Application for Browsing Constraints (ABC)
Documentation

Last Updated: June 28, 2005

Generated in the lab of Arend Sidow (arend@stanford.edu) in the Stanford University Departments of Pathology and Genetics, currently maintained by a PhD student, Greg Cooper (coopergm@stanford.edu). See the following link for more info about the lab, as well as our computational methodology for analyzing multiple sequence alignments (GERP): http://mendel.stanford.edu/SidowLab/

Code written by Senthil Arun Govindasamy Singaravelu.

Table of Contents:

Please use the following citation for discussing the ABC in a publication:

Cooper, G.M., Singaravelu, S.A., and Sidow, A. 2004. ABC: Software for interactive browsing of genomic multiple sequence alignment data. *BMC Bioinformatics* **5:**192

1. Overview

The Application for Browsing Constraints (ABC) is a Java-based GUI for interactive browsing of evolutionary rate data derived from multiple sequence alignments of genomic DNA. The basic input consists of three data sets: a multiple sequence alignment, a set of scores that describe each individual position of the alignment, and a set of annotations describing regional features of the sequences in the alignment. At all but the highest resolution, the main display window plots the score data summarized dynamically to fit in the space allowed. A mobile and scalable zoom window allows exploration, ranging from a view of the entire alignment to very small regions in which individual scores and ultimately individual alignment columns may be viewed. In addition to interactive alignment browsing, other features include:

- Mouse-over highlighting to reveal annotation names, scores, coordinates, etc
- Export of sequence, annotation, or score data in plain text formats
- Searching sequence data for particular strings
- Viewing the phylogenetic tree (if supplied) that relates the aligned sequences
- GoTo feature allowing the user to quickly bring up a particular region

Things the ABC can be used for:

- Exploration of genomic multiple sequence alignments of individual loci (up to several Mb), analyzing the relationship between annotations, such as genes, repeats, and SNPs, and quantitative data like rates of evolution.
- Identifying and isolating sequence elements in a genomic locus for downstream applications like motif-discovery, primer design, etc.
- Generating graphics that characterize a multiple sequence alignment (or region of an alignment) with respect to annotation, sequence, and score data.

What the ABC will NOT do:

- Genome-wide browsing of alignment data. The ABC was not built for this scale.
- Generation of alignment or alignment scores. The ABC is a front end only, and requires the alignment and score files to be pre-generated. While we do have a computational method for doing this (see the lab website), the ABC will not.
- Tree-building or phylogenetic analysis. While an analyst performing these tasks may certainly employ the ABC, it has no inherent phylogenetic or evolutionary analysis capabilities.

2. Getting Started

The ABC requires Java 1.4 or later, and has been successfully tested on Windows, Linux, and OS X platforms. While it can be RAM-intensive, it is generally fast and efficient on reasonably well-equipped desktop computers; we recommend it be used on a computer with no less than 256 Mb of RAM and a 1GHz processor. It was developed on, and routinely used by, machines running OS X, with 1 GHz processors and 1 GB RAM; on these machines, it can efficiently handle a 2 Mb alignment of 30 sequences.

To run the ABC, create a working directory called '*YourDir*' containing the required data and settings files (see File Inputs), add the ABC.jar file to your classpath, and invoke the ABC. On a UNIX machine, for example, simply *cd* to the directory containing the ABC jar file, and type:

```
java –cp ABC.jar ABC PathtoYourDir
```

Where *PathtoYourDir* is the file path specifying the working directory created above.

If you're dealing with a large alignment (>1 Mb, > 10 species) and find the ABC to be slow, you may wish to allocate more memory by using the following command:

```
java -Xmx100M -cp ABC.jar ABC /PathtoYourDir
```

You can replace *100* with larger values as needed or as available on your computer.

3. Zoom Levels

The fundamental units of display are ‘zoom levels’, which are interactive and display score and annotation data for a defined region of the alignment. Within the main viewing rectangle (Fig 1), the score data for the defined region is displayed in one of two forms, a histogram or wiggly plot, depending on the resolution of the current zoom level (see Score Displays). Above the main viewing rectangle, four or more annotation tracks are visible, identifying the locations of sequence features like genes and repeats. Above these annotations, three values are displayed: Position, corresponding to the first and last alignment coordinates of the zoom level; Width, or the number of alignment coordinates being displayed; and Resolution, defined as the number of scores plotted per pixel.

