CPQt

1-D change point modelling with constant basis functions

CPQt is an application which allows you to model an unknown number of abrupt changes in the mean value of one or more data series, with either known or unknown errors on the individual data series. When more than one data set is used, it is assumed that the changepoint locations will be the same for all data sets. The mathematical approach underlying CPQt is transdimensional Markov Chain Monte Carlo (MCMC) and the details are given in Gallagher et al. (2011), referenced below.

Gallagher, K., Bodin, T. Sambridge, M, Weiss, D, Kylander, M, and Large, D. (2011) Inference of abrupt changes in noisy geochemical records using Bayesian transdimensional changepoint models, Earth Planet. Sci. Letts., 311, 182-194

PLATFORMS AND INSTALLATION

CPQt can run on both Macintosh and Windows and the installation process requires making sure certain library files are installed in the correct place.

Macintosh

Copy the 3 framework directories either to the top level directory

/Library/frameworks/

or to the equivalent in your home directory

MyHomeDirectory/Library/frameworks/

Windows

You need to keep all the .dll files in the same directory as the executable (CPQt.exe)

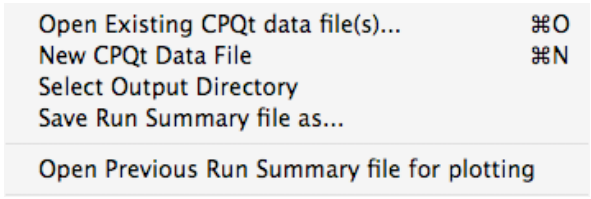
In the descriptions that follow, all of the GUI illustrations shown are as seen on a Macintosh.

Currently, CPQT is limited to deal with a maximum of 20 data files for a given run.

CPQT MENUS

The main menu bar is shown below

FILE MENU



If you have previously opened files with CPQt, you will also see a list of the most recently opened files (up to 10) at the bottom of this menu.

Open existing CPQt data file(s)…

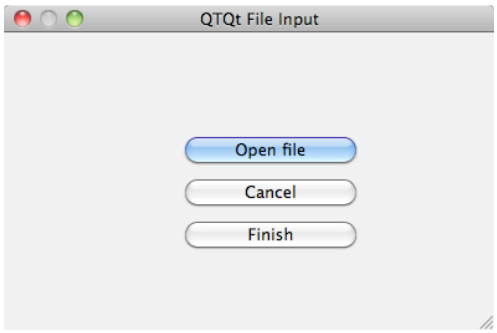
This allows to open one or more data files for a particular run. The data files are simple text files with the following format

Line 1 : the number of data points (N)

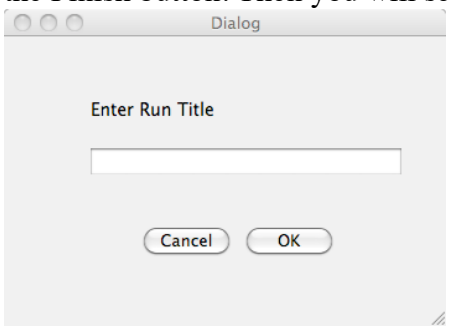
Line 2 : either 1 or 0 as a flag to indicate whether there are errors (=1) or not (=0) in the data file

Lines 3 to N+2 : 2 or 3 columns with location (e.g. x, time), data value, data error. The columns can be separated either by spaces or a tab.

You will see a window as below



You can keep opening files (or open more than one at the same time), until you click on the Finish button. Then you will see the window below,



This allows you to give the run a particular name and this name will be used in output files (including graphical output). The default for the run name is CPQt.

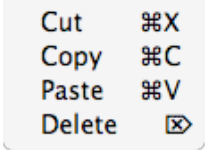
New CPQt data file
This allows you to paste data (either 2 or 3 columns as above) into a spreadsheet window and save the file for later use. You can use the **Edit** menu to paste, cut, etc.

