Match 2.0 User’s Guide

1 Copyright

Any questions regarding this license or the operation of this software may be directed to Lorraine Lisiecki <lisiecki@alumni.brown.edu>.

2 References

If you publish results generated by this software please cite the following references:

Lisiecki, L. E, and T. D. Herbert (2007), Automated Composite Depth Scale Construction and Estimates of Sediment Core Extension, *Paleoceanography*, doi:10.1029/2006PA001401.

Lisiecki, L. E. and P. A. Lisiecki, The application of dynamic programming to the correlation of paleoclimate records, *Paleoceanography*, 17(D4), 1049, doi:10.1029/2001PA000733, 2002.

3 Acknowledgments

This research used samples and/or data provided by the Ocean Drilling Program (ODP). The ODP is sponsored by the U.S. National Science Foundation (NSF) and participating countries under the management of Joint Oceanographic Institutions (JOI) Inc. Funding for this research was provided by a Schlanger Ocean Drilling Fellowship, which is part of the NSF-sponsored U.S. Science Support Program (USSSP).

4 Description

This program is designed to correlate paleoclimate records and sediment core data using dynamic programming, which finds a globally optimal fit. Penalty functions allow the user to ensure that the solution be physically realistic. An additional program, Autocomp, generates a composite depth scale for matched sediment core data.

5 Questions and Updates

Any questions regarding this software may be directed to Lorraine Lisiecki <lisiecki@alumni.brown.edu>. Also send email to this address if you would like to be notified of any updates to the software and to report any bugs that you may discover. Updates will be posted on http://Lorraine-Lisiecki.com.

6 Software Specifications

6.1 *System Requirements*

The match program is written in C++. It requires a command line interface to run. It should run on any Unix-style operating system, in Cygwin under Windows, and on Macs running MacOS 10 and should be easily portable to any platform with a C++ compiler. The matching algorithm is fairly memory and computation intensive, especially as the number of matching intervals increases.

A graphical user interface for the Match program is available and runs in Matlab version 6. However, the Match program can be operated using only text files for configuration.

6.2 Downloading and Compiling the Software

The Match software package is available for download from from http://Lorraine-Lisiecki.com/match.html . To extract the source files, run "gunzip -dc match-2.3.tgz | tar -xvf -". To generate an executable version of match type "make" in the Match directory containing "match.cc" and "Makefile". The enclosed makefile should allow Match to be built on any GNU-based compiler platform as well as many others. You may need to modify the makefile if your compiler is not named "c++" or requires different command line options.

From the Examples directory the command line to run the program should be something like "../Match/match –v example.conf". The "-v" is optional and will send log information to the screen in addition to the log as the program runs. A second "-v" will output a huge amount of information to the screen that will be unnecessary in most situations.

7 Match Algorithm

7.1 *Original Publication*

A copy of the article published in Paleoceanography, which explains how the algorithm works, is included in this directory as a pdf document (lisiecki02.pdf). For alignment of multiple cores, please also see *Lisiecki and Herbert* [*Paleoceanography*, 2007].

7.2 *Modifications*

The algorithm has been modified to include special behavior for record gaps. The matched size of gaps is no longer limited and overlaps at gap points are even considered. A new gap penalty is used to constrain gap sizes.

8 Input Files

8.1 *Data Files*

Data files should be in an ASCII text format with columns of numbers separated only by spaces or tabs. Text (including column headers), blank lines, and missing numbers are not allowed in these files.

8.1.1 Core data

If the data are core measurements, data files must consist of three columns. The first (leftmost) column should contain integer core numbers. The next column of the file should contain the depth of the measurement, and the rightmost column is for the measurements to be matched (e.g. reflectance). If depth values are not in order within each core, the Match software will automatically sort them and notify you that sorting has been performed. Duplicate time or depth values within a core will be averaged together, and you will be informed of the number of duplicates found. Core data may be matched to non-core data.

8.1.2 Non-core data

If the data are not measurements from cores, the data files must contain only two columns. The first (leftmost) column should contain the depth/age of the measurement, and the second column should contain the measurements to be matched (e.g. isotopic value). If depth/age values are not in order, the Match software will automatically sort them and notify you that sorting has been performed. Duplicate time or depth values will be averaged together, and you will be informed of the number of duplicates found. Known gaps (e.g. hiatuses) may be added in the graphical user interface or by the creation of a gap file (to be described later). The software will output non-core matches in the core format by adding a column to the left that represents a core number. You may put non-core data in the core format by adding a column on the left that has the same integer value for all measurements.

