

RDPquery

By: Glen Dyszynski

Introduction:

RDPquery is a Java application for retrieving taxonomic identifications for 16S rRNA prokaryotic gene sequences. The program utilizes The Ribosomal Database Project's³ (<http://rdp.cme.msu.edu>) online sequence match tool to retrieve classification information. RDPquery was created by Glen Dyszynski and Wade Sheldon in the Departments of Microbiology and Marine Sciences at the University of Georgia. The program makes use of another Java application created by Ahmed Moustafa called JAligner¹, which creates alignments and performs comparisons on sequence data. Please read all of the requirements before using the program.

The general strategy used is as follows (Figure 1). For each query sequence, RDPquery asks the RDP to find the 10 entries (or some specified number of entries from 1-20) with the highest Sab values. However, the sequence with the highest Sab value is frequently not the sequence with the highest similarity, in the same way that the sequence with the highest BLAST score frequently does not have the highest similarity. Therefore, RDPquery uses JAligner to calculate the sequence similarity for each of the sequences with high Sab values. To limit the number of requests on the RDP, this action is done locally using downloaded copies of all the RDP sequences. Therefore, it is necessary to download the most recent version of the RDP sequences so that all of the matching sequences returned by RDP can be found in the RDP database FastA file.

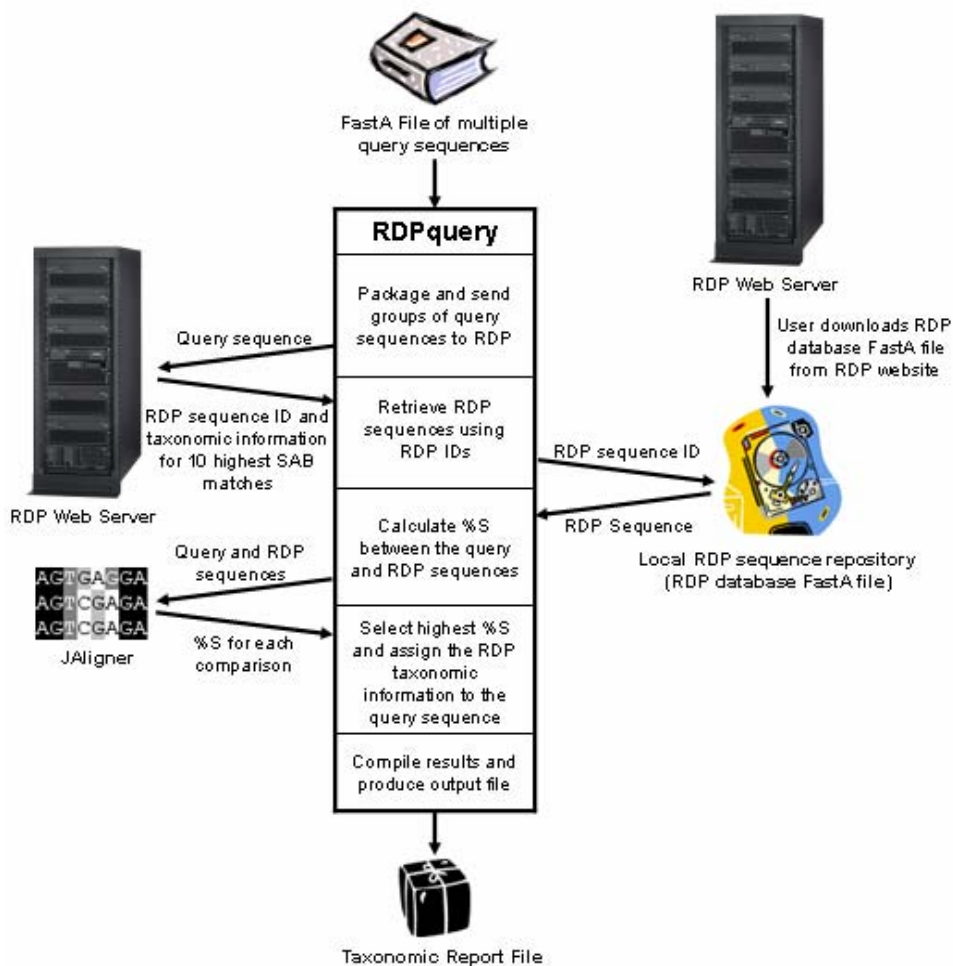


Fig 1. Overview of RDPquery

RDPquery then identifies the sequence with the highest similarity and creates an output file with two sets of taxonomic identifications. The first set contains all the taxonomic data provided by RDP for the sequence with the highest similarity. The second set, however, contains only those taxonomic identifications where the similarity value exceeds a predetermined cutoff. These cutoffs were generated by a survey of the taxonomy in Bergey's Manual of Systematic Bacteriology² (Figure 2). The default cutoff values were set to represent the similarity value at which one would be 95% confident in declaring a given taxonomic assignment. For instance, 95 % of the comparisons we surveyed between members of different genera from within the same family possessed less than 95 % sequence similarity. Similarly, 95 % of the comparisons between members of different families from within the same order possessed less than 92 % sequence similarity. Thus, a clone possessing 94 % sequence similarity to a type strain would be classified in the same family but not in the same genus. Thus, the guidelines are conservative and tend to assign clones to taxonomic groups when there is a high level of confidence.