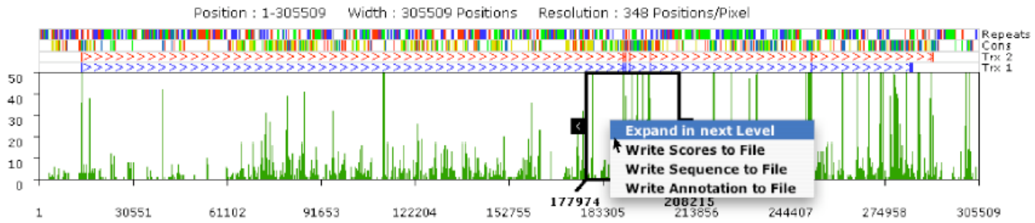


Figure 1. A zoom level showing scores and annotations for an alignment of length 305,509 positions; the main window displays a green histogram describing regional alignment scores; four annotation tracks highlight specific features of the aligned sequences; the black box can be used to zoom in to a specific region of interest; right-clicking within the black box brings up a menu to zoom in or output data for that region.

A black box will be visible within the main display, and is the mechanism for moving about the alignment; immediately below this can be seen the width, start, and stop coordinates of the box. Moving and zooming can be done by dragging, resizing, and right-clicking this box:

1. To move the rectangle along the length of the main display, left-click anywhere within the black box and drag it in either direction. The start and stop coordinates on which the box rests will update in real-time.

2. To resize the box, click and drag either of the handles located on its sides. The handles are shown as black squares with white arrows. Dragging one of the sides will shrink or expand the black box accordingly, holding the opposite side of the box fixed. Note that the width and start or stop values of the box will update in real-time.

3. Once a region of interest is chosen, a menu will appear by right-clicking anywhere within the black box. The options consist of:

- Expand in next Level – This option will generate a new zoom level immediately below the current level, defined by the start and stop coordinates on which the black box is resting. All scores and annotations will adjust appropriately.

- Delete Levels – This option will remove the current zoom level, and all levels that had been generated beneath it (not available at the top-most zoom level).

- Write Scores to File – This option will generate a score file (in the same format as the score input file) corresponding to the region highlighted by the black box.

- Write Sequence to File – Similar to option (c), except it will generate a multi-fasta file corresponding to the alignment columns in that region.

- Write Annotation to File – Similar to (c) and (d), above, except it will generate an annotation file relevant to the highlighted region.

4. If the region of interest is 500 alignment columns or less, the normally black box will turn blue (Fig 2). In this circumstance, the menu brought up by right-clicking in the box will include an option to "View Sequence", either in the same, current window or a new, separate window (functionality remains otherwise identical). When this option is selected, a display containing the alignment columns, along with the appropriate annotations, will appear (Fig 2). Placing the pointer above a column will cause a red border to appear around the column; this border has a red bar counterpart in the zoom level immediately above the sequence, placed at the coordinate that is highlighted in the alignment. Also, the sequence names will be displayed to the left of the alignment, along with a drawing of the tree topology (branch lengths will not be drawn to scale; see File Inputs) that relates the sequences, if it has been supplied. If you scroll to the right and away from the sequence names, simply click anywhere within the sequence view panel to bring up the sequence names in the visible area.

