the CTH neurons at their stereotaxic coordinates and to color them based on what cluster they are in.

The cth summary window was not drawing the one tick bin plots correctly. Fixed.

The cth summary window did not work correctly on exported subsets of clusters. Fixed

If running the cth_project function in terminal mode, a non-existent function was being called. Fixed.

The example oneexp-100.cth file contained all of the experiments, not just one. Fixed. If you have already created a working environment, you may want to run `cth_env` again to pick up the new file, or look at the script in /usr/local/bin/cth_env and copy it by hand.

The file that contains all of the iterations of the curve fitting process has been moved to a new package, the cth-curves package. These do not change with every release (though sometimes they do), and it is a very large file to carry around in the cth package.

The GUI has been tweaked, as follows:

> The buttons on the left were rearranged so more commonly used buttons occur higher in the list.
>
> A -d debug flag was added that shows the Connect To Octave button. This button is almost always not needed and is now hidden by default.
>
> The high contrast color option and the 2D/3D projection options are rarely used and have been changed to checkboxes and moved to the Rarely Used Options box. The high contrast color choice has been renamed to Color Blind Friendly because that is what it is.
>
> The GUI elements move around now as the window is resized.
>
> The size and location of the window and its maximized state is now kept in a configuration file in the user's home directory in .config/cthgui/cthgui.conf.
>
> A color blind friendly palette was added.

QUICK START

The CTH cluster package is installed on all of the lab's Linux systems. You may have already set up an environment using version 1.2.0 or version 1.2.1. This version is not compatible with files generated by the previous versions, so you need to adjust your existing environment by recreating your own local .cth files. If you have not set up an environment, here is how to do that. In a command line window, change to your home directory or other working directory where you have write permission. Type this:

```
cth_env
```

This runs a script that creates a directory named cth and copies some files to this directory. The script then instructs you to type:

```
cd cth
```

and then:

```
run_cthgui
```

This is a script that starts an instance of the octave program and tells it to executes the cth_project function. This, in turn, starts the cthgui program. When the GUI opens, if you have installed the cth-demo package, you should see some demo files in the file list box. They may include these:

oneexp-10ctl.cth	-	one experiment, 10 bins, control period
oneexp-10cco2.cth	-	one experiment, 10 bins, carotid CO2 periods
oneexp-10vco2.cth	-	one experiment, 10 bins, vertebral CO2 periods
oneexp-20ctl.cth	-	and so on for 20 and 100 bins
oneexp-20cco2.cth		
oneexp-20vco2.cth		
oneexp-100ctl.cth		
oneexp-100cco2.cth		
oneexp-100vco2.cth		
allexp-10ctl.cth	-	all experiments, 10 bins, control period
allexp-10cco2.cth	-	and so on for carotid and vertebral 10, 20 and 100 bins
allexp-10vco2.cth		
allexp-20ctl.cth		
allexp-20cco2.cth		
allexp-20vco2.cth		
allexp-100ctl.cth		
allexp-100cco2.cth		
allexp-100vco2.cth		
100vago_vco2.cth		
100vago_cco2.cth		
100vago_ctl.cth		
100nonvago_vco2.cth		
100nonvago_cco2.cth		
100nonvago_ctl.cth		
allexp-100swall1.cth		
allexp-100ctl_swall1.cth		

It can take a long time to load all of the experiments included in the experiment list. The oneexp files load quickly and are useful when you want explore the features of the program.

You can now select different options and create plots using these files. Most of the controls have pop-up help.

The clustering software assumes you have two monitors that your window manager provides at least two workspaces. It will work with a single screen and workspace. I run it on my laptop without any problems, but the screen gets cluttered.

You should keep the terminal window you used to start run_cthgui open during the session. Text sometimes may be written to this window. If errors or other issues cause run_cthgui to terminate, please preserve the text. It will help figuring out what caused the error.

When the clustering software is running, the plot windows steal the focus from any other window that has the focus. If you switch to another window while waiting on the plots to be created, the next plot window that opens will take it away that window. If you switch to a different workspace, the plot windows will be displayed on it. Once the clustering run has finished, the GUI window will be brought back to the foreground. If you select the optional k-means plots, the clustering software will switch to the workspace to the right (if there is one) and display the plots there, then switch back to the left workspace.