8.1.3 Data pre-processing

If two different data types are being matched, it is often necessary to normalize the datasets before running Match. Normalization entails subtracting the mean from each series and then dividing by the series' standard deviation to create a series with a mean of 0 and a standard deviation of 1. Normalization can be helpful for very similar series as well because the effectiveness of penalty values can depend on the means and standard deviations of the series. I have included with this software package a matlab program (normalize.m) that normalizes series and writes them out in the correct format for Match.

8.2 *Configuration Files*

A configuration file (typically with the extension “.conf”) tells Match which files to read and what values to use for the alignment parameters. The configuration file may be generated using the graphical user interface Match_gui (See Section 10.1.) or using a text editor and following the format described in Section 10.2. The Match program can also make use of tie points provided in a tie file and add gaps to the data through a gap file. These are also text (ASCII) files which can be generated by Match_gui or by the user. (Match_gui can greatly reduce the effort involved in generating tie points.)

9 Output Files

9.1 *Aligned data*

The program will output matched versions of the two series in files with '.new' appended to the original file name (e.g. the matched version of 'signal.dat' will appear as 'signal.dat.new'). These new files contain all of the data points in the overlapping depth/time range of the two series. The time/depth of the points in the signal will be transformed to the coordinates of the target. Likewise, the time/depth of the target will be transformed to the coordinates of the signal. These two files have the format of core-data input files, but contain a row of “NaN NaN NaN” at core gaps.

9.2 *Match description*

This file contains a “definition” of the produced match. Its filename is set in the configuration file. (Match_gui assigns it the same name as the configuration file, but with a “.match” extension instead of “.conf”.) Four columns give the signal core number, the signal depth, the target core number, and the target depth at every position for which the matching speed (i.e. relative sedimentation rate) changes. These points represent the positions of interval matches generated by the program, not actual data points. Data may be converted from one depth/time scale to the other by linear interpolation between the points in the match file. The match file can also be used to calculate the relative “sedimentation” rate between the two datasets.

9.3 *Log*

The program automatically generates a log file with a name set in the configuration file. (Match_gui assigns it the same name as the configuration file, but with a ".log" extension instead of ".conf".) The log contains the configuration state used to perform the match as well as information about the signal and target and the score components of the resulting match.

10 Configuring and Running Match

The matching program uses a configuration file to get all the parameter values and data file names that it needs to run. A sample configuration file "example.conf" is included in this directory. Configuration files can be modified directly with a text editor or with "match_gui," a user-friendly graphical user interface that runs in Matlab version 6 or 7. Match_gui is especially helpful for the creation of matching "tie points."

10.1 *Graphical user interface*

First, start Matlab and set Matlab's current directory to the Examples directory (or folder). At the Matlab command prompt, add the directory containing the user interface (GUI) to Matlab's path. The command will be something like "path(path,'c:\cygwin\home\username\Match-2.3\GUI')". To start the user interface, type "match_gui" at the command prompt and a window should open containing buttons, text boxes, and graph axes. The main Matlab window will display any warning or error messages that Match_gui generates. As with most programs, it may be wise to save your configuration file often in case you accidentally kill the program or lose some of your configuration information.

10.1.1 Getting Started – An example

To load the example match, click on the button labeled "Load Configuration." A smaller window will open that contains the list of files configuration files (and subdirectories) in your current directory. Double click on "example.conf". The smaller window should disappear, and the main window will now contain the graphs of two data sets and text in many of the text boxes. This is reflectance data from two drill holes from ODP Leg 154, Site 929. Press the "Set axes to endpoints" button at the top of the screen to better see the part of the data used in the match.

10.1.2 Graphs

The top graph, plotted in pink, shows the signal, which is the data to be aligned to the target, plotted below in blue. (In actuality, Match generates two new files: one with the signal aligned to the target and one with the target aligned to the signal, but Match_gui will not plot the second alignment.) A red dot marks the end of each core in the data sets. Green triangles mark the start and end of the portion of the data that will be matched. (The files from which the signal and target are read as well as the begin- and end-points can be changed using the text boxes near the top of the window.) Press the button at the top of the window labeled “Set axes to endpoints” to zoom in on the data that will be matched.

On the graphs, black crosses labeled with black numbers mark tie points that have been set for this match. Each tie point on the signal will be aligned to its corresponding tie point in the target. Tie points can be very helpful when the signal and target are not very similar, especially for core data, but are not necessary to perform a match. Tie points can be added, moved, or deleted using the controls above the signal graph. User-added gaps, which are not present in this example, are marked with red crosses and numbers. Detailed information on using tie points and gaps is in Sections 12 and 13.