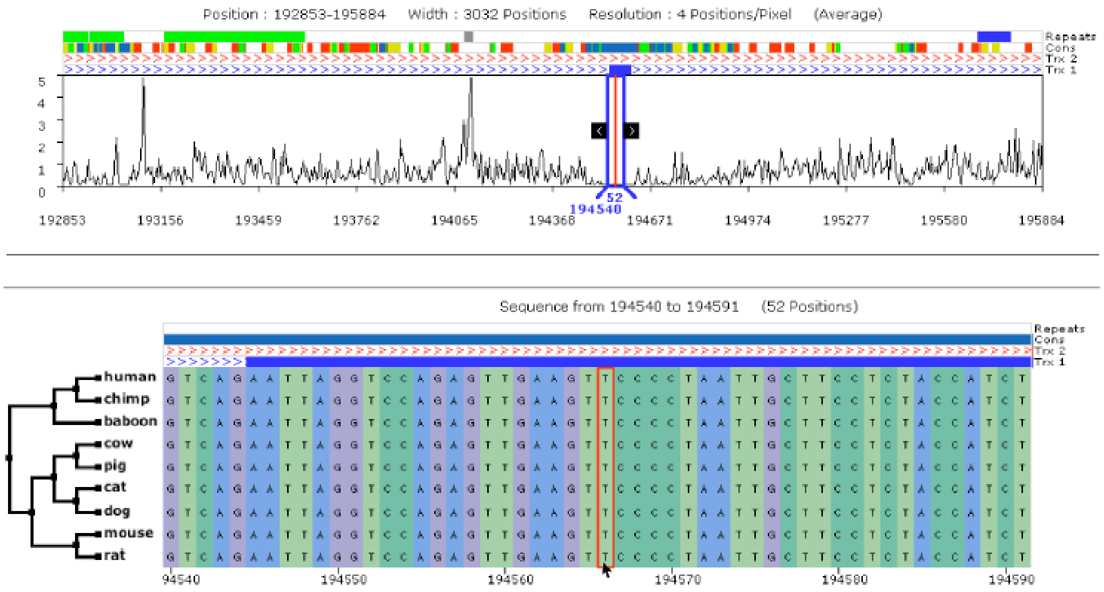

Position : 192853-195884 Width : 3032 Positions Resolution : 4 Positions/Pixel (Average)
Repeats
Cons
Trx 2
Trx 1
5
4
3
2
1
0
52
194540
192853
193156
193459
193762
194065
194368
194671
194974
195277
195580
195884
Sequence from 194540 to 194591 (52 Positions)
Repeats
Cons
Trx 2
Trx 1
human GTCAGAATTAGGTCCAGAGTTGAAGTTCCCCTAATTGCTTCCTCTACCATCT
chimp GTCAGAATTAGGTCCAGAGTTGAAGTTCCCCTAATTGCTTCCTCTACCATCT
baboon GTCAGAATTAGGTCCAGAGTTGAAGTTCCCCTAATTGCTTCCTCTACCATCT
cow GTCAGAATTAGGTCCAGAGTTGAAGTTCCCCTAATTGCTTCCTCTACCATCT
pig GTCAGAATTAGGTCCAGAGTTGAAGTTCCCCTAATTGCTTCCTCTACCATCT
cat GTCAGAATTAGGTCCAGAGTTGAAGTTCCCCTAATTGCTTCCTCTACCATCT
dog GTCAGAATTAGGTCCAGAGTTGAAGTTCCCCTAATTGCTTCCTCTACCATCT
mouse GTCAGAATTAGGTCCAGAGTTGAAGTTCCCCTAATTGCTTCCTCTACCATCT
rat GTCAGAATTAGGTCCAGAGTTGAAGTTCCCCTAATTGCTTCCTCTACCATCT
94540
194550
194560
194570
194580
194590

Figure 2. At higher resolution, a wiggly plot is used to summarize the score data (upper zoom level). When highlighting a window of less than 500 bp, the black box becomes blue, and right-clicking now displays an option to "View Sequence", shown in the lower zoom level. Placing the mouse on an individual column highlights that column and identifies its location in the zoom level above (red bar in upper zoom level).

4. Score Displays

Depending on the resolution of the zoom level, the score data for the region are displayed in either a histogram or wiggly plot format. The resolution of the zoom level is defined as the ratio of the number of alignment positions in the region to the number of pixels used for display. Each zoom level uses the same number of pixels (by default, 800). The region is comprehensively partitioned into non-overlapping windows of a size equivalent to the resolution. A summary value is then tabulated from the scores within each of these windows and used to make the plot.

At low resolution (by default > 50 positions/pixel), the total number of scores at or below a given value (by default, this threshold is 0) is determined for each window. These values are then used to create the histogram (Fig 1). At higher resolution (by default, <= 50 positions/pixel), a wiggly plot (upper portion of Fig 2) is made by plotting the numerical averages of the scores in each window. Thus, at low resolution, a histogram is plotted such that regions of the alignment containing many low scores will have tall peaks; at high resolution, the opposite is true, and regions with many low scores appear as valleys.