Select Output Directory
This lets you choose a directory for saving output from CPQt. By default, the output directory is set to the directory containing the last data file opened before a run.

Save Run Summary file as…
This lets you select a name for the output file for a given CPQt run. The default is the run name.

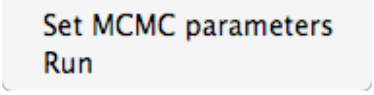
Open Previous Run Summary file for plotting
This allows you to regenerate plots from a previous CPQt run. You select the file and then use the plot menu as desired.

EDIT MENU

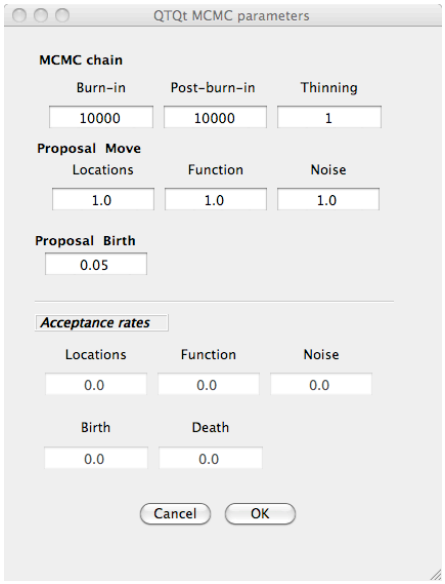


This is a standard edit menu with cut, copy, paste and delete selections. Note that sometimes the key substitutes shown to the right for these operations may not work with the Windows application, so you need to select the option explicitly from the menu.

MCMC RUN MENU



Set MCMC parameters
This allows you to set various run parameters required for the MCMC sampling.
You will see the window below,

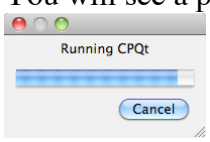


Burn-in : the number of iterations which will be discarded from the final inference (as the algorithm tries to reach a stationary sampling distribution)
Post-burn-in : the number of iterations which will be used for the final inference (under the implicit assumption that the algorithm has reached a stationary sampling distribution)
Thinning : the models from the post-burn-in iterations are resampled with a frequency given by the thinning parameter (e.g. if thinning = 5, the final inference will be based on every 5th post-burn-in model).

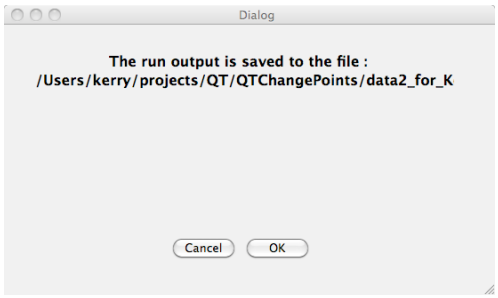
Proposal Move
There are 3 possible move perturbations to a given model which sample a changepoint **Location** parameter (e.g. distance, depth, time…), the **Function** value (which is more or less the mean of the data between 2 changepoints for a given dataset), and the **Noise** which is the standard deviation of the unknown error (normal) distribution for a given data set (this is only relevant if no errors have been input with the data values).
The parameter values you enter are the proposal scale parameters used to perturb the current model to produce a proposed model (see Gallagher et al. 2011 for details).
The appropriate values for these 3 proposal scales are selected through a few trial runs, and monitoring the acceptance rates (shown at the bottom of the MCMC parameter dialog above). The pre-run values of the acceptance rates will be zero.

The birth proposal value is used to decide how to choose a new changepoint. Currently this value does not need to be changed.