The limits of the graphs’ axes can be controlled in a number of ways. In addition to the “Set axes to endpoints” button at the top, there are “zoom” and “flip” buttons and a slider for each graph. The slider will only function if the corresponding graph contains only a subset of the data, due to zooming, etc. Clicking on the trough of the slider moves the axes by more than clicking on the arrow buttons. The zoom button keeps the minimum x value the same, but cuts the range of x-axis in half. The y-axis is adjusted to fit the data. Text boxes at the bottom of the window allow the axes to be set manually. They should contain four numbers separated by commas: minimum x, maximum x, minimum y, maximum y. (Warning: Setting the x-axis limits beyond the range of the data will cause the sliders to stop functioning and produce error messages in the Matlab window. When the axes are again within the range of the data, the sliders will function again and the error messages will stop.) Pressing the synchronize button sets the axes of the signal to those of the target. While this button is depressed, the program will automatically change both axes whenever any of the axes buttons, sliders, or text boxes are used. Press the button a second time to end synchronization. (Note: The synchronize button also controls the results plot generated after matching.)

10.1.3 Generating an example match

Pressing the “Generate File” button saves your match configuration to a configuration file with the name displayed in “Output filename” text box directly above the button. In this case the file will be “example.conf”. As you experiment with the program, you may want to change the output file to “my_example” so that you will not overwrite the original example. After changing the filename and pressing the button, you must run Match from your computer’s command line (in Cygwin if using a Windows machine, NOT the Matlab window). Instructions for compiling Match are in Section 6.2. From the Examples directory, type the command line "../Match/match –v example.conf". If you get an error about the match command or the configuration file, you or your file is probably not in the correct directory. As the program runs it will display information about the match in the command window. (If you prefer not to see this information, leave out the “-v”.) When a new command prompt appears, the program has finished running. (This may take several minutes.) The results of your match are saved in several files (See Section 9: Output Files.) and can be viewed in Match_gui.

10.1.4 Viewing results

To see the results of your match, press the “Plot Results” button in Match_gui. This button will open a new window each time you press it. To close these windows, press the ‘x’ in the window’s top right corner, select File->Close in the window’s toolbar or press Ctrl-W on your keyboard while the window is selected. The axes of the results graph may be changed using the text box at the bottom of the Match_gui window or changing the axes of the signal or target graphs while the “synchronize” button is depressed.

The results window contains three graphs. The top graph shows the aligned signal and the target as well as the target positions of any tie points used in the match. The middle graph is magnetic susceptibility data from the same sediment cores that was also used by Match to improve the alignment. (To view the unmatched magnetic susceptibility data, press the “Replot” button in the top left of the main Match_gui window.) The bottom graph in the results window shows the matching “speeds” of the alignment (i.e. the relative sedimentation rate of the signal). A speed greater than one means that the signal is compressed in that part of the alignment, and a speed less than one means that the signal is stretched in the alignment. The “speed” and “speed change” penalties are used to constrain this aspect of the Match alignment.

10.1.5 Fine-tuning a match

If you are unsatisfied with your initial match, try changing the match’s parameter and running Match again. Tips for modifying the parameter values to obtain the best results are in Section 11: Match Parameters. After changing one or more parameter values, press “Generate File,” run Match at the command line again, and then plot the new results. If you do not want to overwrite your previous match, enter a new output filename before generating your configuration file. Plotting the new results will not erase the plot of your previous match, but you will no longer be able to control the axes of your old graph using Match_gui. If you did not overwrite your old match, you can replot that match by reloading the old configuration file and pressing “Plot Results” again.

10.2 *Text-file interface*

10.2.1 Configuration file

Each parameter value is on a separate line and is preceded by a keyword. The parameters may be present in the configuration file in any order. Some of the parameters have default values as indicated below that will be used if that parameter's keyword is not present in the configuration file. The other parameters must be given in the configuration file. No capitalization should be used in any of the keywords, and the keywords should be separated from their values by a space or tab character. Filenames should not have quotation marks. Comments begin with a # sign and can be on their own line or at the end of a line. It will probably be easiest to create your own configuration files simply by modifying example.conf.

Keyword	Type	Optional	Description
#	any	Y	Comments
series1	string(s)	N	Filename(s) of series (signal) to match to target
series2	string(s)	N	Filename(s) of 2nd series (target)
tiefile	string	Y	Filename of tie points
gapfile1	string	Y	File containing gaps to insert into series1
gapfile2	string	Y	File containing gaps to insert into series2
logfile	string	Y	Filename to which Match's log is written
matchfile	string	N	Filename to which Match's match is written
begin1	float	N	Depth or time value for the beginning of series 1.
end1	float	N	Depth or time value for the end of series 1.
begin2	float	N	Depth or time value for the beginning of series 2.
end2	float	N	Depth or time value for the end of series 2.
numintervals1	integer	N	Number of intervals into which series 1 is divided.
numintervals2	integer	N	Number of intervals into which series 2 is divided.
speeds	integer array	N	Possible interval matching ratios. (Numerator and denominator separated by colon, ratios separated by comma or space).
nomatch	float	N	Penalty for each interval that is not matched to an interval(s)in the other series.
speedpenalty	float	Y	Penalty for matching at speeds away from target.
targetspeed	ratio	Y	Integer ratio in same format as speeds (e.g. 1:1)
speedchange	float	N	Penalty for changes in matching speed between adjacent intervals.
tiepenalty	float	Y	Coefficient multiplied by the square of the distance between two tie points.
gappenalty	float	Y	Coefficient multiplied by the square of the difference between matched and expected gap size.

