

How to use blastcl3 and BlastReport2

Blastcl3 is blast client software from NCBI. This program runs on your local machine and uses the BLAST server at NCBI to perform blast searches of the NCBI sequence databases. Blastcl3 uses a text file containing one or more query sequences in FastA format. The output is a text or HTML file of the blast results. Available from NCBI for MacOS, Windows and several Unix machines for free.

BlastReport2 is a perl script that reads the output of Blastcl3, reformats it for ease of use and eliminates useless information. BlastReport2 has been designed specifically for nucleotide searches of htgs and nr databases using microsatellite markers, cDNA sequences or other nucleotide sequences to build genomic maps of small chromosomal regions. Perl is available on Unix, Windows and MacOS and is free.

Installation and Setup

1. Install blastcl3. FTP from NCBI site: *NCBI web*
Internet access is required to run blastcl3. NCBI gives instructions on how to set up blastcl3 and provides information about using blastcl3 when there is a firewall present. I attempted to use blastcl3 using dial-up access to the internet and was not successful.
2. Check availability of perl. Install if needed: *perl web*
On Unix use 'where perl' to see if perl is available or ask your system administrator.
3. FTP BlastReport2 from this website: bmbmac.biochemistry.iupui.edu/blastrep
4. Create a directory or folder (or use an existing directory) for output from blastcl3 and BlastReport. (one directory for both)
5. Customize BlastReport2 to use this directory - See '**BlastReport2 Customization**' below.

Execution of blastcl3 and BlastReport2

1. Create a text file containing the query sequences to be used. These must be in FastA format:
 1-line description starting with '>' For best use with BlastReport2 use a 2-10 character identifier, e.g. D4S1234 or SLC10A2, immediately following the '>' then followed by a space. BlastReport2 will use this identifier when reformatting the output. Lines 2+ contains nucleotide sequence 'acgt'. Imbedded blanks and new lines are ignored.

```
>Name Description
actgatgagctagatggcaa ... atcc
```

2. Run blastcl3 using this text file as input. For Unix and Windows, blastcl3 is started from the command line. On MacOS, blastcl3 operates from a window. The following parameter must be specified either on the command line or by filling in the appropriate box on MacOS.

e.g.

```
blastcl3 -p blastn -d htgs -e 1e-30 -F F -v 10 -b 10 -i "chr17mrk" -o "blast 17 mrk"
```

On Unix or Windows, press enter or return to start. For MacOS, click the OK button to start. All of these are the same as used for the web-based blast except for specifying input and output files. See '**BlastReport2 naming conventions**' for a discussion of file name to be used for output.

parameter	suggested values	command line
program	blastn	-p
database	htgs or nr	-d
expectation	1e-20 to 1e-35	-e
filter	T or F (true or false)	-F
descriptions	10	-v
alignments	10	-b
input	filename for FASTA input	-i
output	filename	-o

3. When blastcl3 finishes, run perl and execute BlastReport2 to reformat the output. See '**Execution of BlastReport2**' below.
4. Print reports or use a text or word processor to view the reports. Review the alignments and determine which of them are good. See '**Description of BlastReport2 Output**'
5. For each database sequence that contains all or part of one of your search sequences, retrieve the sequence. For finished sequences, take the ends of the sequence, for unfinished, select 200-400 nt pieces from the sequence. Save these new sequences in another text file in FastA format to find other clones or sequences that overlap these sequences. Repeat steps 2-5, to build a local map or a genomic region.

BlastReport2 Customization

Before initial use, BlastReport MUST be customized by changing the values for variables that define the disk and folder in which files are located. There are 2 optional groups of variables that can be customized, Folder/filename variables and Print control variables.

BlastReport can be opened in any word processing program for modification, but **MUST be saved as a TEXT file.**

Folder/filename customization is REQUIRED:

\$MyDisk and \$Folder define the location (folder or subdirectory) of the input and output files and must be customized by replacing the placeholder names (e.g. DISKNAME) with the names on your computer system. Look for this banner on the first page of BlastReport to find the variables that must be modified:

```
# -----
#  C H A N G E   F I L E   A N D   F O L D E R   N A M E S   H E R E   :
# -----
```

\$MyDisk = "DISKNAME"; replace the placeholder DISKNAME with the EXACT name of your disk, including the character required between folders and directories in your operating system (see below). **Leave the quotation marks and semicolon.**

\$Folder = "FOLDERNAME"; replace the placeholder FOLDERNAME with the EXACT name of the folder/subdirectory that contains the BLASTCL3 output files, including the character required between folders and directories in your operating system (see below). **REMEMBER that the full name is required;** if the folder/subdirectory is

within another folder/subdirectory, the entire pathname must be used (e.g. "blastfolder:blastresults:"). This will also be the location of the reports generated by this program. **Leave the quotation marks and semicolon.**

For Unix, the diskname can be replaced by a null string "" or a high level directory such as "/home/userid/".