As an example, consider for the following score data for a six position alignment:

0, 0, 0.4, 0.3, 0.2, 0

If 3 pixels were available, then the resolution would be 2 positions/pixel, and the region would be partitioned into three windows: positions 1-2, 3-4, and 5-6. If 2 positions/pixel is larger than the 'high resolution' threshold, then the plotted histogram values would be (from left to right, assuming the default score threshold of 0):

2, 0, 1

If, however, 2 positions/pixel is smaller than the 'high resolution' threshold (as it would be using default settings), the wiggly plot values displayed would be (from left to right):

0, 0.35, 0.1

5. Mouse-Over Highlighting

At all zoom levels, placing the mouse pointer over a feature in the annotation tracks will reveal the name of that feature (Fig 3A). In the case of gene/transcript identification, placing the pointer over any spot in the transcript will reveal the name of the gene, and not the identifier of the individual exons.

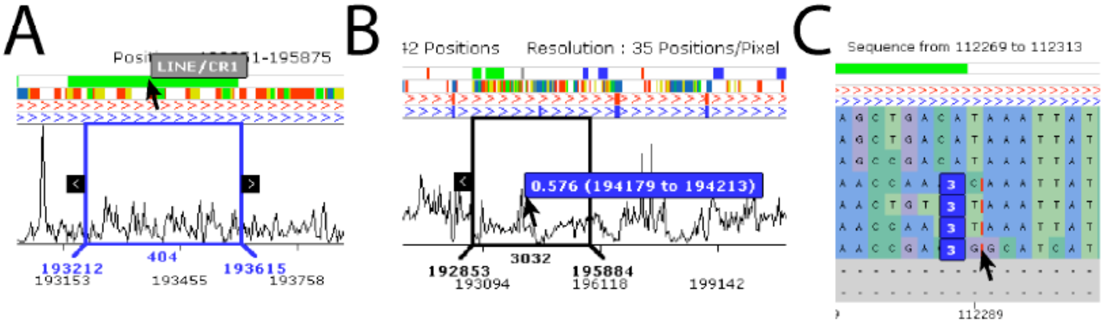

A
Posi
1-195875
LINE/CR1
<
>
404
193212
193615
193153
193455
193758
B
12 Positions
Resolution : 35 Positions/Pixel
<
0.576 (194179 to 194213)
3032
192853
195884
193094
196118
199142
C
Sequence from 112269 to 112313
A G C T G A C A T A A A T T A T
A G C T G A C A T A A A T T A T
A G C C G A C A T A A A T T A T
A A C C A A 3 C A A A T T A T
A A C T G T 3 T A A A T T A T
A A C C A A 3 T A A A T T A T
A A C C G A 3 G G C A T C A T
112289

Figure 3. A. Mousing-over annotation features causes the feature name to be displayed, in this case a repetitive element "LINE/CR1". B. Score data can be obtained using the pointer; in this case, the plotted score is 0.576, determined by averaging the scores from position 194179 to 194213. C. Red hash marks between positions 112289 and 112290 mark coordinates from which a gap in the lead sequence was deleted; moving the pointer over the hash mark reveals that, in this case, 3 bases were removed from each of these four sequences.

Detailed quantitative information contained within the plot can also be obtained. Simply go to the Options pull down menu in the upper left of the browser, and select "Show Value on Mouse Move" (this can be turned off in the same place; also note that turning this feature on will eliminate the red alignment column border mentioned in the Zoom Levels section). When the mouse rests on an individual point within the main display of the zoom level, the plotted value and the coordinates of the alignment scores used to generate that value will be shown (Fig 3B). This feature functions on both the histogram and wiggly plot zoom levels. If you have supplied an *abcgap* file (see File Inputs), when viewing sequence and pointing the mouse at each spot marked with a small red tick mark, the number of bases deleted from that sequence will be displayed (Fig 3C).

6. Input Files

Within the working directory, there are three required input files, and two optional files (working samples of all files are included with the code):

Settings.abcset
YourFile.abcseq
YourFile.abcrat
YourFile.abcann (optional)
YourFile.abcgap (optional)

Note, all these files should be in plain text, with no special characters or formatting.