10.2.2 Running Match

Match is run from your computer's command line (in Cygwin if using a Windows machine). Instructions for compiling Match are in Section 6.2. From the directory containing your configuration file, type the command line "../Match/match –v example.conf" if example.conf is the name of your configuration file. If you get an error about the match command or the configuration file, you or your file is probably not in the correct directory. As the program runs it will display information about the match in the command window. (If you prefer not to see this information, leave out the "-v".) When a new command prompt appears, the program has finished running. (This may take several minutes.) The results of your match are saved in several files. (See Section 9: Output Files.)

11 Match Parameters

Parameters are used to guide the Match program into creating a physically realistic and suitably precise match. The begin, end, speed, and interval parameters define absolute limits for the match. However, the penalty parameters allow the user to encourage certain behaviors in the match while still permitting the program to deviate from those guidelines when justified by sufficient evidence from the signals. Applicable penalties are applied when each group of intervals is considered, and the sum of each type of penalty along the path of the best match is output by the program.

11.1 *Begin and end values*

These values give the minimum and maximum possible depth/time values of each data series permitted in the match. For best results, these values should be set as close to the first and last corresponding points in the series as possible. However, users should be careful because Match will not consider any data outside of a series' begin and end points. The match generated may or may not extend to these points, and the endpoint (a.k.a. "no match") penalty is used to control how strongly Match should try to extend the match to the given begin and end values. Match will return an error if these values are not within the depth/time range of their respective series.

In Match_gui the begin and end values will default to the start and end of the series. To change these parameters enter a new value in the appropriate text box under the labels "Begin" and "End." The beginning points are displayed as left-pointing green triangles in the signal and target graphs, and the end points as right-pointing green triangles. Pressing the "Set axes to endpoints" button at the top of the window uses these values for the x-limit of each graph.

In the configuration file these values must be provided following the keywords "begin1" and "end1" for series 1 and "begin2" and "end2" for series 2.

11.2 *Endpoint or "no match" penalty*

This penalty encourages Match to maximize the overlap between the signal and the target. It is set in the "Endpoint" penalty box in Match_gui or following the keyword "nomatch" in the configuration file. An extremely high value will force the program to use the begin and end parameter values as the start and end points of the match, so that the begin (end) point of the signal will match the begin (end) point of the target. An endpoint penalty value that is too low will cause Match to produce a too small overlap or to return an error message because the begin point of one series is matched to the end of the other.

11.3 *Number of intervals*

Each series requires a parameter value that determines the number of intervals into which it will be divided for during the matching process. These evenly spaced intervals are the basic matching units and determine the scale at which the sedimentation rate is allowed to vary. (See the speeds parameter below). A larger number of intervals will allow Match more freedom to optimize the fit of the two series, but the runtime and memory usage of the program goes as the product of the two numbers of intervals. Reducing the number of intervals is one way to prevent relative sedimentation rates from varying too rapidly. On a modern PC, each series can reasonably be divided into a few hundred to ~1500 intervals. If the endpoints of the match are fairly well constrained, it is generally best to use the same number of intervals in each series. If one series is expected to be longer than the other (and the preferred matching speed is 1:1), divide the longer series into proportionally more intervals.

In Match_gui the value of this parameter for each series can be set using the text boxes under the label “# of intervals”. The configuration-file keywords for these parameters are “numintervals1” and “numintervals2.”

11.4 *Speeds*

The speeds are the ratios at which the intervals of the two series can be matched. For example, a speed of 3:2 means that three intervals of the signal are matched to two intervals of the target. If the interval sizes (length of series divided by number of intervals) are the same for the two series, the speeds may be thought of as the relative sedimentation rate between the two series. If the signal is in depth units and the target is in time units then the speeds are proportional to the sedimentation rate of the signal. The matching algorithm considers the match of each pair of intervals between the signal and the target at each speed given by the “speeds” parameter. Therefore, runtime increases linearly with the number of speeds listed.