Note: Various operating systems use different characters between folder names or between a directory and its subdirectories in a full file name. Make sure to include this character in the \$Folder literal (characters between the quotes, "). The \$Folder and \$MyDisk literals must end with this character.

For MacOS, the folders are connected by a colon ':' DiskName:MyFolder:SubFolder:MyFile

For Windows, folders are connected by a backslash '\' c:\MyFolder\SubFolder\MyFile

For Unix, directories are connected by a forwardslash '/' /HomeDirectory/SubDirectory/MyFile

Perl uses a backslash '\' as an escape character for special uses such as \n for new line. Use \\ instead of a single backslash in Windows file names. e.g. \$MyDisk = "c:\\"; will work correctly

Other file/folder customization. Optional customization for your convenience.

The remaining file and folder name variables can be used to change the names of the files that are read and written by BlastReport. For input, BlastReport reads the directory as defined above and looks for files that start with the literal defined by \$BlastPre. Then select the dataset that you wish to analyze. This requires the files written by blastcl3 to start with this prefix (\$BlastPre value).

Optional parameters to change report/file names:

\$Link = "_"; value used to link parts of file name. This character is used to link the prefix to the rest of the file for the reports generated by this program. The default '_' will work for MacOS, Windows and Unix environments. You may change this to a blank if your operating environment allows.

\$BlastPre = "blast "; prefix of BLASTCL3 output used as input to program. This is the prefix of Blastcl3 output files to be read by this program. e.g. the default value of 'blast ' will select a file named 'blast 04', etc. Your operating environment may require an underscore: 'blast_' instead of a blank to delimit the prefix used in these file names.

\$ReportPre = "BRept "; literal to prefix all reports written by this program. All of the filenames for the reports generated by this program begin with this string. You may modify it if you desire. Keep the literal short, 6 characters maximum. Report names are generated starting with this literal and adding the characters after 'blast' (\$BlastPre) of the input file and using the \$Link variable above. input: blast 04 GABA leads to files named:

BRept_04_GABA_Full

BRept_04_GABA_Sum

BRept_04_GABA_Tab

\$Full = "Full"; literal to suffix name of full report.

\$Sum = "Sum"; literal to suffix name of summary report.

\$Tab = "Tab"; literal to suffix name of tab delimited report.

Print control customization (optional)

These variables control the amount of data that is printed. Look for the following text to find these variables in BlastReport. These variables follow the ones for file and folder names.

```
# -----  
# C H A N G E   P R I N T   C O N T R O L   P A R A M E T E R S   H E R E :  
# -----
```

The print control variables are limits on the amount of information to be printed in the reports generated by BlastReport. The default values set here are limits that we have found useful for our work. You may want to change these to optimize the reports for your use. These merely put limits on the amount of data to be printed in the reports and are not 'intelligent' settings. Increasing some of the settings to very high values may result in an 'out of memory' error, especially if you are using a large number of query sequences.

There are 4 groups of print variables and 1 single control variable. For the groups, there is a 'master' variable followed by a recommended default value for markers and cDNAs. The first or 'master' variable in each group should not be modified. The master variables are set at execution time as indicated by your response to:

Select type of Input:

1 Marker

2 cDNA

>Enter number for type of file or Q to quit:

Other types of data may be used. Use 'marker' settings for small query sequences or any data that is expected to have 1 or 2 local alignments per database sequence, such as individual exons. cDNA's have alignments for each exon, so more alignment regions will be found, so the cDNA settings print more data.

Below is a listing of all the print control variables in BlastReport1.0 and an explanation of their usage.

\$RegionConst: This limits the number of alignment regions printed per line in the report, 4 will typically fit on a page printed in portrait. You may want to increase this if you are printing in landscape.

\$DBseqPrint: Modify \$cDNADBseq and/or \$Marker DBseq This limits the number of sequences for which alignment regions will be printed. Typically for markers, there will be no more than 4 significant alignments. Any others are usually alignments to the repeat area or to a repetitive sequence in the marker, e.g. ALU. For cDNAs, the setting is usually set to the same number as the limit on the number of alignments in the original Blastcl3 run, 10 is suggested.