Settings.abcset:

The *Settings.abcset* file contains all the necessary parameters to launch an ABC browser session. Most are self-explanatory, and include the height (in pixels) of the zoom levels and annotation tracks, the resolution at which the browser switches from 'low to 'high', and the threshold score value used for calculating histogram values (see Score Displays).

The File Path must be specified, with the name of only the file prefix being required. It is assumed that the prefixes are the same for all files, and that all files end in the appropriate suffixes. In the example above, the value specified in the File Path would be '*YourFile'*. A title for the browser session may also be specified; this title will appear in a fixed location at the top of the main ABC window.

Environment colors may also be specified in the Settings file. All colors should be specified in tab-delimited RGB coordinates, where each value is an integer between 0 and 255, and the string (where '\t' represents the tab character) "255\t255\t255" is white, "0\t0\t0" is black, "255\t0\t" is red, etc… You may specify the background, histogram, and text colors, in addition to the colors used for viewing the sequence (one color each for 'A', 'C', 'G', 'T', 'N', 'a', 'c', g', 't', 'n', and '-'), tree, and sequence names.

Finally, the names of the first four annotation tracks may be supplied here (additional tracks can be named in the *abcann* file, see below).

YourFile.abcseq:

The *abcseq* file stores the alignment. With the exception of the first two lines, it should be in a fasta format. The number of characters on each line of sequence does not need to be a particular number (such as 60 or 80), but it MUST be the same for all lines of sequence in the file (except for the last line of a given sequence, which can be any number smaller than the other lines). Each sequence name should begin on a new line with the '>' character, with the sequence belonging to that name residing on all subsequent lines until the next line beginning with '>'. Also, since it should be an alignment, all sequences should be the same length; while this is not enforced by the

ABC, sequences with differing lengths will likely result in an exception being thrown when trying to view the sequence. The following example is in a valid format:

```
>Human
ACGTNAT
ATT
>Mouse
ACGT-AT
ATT
>Rat
ACGT-AT
ATT
```

However, the following example is NOT valid (sequence lines are not the same length):

```
>Human
ACGTN
AT
ATT
>Mouse
ACGT-ATATT
>Rat
ACGT-AT
ATT
```

The top two lines of the *abcseq* should begin with the '#' symbol. The first line should contain the first and last coordinates of the alignment, separated by a space. The second line should contain a standard parenthesis tree describing the phylogenetic relationship of the sequences. Branch lengths are optional. The tree may be unrooted, but only one trifurcation is permitted. If your tree is invalid, or you do not supply one, no tree will be displayed. If you do not supply a tree, the second line of the file must still begin with '#'.

YourFile.abcrat:

The *abcrat* file contains the scores associated with each alignment column. The first line of the file should be identical to the first line of the sequence file, beginning with '#' and consisting of the first and last coordinates of the alignment separated by a space. The second line should begin with '#', and consist of the exact number of scores contained in the file. This number should correspond exactly to the length of the alignment (last coordinate – first coordinate + 1).

The scores should begin on the third line, one score per line. Each line should contain two numbers, tab-delimited. The first field is the score itself, and the second field should be an integer between 0 and 100. The second integer is required, and is used to determine the color of the wiggly plot at that point. A value of 0 corresponds to black, and a value of 100 corresponds to yellow. Intermediate values will produce intermediate colors, with small numbers more similar to black and large numbers more similar to yellow. If you want a black line, simply fill this column in with 0s. This feature is included to allow another dimension of data to be displayed, as long as the data can be normalized to a percentage scale. For example, the percentage of gap characters in a column, GC content, polymorphism content, etc, could be used to color the plot.

One general recommendation is to supply score values that are within a sensible range. The ABC was designed to display data in terms of evolutionary rates over an alignment, and these values are all positive numbers, generally less than 10. If you have negative numbers, or a skew to large values, it may be necessary to normalize them in some way.