The “speeds” parameter is a list of ratios formatted with colons and separated by commas and/or spaces, e.g. “1:3, 2:5, 1:2, 3:5, 2:3, 3:4, 1:1, 4:3, 3:2, 5:3, 2:1, 5:2, 3:1”. Speeds may only contain integers. Generally, the highest density of speeds should be near the preferred or “target” speed (usually 1:1- See below.). For example, you may also wish to include 4:5 and 5:4. If the (relative) sedimentation rate of your series is highly variable, you may need to include 1:4 and 4:1 or larger ratios.

The “Set speeds” button allows the user to change the list of possible speeds from their default values in Match_gui. In the configuration file, the list of speeds follows the keyword “speeds.”

11.5 *Preferred or “target” speed*

This is the speed or ratio at which you expect the two series to be matched on average. It is used only by the speed penalty, which allows Match to penalize matches that contain large deviations in sedimentation rate. If the begin and end points are set near the expected ends of the match and both series have the same number of intervals, then the preferred/target speed should be 1:1.

In Match_gui the preferred speed may be selected from the list of all possible speeds from a pull-down menu to the right of the “Set speeds” button. It is set in the configuration file by using the keyword “targetspeed.”

11.6 *Speed penalty*

The speed penalty parameter controls how easily the Match program may use speeds, or relative sedimentation rates, other than the preferred or target speed. When Match considers the alignment of a group of intervals, i.e. a subset of the series, it applies a speed penalty based on the difference between the considered speed and the preferred speed (usually, 1:1). A value of zero for this parameter will prevent the program from applying this penalty, so that no speed is preferred over another. A very high value will strongly discourage the match from containing relative sedimentation rates that are much larger or much smaller than the preferred. The matching speed may also be restricted in slightly different ways by reducing the range of matching speeds or setting a high speed change penalty. The speed penalty should be preferred if there is a physical reason to believe that the two series have very similar sedimentation rates but you would still like to allow frequent small changes in speed.

The speed penalty is set in the graphical user interface by entering a number into the text box labeled speed in the “Penalties” section or in the configuration file using the keyword “speedpenalty.”

11.7 *Speed change penalty*

This penalty controls how easily the Match program may change speeds (relative sedimentation rates). A very high speed change penalty will encourage the program to pick one relative sedimentation rate and stick with it for as long as possible, and a small value will allow the sedimentation rate to change rapidly and frequently. Because the speed is defined in terms of groups of intervals, e.g. three signal intervals matched to four target intervals, the size of each interval (as defined by the number-of-intervals parameters) places an absolute limit on how quickly the relative sedimentation rate may change. The speed change penalty only discourages larger deviations in relative sedimentation rate by creating a smoother sedimentation rate curve. A combination of the speed and speed change penalties is the most effective way to prevent large spikes in relative sedimentation.

The speed change penalty is set in the graphical user interface by entering a number into the text box labeled speed change in the “Penalties” section or in the configuration file using the keyword “speedchange.”

11.8 *Tie Point penalty*

If a user defines any tie points that they feel should match between the signal and the target, the tie point penalty controls how strongly the program should try to make the tie points match. The penalty is a coefficient applied to the square of the distance between the tie points in a match. Therefore, a sufficiently high penalty will force tie points to correspond as closely as possible (within the constraints provided by the number of intervals) and a very low penalty will allow the tie points to have a larger separation. Because this penalty is applied to the difference in depth or time, its effectiveness will be affected by the depth/time scales of the signal and target in addition to the scale of the dependent variable.

The tie point penalty is set in the graphical user interface by entering a number into the text box labeled “Tie Point” in the “Penalties” section or in the configuration file using the keyword “tiepenalty.”

11.9 *Gap penalty*

If either the signal or target contains any gaps, from the series files or a separate gap file, the gap penalty is used to constrain the size of the gap in the match. The penalty is a coefficient applied to the square of the difference between the gap length in the original file and in the match. Therefore, a sufficiently high penalty will force gap to have the same size it does in the original file (within the constraints provided by the number of intervals) and a very low penalty will allow gap to have any size. Because this penalty is applied to the difference in depth or time, its effectiveness will be affected by the depth/time scales of the signal and target in addition to the scale of the dependent variable.

The gap penalty is set in the graphical user interface by entering a number into the text box labeled “Gap” in the “Penalties” section or in the configuration file using the keyword “gappenalty.”

11.10 *Auto-calculate penalties*

The button labeled “Auto-calculate” in the penalties section of the graphical user interface provides an estimate of what value to use for each penalty. These values should only be used as guides to the appropriate order of magnitude for these parameters based on the mean and variance of the signal and target and, in the case of tie and gap penalties, the depth/time scale of the target. Users of this feature should run an initial match with these penalty values and then make adjustments to them based on the results of the match. There is no guarantee that these values will be appropriate for every type of data. In particular, the gap penalty value is geared for use with inter-core gaps from ocean sediment cores. Also, the tie point penalty assumes that the user is quite certain that the provided tie points should match.