So, once you start a clustering run, just sit back and enjoy the show.

DETAILS

The cth cluster package is composed of five components. The first two components take experiment data as input and create information about the data as output. The next two are data visualization and cluster analysis tools that work together to display and manipulate the information. The final component is a set of convenience scripts and data files.

1. cth_cluster

This is a command-line program that creates the .cth files. It reads an excel spreadsheet file containing lists of paths and file names to other excel spreadsheets, text files, .adt, .bdt, and .edt files. This is the "experiment list" referred to above. It creates one or more text files that contain information that other parts of the cth cluster package use. The example .cth files that were copied to your cth directory were created using this program.

2. cth_curve.R

This is an R program that takes an output file from cth_cluster and performs a circular b-spline curve fit to the CTH data. It can be executed as a stand-alone program or cth_cluster can execute it as part of its processing. The output is a text file that can be read by cth_cluster for inclusion in the .cth file. The curve fitting is very CPU intensive and will use multiple cores, so it is best run on one of the lab's high-end machines. The curves are independent of other variables, so, for example, the same fitted curve file can be use to create .cth files with different numbers of bins. There are examples of this later in the document.

3. cth_project.m and other octave functions

The bulk of the cluster analysis software is a collection of octave function files. The clustering and data visualization software is started by running octave and then calling the cth_project function in the cth_project.m file with some optional arguments. Most users will never run any of these directly. There is are scripts that manage the details for you.

4. cthgui

This is a program that provides a GUI user interface to the clustering software. You do not run this program directly, the cth_project function will start it for you.

5. brainstem

This is a replacement for the atlas program. It can read existing .dx files and can also read .cth files generated by the cthgui program. It needs its own User's Manual, and perhaps we will write one eventually. There is extensive tooltip help that is useful for first-time users.

6. Convenience scripts and data files

cth_env

This creates a directory and copies several files to provide an initial environment for the CTH clustering software.

run_cthgui

This is a shell script that you run in a command-line window to start a session using the GUI. It starts an instance of the octave program that calls an octave function that starts the cthgui program.

run_cthterm

This starts a session using the command-line interface instead of a GUI. This mode is intended to be used during development and debugging, but you can do almost everything from it that you can from the GUI.

allexp_from_r.txt

This is the output of cth_curve.R and contains the final fitted curves for all of the CTHs.

allcurves.tar.gz

This contains .pdf files that contain plots of all of the curve fitting iterations for all of the CTHs. This is a very large file and is not installed by default.

The purpose of these was to explore a conjecture that it might be possible to characterize CTHs using features of a fitted curve, such as number and location of peaks and valleys, width of these regions, and so on, to create objective types.

The curve fitting process is an iterative process. It fits piecewise cubic polynomial curves to the data. The endpoints are known as knots. The fitting process starts with 1 knot and continues up to 14 knots. At each stage of the iteration, some goodness-of-fit numbers are calculated. Version 1.2.6 stops iterating when the p-value is greater than or equal to 0.9. If it reaches the 14th iteration and the p-value is still too small, it stops and picks the iteration that has the largest p-value. The run time goes up significantly as the number of knots increases. I did a run with a single CTH using up to 50 knots and it ran for a couple of days. The largest p-value was never greater than 0.9. The problem with more knots is that the plots contain sharp V-like lines and do not look like smooth curves, though mathematically they are still continuous. The values of 14 and 0.9 are rather arbitrary.

The state of this at this time is that there is no obvious way to objectively detect curves that "look smooth" and curves with V lines. The curve fitting machinery could certainly use more work.

The allcurves.tar.gz file contains each iteration for all of the CTHs. It is a large file and is not copied to your working cth directory. If you want to look at the iterations, open a terminal window and cd to your working cth directory, then type this:

```
tar xf /usr/local/share/cth-curves/allcurves.tar.gz
```

This can take a while and requires 8G or more of space. This creates the fits/ directory under the one you are in and extracts the .pdf files that contain the plots. Then type:

```
cd fits
```

Each CTH has a unique number that identifies it. You can see these in the cth bar chart plots and other places. If you were interested in, say, CTH # 147, here is one way to view it, using the evince pdf viewer:

```
evince *_147_*.pdf
```

This will open as many windows as there are iterations for CTH #147. Evince stacks these, so it may look like there is only one, so you will have to move them around with the mouse. You can view the plots with any app than can open .pdf files.