YourFile.abcann:

The *abcann* file contains annotations for features of the sequence data. It is not necessary to supply this file; however, if it is not given, four empty annotation tracks will be displayed. Two header rows are required at the top of the file, each beginning with the ‘#’ symbol. The first line should be the same as the first line of the sequence and score files. The second line should consist of titles for any ‘extra tracks’, and these should be tab-delimited. By default, four tracks are displayed; you may, however, leave any or all of these tracks empty. Note also that annotations should be 1-based, i.e. the first alignment column should be 1 (not 0).

The format of the annotation for the first two tracks is designed to allow the annotation of transcripts, with exons, introns, and orientation. The format is as follows (‘\t’ and ‘\n’ are tab and newline characters, respectively):

For a gene on the first track, positive strand:

```
>GENE\tStart\tStop\tR\tG\tB\n
Start\tStop\tEXON\n
Start\tStop\tEXON\n
```

….

Note that “GENE” should be replaced with the name of the gene, ‘R’, ‘G’, and ‘B’ should be replaced with red, green, and blue color values, and EXON should be a word that describes the exon (may be ‘utr’, ‘exon’,’cds’, etc).

For a gene on the first track, negative strand, use the above format, except replace ‘>’ with ‘<’. For example:

```
<GENE\tStart\tStop\tR\tG\tB\n
Start\tStop\tEXON\n
Start\tStop\tEXON\n
```

….

To annotate genes on the second track, use the above format, replacing ‘>’ and ‘<’ with ‘+’ and ‘-‘ to designate positive and negative strands, respectively.

For all remaining annotations, each line is a separate feature, and each feature consists of a name, start and stop coordinates, and an RGB color string, all tab-delimited. To place features on the third track, begin the line with ‘t3_’, followed immediately by the feature name; for the fourth track, begin the line with ‘t4_’; for the fifth and subsequent tracks (these will only appear in the browser if requested), begin the lines with ‘t5_’, ‘t6_’, etc. Note that if any annotations appear using the ‘t5_’, ‘t6_’, etc., notation, a title for this track MUST be supplied at the top of the file.

For example, consider the following set of annotations:

```
t3_ConsElement\t50\t100\t255\t0\t0\n
t4_Sine/Alu\t125\t300\t0\t255\t0\n
t5_Promoter\t400\t500\t0\t0\t255\n
```

This would place a feature called ‘ConsElement’ on the third annotation track from positions 50 to 100, and it would be colored red; a feature called ‘Sine/Alu’ on the fourth track from positions 125 to 300, colored green; and a feature called ‘Promoter’ on the fifth track, from positions 400 to 500, colored blue.

Note that annotations are painted in the order they are placed in the file. Therefore, if two features on the same track have a partial overlap, the feature that comes second in the file will be visible and will partially hide the first feature. If two features overlap completely, only the second will be visible.

YourFile.abcgap:

The *abcgap* file is an optional file that contains data about gaps that may have been eliminated from an alignment relative to a ‘lead sequence’. This often takes place during sliding-window analyses designed to find evolutionarily constrained regions in a specific sequence. Eliminating gaps in this manner establishes the coordinates of the lead sequence as the coordinates of the alignment, making annotation much more convenient. If you’re not sure what this is about, or don’t care, this section can be safely ignored.

The data in this file should consist of tab-delimited values, with one column for each of the sequences in the alignment. The first line should contain the names of the sequences. The first column name should be the lead sequence, and the rest may come in any order. Data in the first column should contain the coordinates relative to the compressed alignment from which a gap was eliminated. In this manner, 0 indicates a gap that was deleted before the first base of the lead sequence, and 1 indicates a gap eliminated just after the first base. Each subsequent column should provide the number of bases that were deleted within that gap from the sequence whose name heads that column. If no bases were eliminated from that sequence (i.e. they were gapped as well) then a 0 should be supplied. Any species containing bases that were deleted at this location will be marked in the ‘Sequence View’ zoom level (Fig 3C). If mouse-over is enabled, the number of deleted bases will also be visible (see Mouse-Over Highlighting).