No auto-calculate feature is available for the outside of the graphical user interface, but below are the formulas that the auto-calculate feature uses.

Endpoint (a.k.a. NoMatch) = 15(s + 4m)
Speed = 1.5(.2s + 4m)
SpeedChange = 3.5(.15s + 4m)
Tie = 5000(s/d + 4m)
Gap = 1000(s/d + 3.2m)

The variable *s* is the square of the larger of the two standard deviations for the signal and the target; *m* is the square of the difference in the means of the two series, and *d* is the difference between the begin and end values of the target.

12 Tie Points

A tie point tells the Match program that a certain point in the signal should match a corresponding point in the target. However, Match will not assign these correlations automatically; the tie point penalty determines how strongly the program will be inclined to match each pair of tie points. (See section 11.8 above.) Even with a very high penalty value set, there may be a small offset between the tie points because finite interval sizes and a limited set of matching speeds constrain the relative positions of two points in the match. Smaller interval sizes will allow Match to align tie points more precisely.

12.1 *Graphical User Interface*

In Match_gui tie points are represented as black crosses, +, that are labeled with black numbers. Each label number will occur in both the signal and the target to indicate which pairs of tie points correspond. The controls for adding, deleting, and modifying tie points are located in a box directly above the signal graph in the graphical user interface. To perform any operation involving tie points the upper radio button labeled “Tie points” at the very left of this box must be highlighted; otherwise, the buttons will operate on gaps. As with most programs, it may be wise to save your configuration file often to prevent the accidental loss of tie points or other parameters.

12.1.1 Loading tie points

If a listing of tie points between the signal and target already exists, those tie points can be loaded into the Match_gui by clicking “Load” while the tie point radio button is selected. Loading tie points automatically deletes all pre-existing tie points. The “Load” button opens a window displaying all of the files in the current directory that have a “.tie” extension. If the file your wish to load does not end in “.tie”, select “All” from the pull-down menu to show all files in the current directory or simply type in the file name. To select a file and close the window, double click on the filename in the menu, single click and press “OK” or type the filename in the white text box and click “OK.” The tie points from that file will then be displayed on the signal and target graphs. To close the window without loading a file press “Cancel.”

The tie file must be in text format and contain only numbers. If the signal or target has multiple cores, the tie file must contain four columns: signal core #, signal depth/time, target core #, target depth/time. If neither series has gaps, the tie file can contain two columns: signal depth/time and target depth/time, or all four columns with zeroes in the core number columns. Tie points from previous configuration files may be loaded; they will have the same name as the configuration file but with the extension “.tie” instead of “.conf’.

12.1.2 Adding tie points

To graphically add tie points, press the button labeled “Add” while the tie point radio button is selected. The mouse pointer will then turn into crosshairs. Use the mouse to center the crosshairs over the point in the signal for which you wish to make a tie point, and then press the left mouse button. You do not have to click exactly on the signal line in the graph. A tie point will appear on the signal at the x-coordinate of the point you clicked and the crosshairs will reappear. If the tie point did not appear in exactly the right place or you change your mind, you will be able to move it later. To finish making the tie point, center the crosshairs over the corresponding point in the target and click again. You now have a complete tie point pair, but the crosshairs will still be up. You may click on another part of the signal to begin making a new tie point or press “Enter” on the keyboard to stop adding tie points. Clicking on the wrong graph at any time will also make the crosshairs disappear. If you stop adding points while the last tie point is incomplete, the incomplete tie point will disappear. Each new tie point will be assigned a number one higher than the last, even if that tie point has since been deleted unless the “Reorder” button is pressed. (See section 12.1.5 below.)

12.1.3 Moving tie points

Clicking on the “Move” button depresses it and changes its label to “Press to lock.” While the button labeled “Move” is depressed and the tie point radio button is selected, you will be able to move any tie point. Click on the number above the tie point you wish to move and use the crosshairs to select its new position. The tie point will move to its new location and the crosshairs will disappear. To move another point, simply click on another tie point number. You may move the signal or target end of a tie point at any time, in any order, and without moving both ends. The tie point's number will not change even if you move the points out of sequence. You may leave the "Move" button depressed while performing other operations in the window. At any time if wish to prevent yourself from accidentally moving any of your tie points, press the button now labeled "Press to lock." Its label will change back to "Move" and pressing it again will allow you to move tie points again.

12.1.4 Deleting tie points

To delete a tie point, type its number in the white text box labeled "Delete #." When enter is pressed or you click somewhere else in the window, the tie point will be removed. The number labels of the other tie points will not change. If the tie point does not exist, a message in the main Matlab window will alert you.