MAKE YOUR OWN CTH FILES

You must be able to access files on cisc3 via the /raid and /dsk5 mount points. If you do not have /raid and /dsk5 directories in your root directory, ask someone with root access to create these and mount the cisc3 file systems.

The .cth files were created by the cth_cluster program. It uses an Excel spreadsheet that contains paths, file names, and other information. There is a copy of this installed as part of the cth cluster package and a copy is placed in your working cth directory. The original lives here:

/usr/local/share/cth-cluster/cth_data.xls

This is a symbolic link to the current version of the spreadsheet. The version may change in future releases, but it will be transparent since the symbolic link will be updated to refer to the new version. The cth_cluster program will use this by default if you do not specify an xls file. The current file will also be copied to your local work directory if you want to edit it and create subsets of the experiments.

The cth_cluster program has a lot of options. Here is what you get when you type:

```
cth_cluster -h

(1) Read .edt, .bdt, and .adt files.
(2) Accumulate firing rates of neuron channels in control and
    optionally other periods into a variable number of bins.
(3) Generate text file(s) with CTHs and other information.
The file(s) created by this program are inputs to cth cluster analysis software.

Usage: cth_cluster -f spreadsheet.xls [-o basename] [-b number_of_bins | -e number of
E bins -i number of I bins] [-np | -nu | -nr] [-c] [-g] [-pa] [-tc | -tw | -tr] [-h] [-
v0|1|2]

-f spreadsheet.xls.  Spreadsheet file containing experiment info.
                     This is a required argument.
-o basename  Use basename for all input and output files.
             The final output will be basename.cth.
-c Create a .csv file.  Will create file: basename.csv
-g Create a .csv file that ggobi likes.  Will create file: basename_g.csv
-np Peak normalization. Default is mean (area) normalization.
-nu Unit normalization.
-nn No normalization, output is raw spikes/sec values.
-pa Include all periods in files and save to one file.
    Default is to include all periods and save to separate files.
    Note that the brainstem program will not be able to load .csv files
    created from these files. The clustering program can load these files.
-tc Do complete histogram curve fitting related processing using cth_curve.R.
    By default, this is not done and the output file will contain zero values
    for the curve plotting vectors if no other -t option is used.
    It can take a VERY long time to do this on the typical workstation.
    (If you really want to do this, run this on cisc5 and
    copy the output file to someplace visible to your workstation.)
-tw Create an output file that cth_curve.R can read, but do not run cth_curve.R
   The file will be basename_to_r.txt.
-tr Read the file created previously by cth_curve.R and add it to the output.
    By default, the file will be basename_from_r.txt (see next option.)
-t filename Do not automatically create a curve fitting filename from the -o option.
Read filename instead.  This lets you share a curve file.  Implies -tr.
-vn  n = 0 Exclude vagotomized experiments. This is the default.
     n = 1 Include only vagotomized experiments.
     n = 3 Include all experiments.
-h This help.
Default output: cth_[# of bins].cth
Default E bins:   50
Default I bins:   50
```

This is the command to create the allexp-100 file:

```
cth_cluster -f cth_data.xls -o allexp-100
```

When the curve fit files are available, you can include the curves with this command:

```
cth_cluster -f cth_data.xls -o allexp-100 -t allexp_from_r.txt
```

If you want to create a .cth file with a subset of the experiments in them, copy cth_data.xls to another name (e.g., cth_data_just3.xls), open it with libreoffice, openoffice, or other spreadsheet program, then remove the experiments (rows) that you do not want. If you are sharing directories between Windows and Linux, you could edit it with excel. When you save the file, be sure to save it as an xls file. The cth_cluster program cannot read xlsx files.

To create a new cth file using your edited spreadsheet using 60 bins, you would type:

```
cth_cluster -f cth_data_just3.xls -b 60 -o just3_60
```

Another way to create cth files is to pick subsets of clusters from the GUI and save them out to a .cth file. It is interesting to take a cluster of, say, all I phase CTHs and save them and then load them in as a single set and re-cluster them into several clusters.