For example, consider compressing the alignment below (on the left) so that the human sequence is ungapped (below and on the right).

```
>human                        >human
ACGT—-ATGGA-TT                ACGTATGGATT
>mouse                        >mouse
ACGT--AT-GAATT                ACGTAT-GATT
>rat                          >rat
-CGTCCATGGA-TT                -CGTAGGGATT
```

2 bases were deleted from the rat after human position 4, and 1 base was deleted from the mouse after human position 9, so the *abcgap* file would be:

```
human\tmouse\trat\n
4\t0\t2
9\t1\t0
```

Please note that while the *abcgap* file option was designed to show locations where gaps were eliminated, this option could be used to mark sequences at a given alignment coordinate for any desired reason. Ultimately, what is displayed in the browser is a small red tick mark at each position for each species (not including the ‘reference’ sequence) containing a non-zero entry for that position. The numbers do not necessarily need any meaning beyond presence/absence (1/0, for example).

7. Miscellaneous Options

The following are available using the pull-down menus at the top left of the ABC.

- Settings – a dialog box allowing changes in titles and resolution/histogram thresholds, similar to the settings specifed in the *abcset* file (see File Inputs)

- Environment Colors – a dialog box allowing changes in background and other colors, similar to that specified in the *abcset* file (see File Inputs)

- GoTo – a dialog box that allows the user to quickly jump to a specific region of the alignment; midpoint and width or start and stop of the desired window can be specified. Note that this will delete any zoom levels below the top-most level

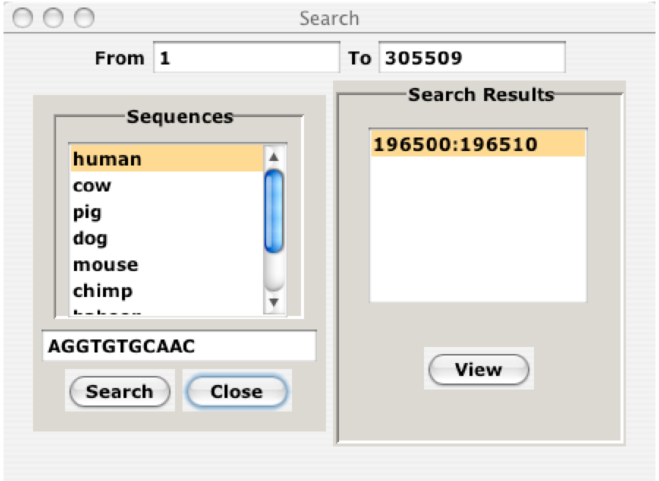
- Search – a dialog box (Fig 4) that allows the user to search for a particular string (using a literal search) within the alignment sequences. The search space may be restricted to specified coordinates. If more than 100 matches are found, only the first 100 will be displayed. Use of regular expressions is not implemented.

- View Tree – this option will open a new window and display the tree topology specified in the *abcseq* file (see File Inputs). Branch lengths will be drawn if supplied (this option is not available if the tree is invalid or not supplied).

- Export as JPEG – Export the currently visible zoom levels as a JPEG file

- Export sequence as JPEG – Export the currently visible sequence as a JPEG

- ShowHide Zoom – Will remove the black zoom rectangle



From 1 To 305509

AGGTGTGCAAC

Search Results: 196500:196510

Figure 4. The Search dialog box allows the user to search for strings within each of the sequences, and also allows the user to restrict the search range by supplying target coordinates.

8. Troubleshooting

Typically, the most frequent and annoying problems derive from simple textual formatting mistakes in the data files (see Input Files). Your best bet is to copy the sample files included with the code, replacing the appropriate lines with your data. Be sure they are all plain text with no special characters. Do NOT use programs like Microsoft Word or Excel, Mac TextEdit, WordPerfect, etc. that introduce lots of special characters. If you run OS X, be sure to eliminate 'meta-characters' which have a tendency to appear often (they appear as ^M in editors like *pico* or *vi*). Save files with Unix line feeds. Delete empty lines, including at the very end of the file. Make sure your parenthesis tree is well-formatted; you can check it by using tree-drawing software from the web. If you are using more than four annotation tracks, be sure the names of the additional tracks are specified at the top of the *abcann* file (see File Inputs).

The ABC is used often and is quite stable for most purposes; however, there are certainly bugs. While we will try to fix them, we make no promises of any kind. Additionally, we are currently working on improving several aspects of the ABC's functionality. Contact information is supplied at the top of this document if you would like to send us bug notices.