To delete all tie points, press the 'Delete all' button on the far right. The program will ask you to confirm the deletion before executing this function.

12.1.5 Renumbering tie points

Adding, moving or deleting tie points may make the tie points' number labels non-consecutive. The labels are not changed during movement or deletion to avoid confusion. If you wish to renumber your labels to make them increase consecutively, press the "Reorder" button. The number label of each pair of tie points will be updated, and the next new tie point will be assigned the number following the last tie point number. If any tie points are crossed (not in the same order in the signal and target), the program will produce an error message and the points will be placed in consecutive order in the signal. The Match program will work with crossed tie points, but it will not be able to fit both tie points unless a gap of negative size (e.g. overlapping cores) occurs between the points.

12.2 *Text files*

To use tie points in a match, include the keyword "tiefile" in the configuration file followed by the name of a file containing the desired tie points. The tie file must be in text format and contain only numbers. If the signal or target has multiple cores, the tie file must contain four columns: signal core #, signal depth/time, target core #, target depth/time. If neither series has gaps, the tie file can contain two columns: signal depth/time and target depth/time, or all four columns with zeroes in the core number columns.

13 Gaps

A gap tells the Match program that a discontinuity occurs in the data at a certain point or over a certain interval and that this discontinuity may have resulted in a portion of the signal being missed or duplicated. Gaps are automatically inserted between core breaks but other gaps can be using the graphical interface or with gap files. Match will adjust the gap size to produce the best fit. The gap size is constrained by a gap penalty which is applied to the difference between the size of the gap in the signal and its corresponding length in the target.

13.1 *Graphical User Interface*

In Match_gui gaps are represented as red dots or hollow red squares. Red dots are used to mark gaps denoted in the input file by a change in core number; these gaps are not labeled and cannot be modified. Hollow red squares mark user-inserted gaps; these gaps are labeled with red numbers and can be moved and deleted. Any data falling inside a user defined gap is plotted in red. The controls for adding, deleting, and modifying gaps are located in a box directly above the signal graph in the graphical user interface. To perform any operation involving gaps the lower radio button labeled "Gaps" at the very left of this box must be highlighted; otherwise, the buttons will operate on tie points. As with most programs, it may be wise to save your configuration file often to prevent the accidental loss of user-defined gaps or other parameters.

13.1.1 Loading Gaps

If a listing of user-defined gaps already exists for the signal or target, those gaps can be loaded into the Match_gui by clicking "Load" while the gap radio button is selected. Loading gaps automatically deletes all gaps that may have already been added to the signal or target. The "Load" button opens a window that asks if the gaps should be added to the signal or target and then displays a list of files in the current directory that have a ".gap" extension. If the file your wish to load does not end in ".gap", select "All" from the pull-down menu to show all files in the current directory or simply type in the file name. To select a file and close the window, double click on the filename in the menu, single click and press "OK" or type the filename in the white text box and click "OK." The gaps from that file will then be displayed on the signal or target graphs. To close the window without loading a file press "Cancel."

The gap file must be in text format and contain only numbers. Each gap file must contain three columns: core #, start depth/time, and end depth/time. For a gap with no series extent, the start and end values should be the same. If a series does not have cores, the gap file should contain zeroes in the first column. Gaps created for previous configuration files may be loaded; they will have the name of the signal or target followed by an underscore and the configuration filename with the extension ".gap" instead of ".conf".

13.1.2 Adding Gaps

To graphically add gaps, press the button labeled "Add" while the gap radio button is selected. The mouse pointer will then turn into crosshairs. Use the mouse to center the crosshairs over the point in the signal or target where you wish to insert a gap, and then press the left mouse button. You do not have to click exactly on the signal/target line in the graph. A dialogue box will open and ask you to select what type of gap you wish to create.

The first four options create a gap that covers part of the series data. The data that falls inside a user-defined gap will not be used by the Match program when creating a match. This is useful if you have reason to believe that a portion of your data is corrupt. If the data is composed of cores the first three options allow you to create a gap that covers an entire core, from the beginning of a core to the point just selected, or from the point just selected to the end of a core. If the data are not composed of cores then these operations will act on the entire signal. The fourth option allows you to select another point and creates a gap between the two selected points. (Gaps may not extend across multiple cores.) The fifth option creates a gap at a single point and should be used when all data is valid but the record is discontinuous, e.g. because of a depositional hiatus. The sixth option allows the user to cancel the operation and not create a gap.