CURVE FITTING

The curve fitting is a CPU-intensive process than can a day or more to complete. The best way to do this is a two stage process. First, create the file for cth_curve.R, like so:

```
cth_cluster -f cth_data.xls -o allexp-any -tr
```

Note that the number of bins are not important because cth_curve.R uses one tick bins. This will create allexp-any.cth and allexp-any_to_r.txt. The second file is what we need.

Copy this to cisc5 in a working area. To start the curve fitting, type this:

```
cth_curve.R allexp-any_to_r.txt allexp-any_from_r.txt
```

The first is the input file name and the second is the output file name. They do not have to be similar.

Come back tomorrow. Once cth_curve.R completes, you can copy allexp-any_from_r.txt back to your local work area. If you wanted to, for example, do a run using 30 bins, you would type this:

```
cth_cluster -f cth_data.xls -o allexp-30 -b 30 -t allexp-any_from_r.txt
```

DETERMINISTIC ARCHTYPES

Each cluster has a centroid that represents the average of all of the CTHs in the cluster. This is called an archetype. The centroids for all of the CTHs can be exported to a .type file. This file can then be used to create clusters using the archetypes. The goal of this is to create a set of firing pattern types that can be used in software not included in this package to objectively classify CTHs. Historically, the types had names such as E Aug and I Dec and so forth. Assignment to one of the types was done by a person which entails an element of subjectivity. It also is not always a repeatable exercise, in that the same person later, or another person, may assign a CTH to same type. The goal of the deterministic archetypes is to create an objective classification procedure that is repeatable.

Of course, the elements of subjectivity and repeatability now resides in creating the set of archetypes. The clustering algorithms are repeatable, but all of them require a person to decide the number of clusters and on which algorithm generates a "good" set of archetypes.

The 1.3.0 release includes 100vago_8i_8lrm_8e.type. This was produced from all of vagotomized experiments in the current .xls file. It includes 8 predominately I phase archetype CTHs, 8 low respiratorily modulated archetype CTHs, and 8 predominately E phase archetype CTHs.

WORKING OFFLINE

If you are not on the lab's local network, say, working on a laptop at home, you can copy all of the files you need to your laptop. I do this both to have perhaps faster access to the files and also so I can work on the clustering software at home. It is a bit of work. You have to open the spreadsheet to find out where all of the files are located on cisc3. Make a subdirectory in your local cth directory, say, cth_files. Copy the files listed in the spreadsheet from the paths and file names to cth_files/. Then create a new copy of the spreadsheet, e.g., local_cth_data.xls. Open it and modify the spreadsheet paths to point to locate directory where you copied the files. You need to specify absolute file names, such as /home/you/cth/cth_files/. You will note that there are several paths in the master spreadsheet. None of the files have the same name, so the files do not have to be in separate directories, the path cells in the spreadsheet can all be the same. The complete set of files at the time of this writing use about 4.6G of disk space.

KNOWN ISSUES

The octave plot windows offer the option to save the plot to a file. This silently fails. This is a known bug and a future version of octave will fix this. For now, the best you can do is do a screen capture and save it to a file.

If you accidentally ask for 500 clusters, there is no simple way to abort the process. You will hit a max-number-of-windows limit error from the window manager before you actually complete the clustering. You may want to kill the clustering program from the command line or with a task manager.

You can click on buttons and other things in the GUI while octave does its work. These are queued up and will eventually be responded to, but there may be some unexpected actions. Clicking fifty times on the Create Plots button is probably not a good idea. A typical way of dealing with this is to disable inappropriate controls when the program is busy. At this time, the GUI program does not know when the octave program is busy, so it is hard to know when to enable or disable parts of the GUI to avoid this situation.

The cth_cluster program assigns a sequence number to each cth. The cthgui program uses this as a key to keep track of the CTHs. The order of the sequence numbers is determined by the order of the input files in the input spreadsheet. If a file is removed from the spreadsheet, the sequence numbers for CTHs will no longer be the same for CTHs after the deleted file, so it is very difficult to compare different runs using subsets of the files. This is not a problem if you are adding new files at the end of the spreadsheet.