\$RegionPrint: Modify \$cDNARegion and/or \$MarkerRegion This limits the number of alignment regions to be printed for each Database sequence that aligns with the query sequence. For markers, the limit of 4 is usually fine. Typically you will only see 1 or 2 significant alignment regions for a microsatellite marker. This prevents alignments to repetitive elements from being printed. For cDNAs the number of alignments is equal to the number of exons that are found in a BAC clone or other genomic segment. The largest number of exons that we've seen in one BAC is 55. Normally 15-20 are seen. The default is set at 55 for this reason. It's not a magic number. Use a limit just to make sure that if something unusual happens, a large amount of data isn't printed. The total number of regions found is printed in the report. To see more of them change the value for \$cDNARegion or \$MarkerRegion and rerun BlastReport. The input file is not deleted by the BLastReport.

\$MaxAlign: Modify \$cDNAAlign and/or \$MarkerAlign For each query sequence, the number of alignments are reported and a summary line is printed for each alignment: region of query that aligned, expectation, number of matching nucleotides, and the name of the database sequence for this alignment. \$MaxAlign limits the number of summary lines to be printed. We've found that 12 is sufficient for markers. This allows you to see significant alignments and a few repetitive alignments to determine where repetitive elements may be in a marker. The default for cDNA sequences is 55, an upper limit that seems to be reasonable.

\$MaxFull: Modify \$cDNA_MaxFull and/or \$Marker_MaxFull This limits the number of actual alignments that will be printed. These are the full alignments that you see in any BLAST output. The limit here is arbitrary, just to keep the report small, 4 or 5 full alignments are usually all that's needed to check the quality of the alignments. The alignments that are printed are the first N alignments (as set by this variable) in the report for this query. To see more alignments, change this parameter and

rerun the program. Alternatively, open the original Blastcl3 output in a word or text processor and search for the alignment(s) you wish to view.

Execution of BlastReport2

Running Perl differs by operating system. Unix and Windows both run from the command line, Perl for Windows runs in a DOS window. On MacOS, there is a GUI interface.

Unix and Windows(in a DOS window) on the command line type: **perl BlastReport2**

MacOS: Start perl by double clicking on the macperl icon. Click **Script** on the menu bar and select **Run Script**. Select BlastReport2 from the folder where it was saved. (Double click or Open).

Once started, BlastReport2 runs similarly on all operating systems.

The execution of BlastReport requires 3 inputs:

1. The chromosome number or whatever number that you have chosen. If you enter 4, it will prefix with a zero to use '04'.
2. The number of the file to be analyzed. A list of choices will be presented.
3. The type of input: marker or cDNA. See discussion below under print customization for an explanation of how 'marker' and 'cDNA' responses modify the reports written by BlastReport.

Sample execution of BlastReport2:

```
==== Begin BlastReport Parameter Selection ====

>Enter chromosome number of blast report to use or Q to quit.
5

blast 05 files:

 1 blast 05 09-05 cDNA
 2 blast 05 AC021079 04-19
 3 blast 05 gabra1 04-19
 4 blast 05 gene nr 04-18
 5 blast 05 genes 04-18
 6 blast 05 ht 03-29

>Enter number of file to use or Q to quit: 1

Using file: blast 05 09-05 cDNA
```

Select type of Input:

- 1 Marker
- 2 cDNA

>Enter number for type of file or Q to quit: 2

Type of file selected: cDNA

==== Begin BlastReport Execution ====

INFO: BLASTIN opened successfully:

IMac03:Folder:blasts:blast 05 09-05 cDNA

INFO: Number of Query Sequences in the run: 18

INFO: Number of Query Sequences with alignments: 16

INFO: Sequences that may have significant alignments: 16

INFO: Full Report: IMac03:Folder:blasts:Brept_05_09-05_cDNA_Full

INFO: Summary: IMac03:Folder:blasts:Brept_05_09-05_cDNA_Sum

INFO: Excel Table: IMac03:Folder:blasts:Brept_05_09-05_cDNA_Tab

==== BlastReport Execution Completed ====

BlastReport2 naming conventions

BlastReport was originally written in a MacOS environment. Macperl is different from other versions (Windows and Unix) because it runs with a GUI interface and not from a command line. Therefore parameters to specify the input file, type of query and chromosome are entered at program prompts, instead of parameters specified on the command line at execution time. This method works in all operating environments. To streamline this data entry, a specific directory is used for all input and output files and a naming convention was adopted for input file names. The directory is specified by customizing the program before the first execution, see folder/filename customization below. All input files have the form:

blast *nn description*

Where blast is a literal that identifies all possible input files, *nn* is a 2-digit number (e.g. chromosome number) and *description* is any additional text desired to identify a file.