After a gap-type is selected, the gap will appear on the signal/target at the x-coordinate of the point you clicked unless the user chose no to create a gap. If the fourth option was selected, the first point selected will just show a hollow red square and the crosshairs should be used to select the other end point of the gap. The signal/target will then turn red between the two points and both ends of the gap will be labeled with the same number. All other gaps will only be labeled in one location, but any data falling within the gap will be plotted in red. Each new gap is assigned a number one higher than the last, even if that gap occurs in the other series or has since been deleted (unless the "Reorder" button has been pressed; see section 13.1.5 below). If the gap did not appear in exactly the right place or you change your mind, you may move or delete it. However, you may not change a gap's type.

The crosshairs will reappear after a gap is created unless you chose not to create a gap. You may now create another gap. If no more gaps are desired, press the "Enter" key on the keyboard.

13.1.3 Moving gaps

Clicking on the "Move" button depresses it and changes its label to "Press to lock." While the button labeled "Move" is depressed and the gap radio button is selected, you will be able to move any gap by clicking on its number. This will cause crosshairs to appear, allowing a new position for the gap to be selected. The move function works slightly differently for each type of gap. For gaps that extend to the end(s) of a core, moving the label to another position within the same core will move the interior end of the gap (unless it is an entire-core gap). Selecting a point on another core will move the gap to the new core but will preserve the gap type. If the gap has two labeled ends, either end may be moved independently, but if the new point selected is on a different core, the other end will be moved to the closest end of the new core. (Gaps may not extend across multiple cores.) Single point gaps are moved in the obvious way.

To move another gap, simply click on another gap label. You may move gaps in the signal or target at any time and in any order. The gap's number will not change even if you move the gaps out of sequence. You may leave the "Move" button depressed while performing other operations in the window. At any time if wish to prevent yourself from accidentally moving any of your gaps, press the button now labeled "Press to lock." Its label will change back to "Move" and pressing it again will allow you to move gaps again.

13.1.4 Deleting gaps

To delete a gap, type its number in the white text box labeled “Delete #.” When enter is pressed or you click somewhere else in the window, the gap will be removed. The number labels of the other gaps will not change. If the gap does not exist, a message in the main Matlab window will alert you. Gaps between cores, which are marked by red dots, cannot be deleted.

To delete all user-defined gaps, press the ‘Delete all’ button on the far right. The program will ask you to confirm the deletion before executing this function.

13.1.5 Renumbering gaps

Adding, moving or deleting gaps may make the gaps’ number labels non-consecutive. The labels are not changed during movement or deletion to avoid confusion. If you wish to renumber your labels to make them increase consecutively, press the “Reorder” button. The number label of each gap will be updated, and the next new gap will be assigned the number following the last gap number. The first gap in the target will be assigned the number following the last gap in the signal.

13.2 *Text gap files*

To add gaps in a series other than those between labeled cores, include in the configuration file the keyword “gapfile1” for series 1 gaps or “gapfile2” for series 2 gaps followed by the name of a file containing the desired gaps. The gap file must be in text format and contain only numbers. Each gap file must contain three columns: core #, start depth/time, and end depth/time. For a gap with no series extent, the start and end values should be the same. If a series does not have cores, the gap file should contain zeroes in the first column.

14 Multiple Signals

If multiple types of data (e.g., reflectance, magnetic susceptibility, and GRAPE density) are available for the each of the cores to be matched, Match can compute the square difference of each data type to find the best match. This should increase the robustness of the match and reduce the number of tie points needed. However, all data types will need to span the entire matching interval, have the same depth scale and, of course, be available for both the signal and the target. An output file will be generated for each input file.

If the data are not normalized before matching, one set of data may be dominant in the match due to the different magnitudes of the signals. In some cases, matching multiple signals may require the use of larger penalty parameters.

14.1 *Graphical interface*

Enter the filenames for the data type that you consider most useful for matching in the signal and target text boxes. Then click on the check box labeled “Multiple signals” in the upper left of the window. A new window will open that will allow you to enter the filenames for up to three more types of data. When all of the filenames have been entered, click “OK” and a new window will open that contains plots of all of the data paired by data type. Unlike those in the main window, these plots are vertical with depth/time plotted on the y-axis. All signal plots are pink and to the left of the target plots, and the filename for each data set appears above its plot. The depth axis of these plots will automatically be set to those of the signal and the target in the main window. Tie points and gaps must be added to the signal and target plots in the main window. These changes will not automatically appear in the multi-signal window, but pressing the “Replot” button next to the multiple-signal check box will update this window. All signals will automatically be displayed in the results window after a match has been found.

14.2 *Text file*

To add multiple signals using the configuration file simply list multiple filenames after the keywords “series1” and “series2.” The filenames for each series should all be on the same line and should be separates only by white space. Of course, the files must be in the same order for both series1 and series2.