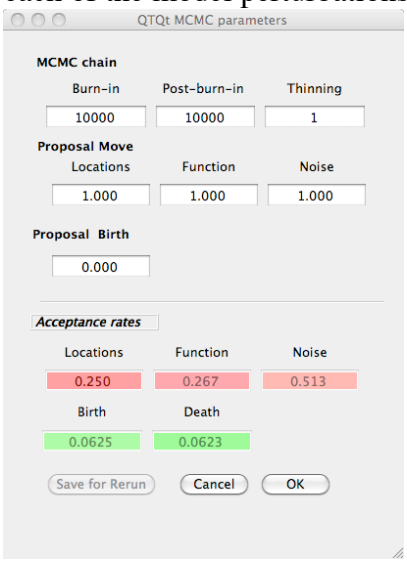
Run
Having entered the MCMC run parameters and proposal scales, you are now able to run the MCMC sampler.
You will see a progress dialog as below, indicating the run progress.



When the run is finished, another dialog window will appear stating that the output will be saved to a file with a given name (depending on what you have selected previously),a as shown below



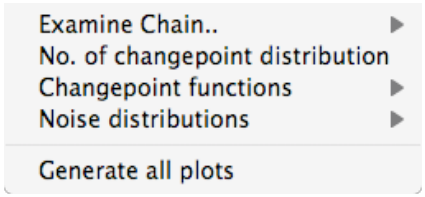
Once you click OK, you will then see a window summarising the acceptance rates for each of the model perturbations (move, birth, death), as below



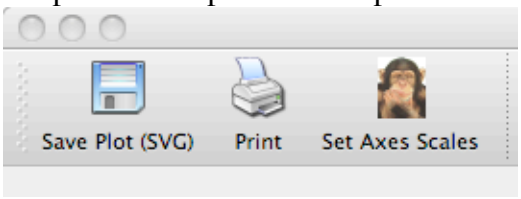
The move acceptance rates for the 3 parameter types are highlighted in red, and typically these values should be around 0.2-0.6. As a rule of thumb, you should increase the proposal scale if the acceptance rate is too high (because we are most likely moving very slowly around the model space), and decrease the proposal scale if the acceptance rate is too low (as we are most likely proposing models far from the current model, at least in terms of how well we can fit the data).

The birth and death acceptance rates cannot readily be tuned, but should be more or less the same if the sampler is behaving well. Also, their values can be relatively low, relative to the 0.2-0.6 range mentioned above for the moves.

PLOTTING MENU



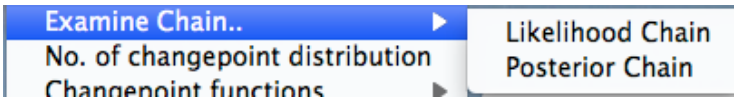
All plots have options in the plot window.



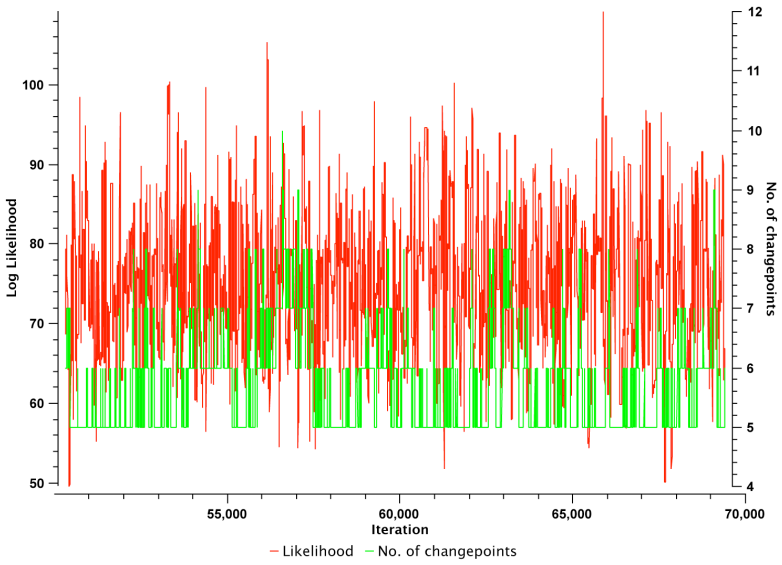
You can manually save the plots (in SVG format), print the plot (including printing to a pdf file) and change the default scales for the axes.

Examine Chain..