e.g. blast 07 mrkers 08-11

The literal 'blast' may be changed by modifying one variable \$BlastPre as discussed below. The requirement of a 2-digit number is not optional. We generally group all of the markers or cDNA's that we use for searching by chromosome. That is, all of the markers for a particular chromosome or portion of a chromosome are saved in a single text file to be BLASTed together. All cDNA's for a chromosome are saved in a second file. These files are used for input to

blastcl3, so a particular output file from blastcl3 will contain alignments for one chromosome and one type of search sequence. This allowed us to optimize BlastReport for markers and cDNAs separately. See the Print customization section below. BlastReport requests a chromosome number, e.g. 07, and then displays all files that start with 'blast 07' in the directory. BlastReport then asks you to select the file to be analyzed from this list. We found this more convenient than trying to type in the fully qualified name of the file. If you don't want to segregate your queries by chromosome, use any 2-digit number.

Files written by BlastReport also have a naming convention. For the example above, 3 files will be written:

BRept_07_mrkrs_08-11_Full

BRept_07_mrkrs_08-11_Sum

BRept_07_mrkrs_08-11_Tab

The prefix 'BRept' may be customized, as well as the suffixes: 'Full', 'Sum' and 'Tab'. The rest of the name (07_mrkrs_08-11_) is from the original input file name so that you can identify the source file easily. To customize these values see '**Other file/folder customization**' above.

Description of BlastReport2 Output

BlastReport reads the output of a blastcl3 query, which contains the search results of one or more query sequences. BlastReport searches the blastcl3 output for alignments and generates 2 reports, Full and Sum, and a tab-delimited file (Tab).

BRept_Full provides a summary for each query sequence followed by the alignments that are most likely to be biologically significant. The information is trimmed and reorganized to make it more compact and easier to use.

BRept_Sum gives a list of the regions of alignment for each query.

BRept_Tab is a tab delimited file that contains the query name, name of aligning sequence, clone name, length, expectation, and the first region of alignment. It can be imported into a spreadsheet and sorted by one or more columns.

To describe the features of BlastReport output, sample output is available on this website. These reports were generated on an iMac, but results on Unix or Windows would be very similar if not identical. Three microsatellite markers were taken at random for human chromosome 17, D17S1868, D17S1298 and D17S1308. A text file containing these 3 sequences in FastA format were used as input to blastcl3, using the following parameters: program (blastn), database(htgs), expectation(1e-30), filtering was turned off (F), descriptions(10), alignments(10), output (blast

17 09-15 mrker). These marker were chosen because they are good examples of what may be seen.

The reports produced by this execution of BlastReport1.0 are:

BRept_17_09-15_mrker_Full

BRept_17_09-15_mrker_Sum

BRept_17_09-15_mrker_Tab

Full Report: BRept_17_09-15_mrker_Full

This the first part of the information reported for marker D17S1308:

(A) ----- D17S1308 -----

(B) Query= D17S1308|gb|G07971 GTAT1A05 **Match AC008087 **hi hit 180-290
(636 letters)

(C)

Sequences producing significant alignments:	Score	E
Value	(bits)	
gb AC008087.3 AC008087 Homo sapiens chromosome 17 clone RP5-1127...	948	0.0
gb AC025743.2 AC025743 Homo sapiens chromosome 17 clone RP11-676...	948	0.0
emb AL121672.10 HSDA109B7 Homo sapiens chromosome 22 clone RP6-1...	624	e-179
gb AC016560.7 AC016560 Homo sapiens chromosome 5 clone CTC-463E1...	186	4e-47
gb AC021988.4 AC021988 Homo sapiens chromosome 19 clone RP11-84C...	186	4e-47
gb AC069503.11 AC069503 Homo sapiens chromosome 12 clone RP11-87...	176	4e-44
gb AC010319.6 AC010319 Homo sapiens chromosome 19 clone CTD-2521...	176	4e-44
gb AC021154.5 AC021154 Homo sapiens chromosome 19 clone RP11-510...	170	3e-42
gb AC011442.4 AC011442 Homo sapiens chromosome 19 clone CTC-2150...	168	1e-41
gb AC021197.5 AC021197 Homo sapiens chromosome 19 clone RP11-798...	167	4e-41

(D) --- D17S1308 alignment analysis ---
Number of alignments: 144

Region(s) of alignment for Query: D17S1308 (first 12 only)

From: 1	To: 636	Expect: 0.0	Match: 602/636	AC008087.3
From: 148	To: 318	Expect: 4e-38	Match: 148/171	AC008087.3
From: 185	To: 316	Expect: 1e-34	Match: 117/132	AC008087.3
From: 180	To: 288	Expect: 1e-34	Match: 100/109	AC008087.3
From: 154	To: 315	Expect: 9e-33	Match: 139/162	AC008087.3

From:	1	To:	636	Expect:	0.0	Match:	602/636	AC025743.2
From:	148	To:	300	Expect:	2e-46	Match:	138/153	AC025743.2
From:	158	To:	288	Expect:	4e-38	Match:	118/131	AC025743.2
From:	148	To:	288	Expect:	6e-37	Match:	125/141	AC025743.2
From:	148	To:	316	Expect:	6e-37	Match:	146/169	AC025743.2
From:	180	To:	301	Expect:	4e-35	Match:	110/122	AC025743.2
From:	180	To:	288	Expect:	1e-34	Match:	100/109	AC025743.2

The first line, (A) is an 'eye catcher' that makes it easy to scan for the beginning of the information for a query. Or you may use a word processor to find '---===!' which is unique to this line. Next is the query description line as listed by BLAST, (B). Immediately following that are the descriptions of database sequences that align to this query, (C). The number of these descriptions is limited by the -v parameter in the blastcl3 execution. The next section (D), is information extracted from all of the alignments for this sequence. It gives the total number of alignments found for this query in this case 144. Regions of alignment for your query sequence are given next, \$MaxAlign displayed is limited by the \$MaxAlign value, in this case the value is 12. (See Print Customization above.) The first alignments regions are given. For each the region of your query in the local alignment, From: /To:, the expectation for this alignment, the quality of the match is given by a ration of identical nucleotides to total number of nucleotides in the local alignment. The first local alignment spans the entire length of the query, 1-636, has an expectation of 0.0 and 602 out of 636 nucleotides match. The database sequence that gives this alignment is AC008087.3. The next 4 local alignments only span part of D17S1308, in the area of 148 to 320, this is an indication that this portion of D17S1308 is a highly repeated sequence, such as an ALU or Line sequence. The fact that there were 144 total local alignments for these 10 database sequences reinforces this interpretation. In fact if no limits are put on the number of descriptions returned, D17S1308 has thousands of alignments with and expectation better than 1e-30. This is one reason to limit the number of descriptions and alignments when running blastcl3, -v and -b respectively. There is one more significant alignment in this group for AC025743.2. Notice the expectation for given in the description line for AL121672.10, this may also be a significant alignment, but no more information is given for it to conserve space. This is an example of when you may want to increase the value for \$MaxAlign, specifically \$MarkerAlign, and rerun the BlastReport. Alternately, you may view the original blastcl3 output to see to alignment. Checking the information in the next section can also help: (some modification to fit page).

>>>>>> ----- <<<<<<<<

D17S1308 may have a significant alignment.

D17S1308	1 -	636	148 -	318	185 -	316	180 -
288							
AC008087.3	35170 -	35784	4707 -	4877	89055 -	88924	48450 -
48558							

```

D17S1308          1 -    636    148 -    300    158 -    288    148 -
288
AC025743.2    58951 - 59566  42245 - 42397  72684 - 72814  34561 -
34701

```

```

D17S1308          1 -    361    347 -    636    158 -    291    148 -
318
AL121672.10 141032 - 141388  141362 - 141645  17300 - 17433  118768 -
118938

```

```

D17S1308          148 -    317    148 -    316    180 -    318    186 -
316
AC016560.7 151389 - 151220  33912 - 34080  23265 - 23402  85425 -
85295