This lets you examine the post-burn-in sampling in terms of the number of changepoints and either the log likelihood or log posterior, with the submenu shown below



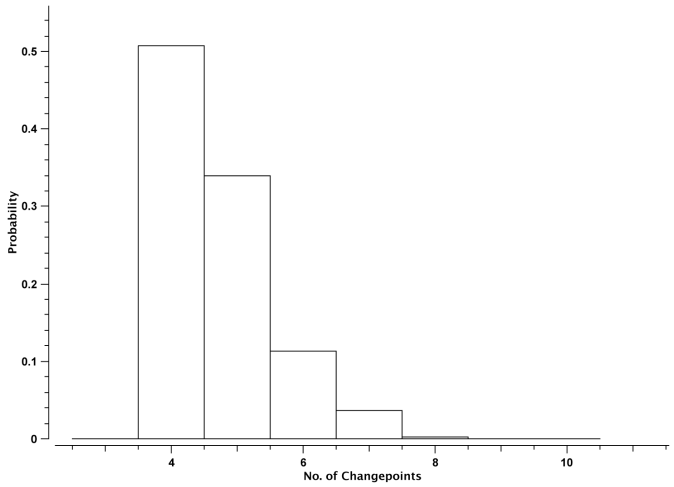
A typical plot showing the log likelihood (in red) and number of changepoints (in green) as a function of iteration is given below



As another check on the performance of the sample, neither of these functions should show an obvious trend with iteration (e.g. overall increase with iterations), while they should show reasonable variation about the mean value (e.g. increasing or decreasing the number of changepoints regularly).

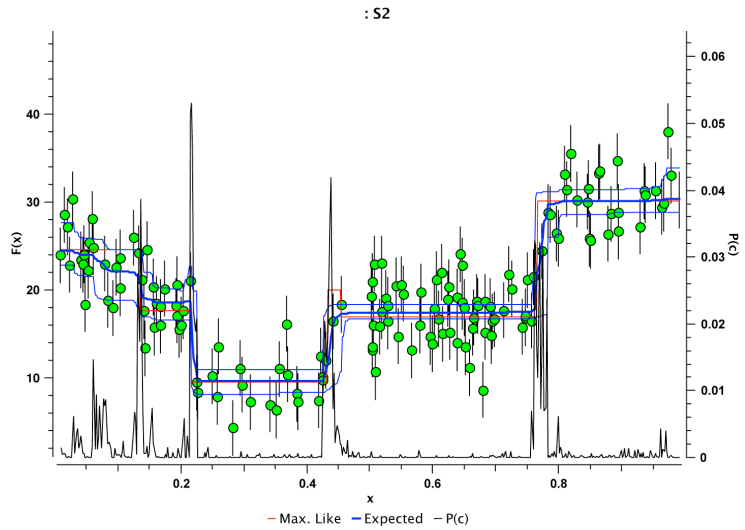
No. of changepoint distribution

This summarises the sampling of the number of changepoints as a histogram, normalised to have an area of 1 (so it can be treated as the posterior probability distribution).
An example is given below



Changepoint functions

For a given dataset, this summarises the location of changepoints (which are the same for all datasets in a given run), the inferred function values between adjacent changepoints and the observed data values with either the input errors or the inferred error.
An example is given below



The black curve with various peaks reflects the changepoint locations and their relative probability (so the higher the peak, the more probable the changepoint). The probability value is given on the right hand axis.
The thick blue curve is the expected function value ($f(x)$), which is effectively the weighted mean of all the models accepted during the MCMC sampling, and the thinner blue lines show the 95% credible interval range above and below the expected model. The red curve is the maximum likelihood model (that is the one that gives the best fit to the data).

Noise distributions
This summarises the inferred distribution of the standard deviation (square root of the variance) of the noise for each data set as a histogram, normalised to have an area of 1. It is assumed that the noise is characterised by a normal distribution with a given standard deviation. The MCMC sampler then produces a distribution of this standard deviation.