```

This section starts with another eye-catcher. For the first 4 database sequences that have alignments, the first 4 regions of alignment are given, see limits **\$DBseqPrint** and **\$RegionPrint** above. For database sequence AL121672.10, the first 2 regions of alignment span the length of the query sequence. This is a case where the length of tandem repeat segments for the marker sequence and the genomic sequence vary significantly. So 2 local alignments are seen instead of 1. You can see this by examining the nucleotide numbers for the database sequence, AL121672.10 and noticing that the 2 local alignments overlap. Viewing the full alignments in the original blastcl3 will confirm this. (See query D17S1298 for another example of this. In this case the alignment is available to view in BlastReport1.0.) For Database sequence, AC016560, the only alignments are in the region of the repetitive segment. D17S1868 shows the usefulness of the print limits. The original blastcl3 report is 58 pages long, most of it (53 pages) are alignments to this repeated region. In some cases, the expectation limit will eliminate these repetitive alignments from the blastcl3 report. But these alignments are good enough and long enough to be included. The conservative print limits eliminated a useful alignment in the first section, but the regions of alignment in the 2nd section lists the alignment so that you can determine if you need to examine this further. Note that the description for AL121672.10 states that it is a chromosome 22 sequence. We have found several sequences in htgs that have been mislabeled. Don't discard a sequence based on the description. Use the sequence to search htgs and see if it aligns with other sequences labeled chr 17 or chr 22.

The last section of the report for a query contains the first four full alignments for this query sequence. (See **\$MaxFull** above for this limit.)

```

>gb|AC008087.3|AC008087 Homo sapiens chromosome 17 clone RP5-1127L24 map 17,
WORKING DRAFT
      SEQUENCE, 12 unordered pieces
      Length = 97573

```

Score = 948 bits (478), Expect = 0.0
 Identities = 602/636 (94%), Gaps = 21/636 (3%)
 Strand = Plus / Plus

```
Query: 1      acagtctccaaggtttcgttttgttttacttttaagattagagactgaatctgcct 60
             |||
Sbjct:35170  acagtctccaaggtttcgttttgttttacttttaagattagagactgaatctgcct
35229
```

```
Query: 61      canacggtgaaatcaaggtagaaaaatgtttctgtgaaactttgtcatcactatacc 120
             ||
Sbjct:35230  cagacggtgaaatcaaggtagaaaaatgtttctgtgaaactttgtcatcactatacc
35289
```

(truncated to save space - see BRept_17_09-15_mrk_Full for complete details.)

Summary Report: BRept_17_09-15_mrker_Sum

The summary report lists only the regions of alignment as seen above. Below is an example from a different search. This search was done using the FastA sequence for a cDNA, NM_001045. The blastcl3 output was analyzed by BlastReport1.0 using the cDNA settings. When cDNA is selected, the regions of alignment are printed in query sequence order instead of ordering them by score or expectation. This ordering helps you to see the exons and to see if any are missing a local alignment. NM_0001045 (SLC6A4) is 2508 nucleotides long. You can see that there are local alignments spanning the entire sequence.

```
Query= SLC6A4 |ref|NM_001045.1|| Homo sapiens solute carrier family 6

SLC6A4      1 - 416      415 - 551      549 - 771      771 -
912
AC024065.5  81587 - 81172  78490 - 78354  77897 - 77675  76864 -
76723

SLC6A4      908 - 1046     1044 - 1148     1149 - 1278     1274 -
1389
AC024065.5  75790 - 75652     75287 - 75183     58354 - 58483     59794 -
59909

SLC6A4      1389 - 1522     1522 - 1622     1621 - 1722     1722 -
1890
AC024065.5  60575 - 60708     61981 - 62081     63391 - 63492     67887 -
68055

SLC6A4      1889 - 2508
AC024065.5  72694 - 73313
```

Tab Report : BRept_17_09-15_mrker_Sum

Locus	Sequence	Clone	Length	Expect	Lalign 1	Lalign 2	Salign1	Salign2
D17S1868	AC069454.2	RP11-708H21	181963		0	1	345	163341 163682
D17S1298	AC002993.1	HCIT169H9	140210	e-172		1	315	74315 74630
D17S1298	AC011189.5	RP11-231G16	180493	e-131		1	234	37512 37745
D17S1308	AC008087.3	RP5-1127L24	97573		0	1	636	35170 35784
D17S1308	AC025743.2	RP11-676J12	193132		0	1	636	58951 59566
D17S1308	AL121672.10	RP6-109B7	146522	e-179		1	361	141032 141388
D17S1308	AC016560.7	CTC-463E16	192430	4.00E-47	148		317	151389 151220

The tab file contains the query sequence, the database sequence that it aligns to, the name of the clone if given, the length of the database sequence, the expectation of the best alignment, and last the alignment regions for the best alignment. Lalign1 and Lalign2, give the begin and end sequence numbers for the query sequence (or locus). Salign1 and Salign2 give the begin and end numbers for the alignment for the database sequence.