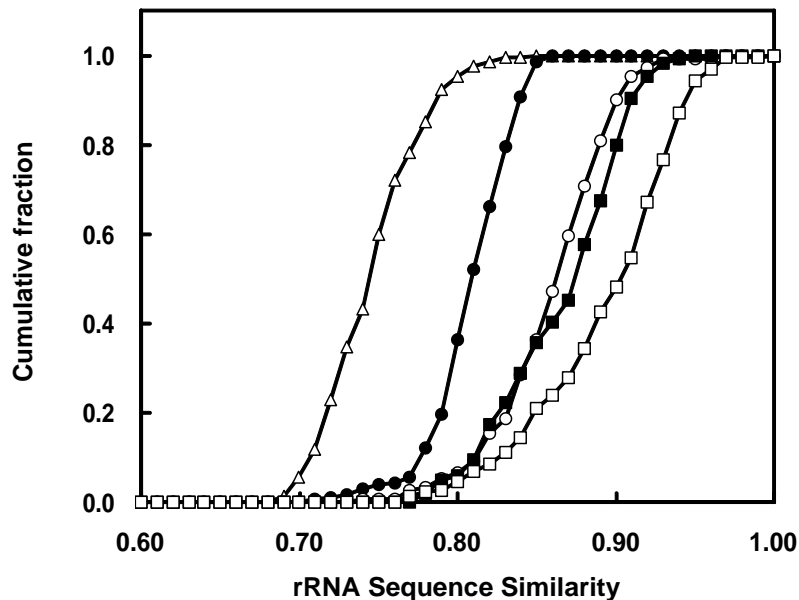


Figure 2. Survey of taxonomic assignments in Bergey's Manual of Systematic Bacteriology. At each level, the rRNA sequence similarity was determined for representatives of different taxa from within the same higher taxonomic group. Thus, at the genus level, representatives of genera within the same family were compared. At the family level, representatives within the same order were compared. All sequences used were from type strains and >1300 bp. No more than six sequences were selected from any one taxon. In total, genera comparisons (◻, 223 comparisons) included representatives of 18 families, family comparisons (■, 104 comparisons) included representatives of 5 orders, order comparisons (◊, 151 comparisons) included representatives of 7 classes, class comparisons (●, 335 comparisons) included representatives of 2 phyla, and phyla comparisons (Δ, 210 comparisons) included representatives of the bacterial domain.

Requirements:

- 1) All input files must be stored in the same directory as RDPquery_2_7.jar
- 2) Query sequences must be in a single FastA formatted file.

Note: See example sequence file provided with the program, queryseqs.fa, or http://en.wikipedia.org/wiki/Fasta_format for a description of this format.

- 3) Query sequence headers must be unique prior to the first space.

Note: If your naming conventions cannot conform with this requirement, you will need to run separate batches so that there are not naming conflicts. RDP uses the text between the ">" character and the first space to uniquely identify each sequence. If two sequence names are not unique, RDP will only return data for the first instance of the name, causing the program to run incorrectly.

- 4) Query sequence headers must contain one and only one ">" character, and it should be the first character of the header line.
- 5) The system must have JRE version 1.5 or later installed (<http://java.sun.com>).

Before you begin:

Retrieving the RDP database FastA file:

- 1) Go to <http://rdp.cme.msu.edu>
- 2) Click on the "Resources" link at the top of the page.
- 3) In the table of files available on the page, download the Unaligned, FastA file.
- 4) Once downloaded, you will need to extract the file. WinRAR (available at <http://www.rarlab.com/download.htm>) and bzip2 (available at <http://sources.redhat.com/bzip2/>) are two applications that can extract the file. Once extracted, the 7MB file will become about 120MB.

Ensure that you are running JRE (Java Runtime Environment) 5.0 or later:

In Windows, this can be done by going to Start -> Settings -> Control Panel -> Add or Remove Programs. Check that it says "J2SE Runtime Environment 5.0 Update 1" or later somewhere in the list.

Note: If you do not have a JRE installed or your version is out of date, download and install J2SE Runtime Environment (JRE) 5.0 or later from "<http://java.sun.com>". It may be necessary to restart one's computer after installation.

Executing RDPquery:

- 1) Open the command prompt and go to the directory that contains RDPquery_2_7.jar.

Note: To open the command prompt in Windows, select Start -> Run, then type "cmd".

- 2) Execute RDPquery_2_7.jar with the following command:

```
java -jar RDPquery_2_7.jar
```

Note: Alternatively, you can execute the program using:

```
java -Xmx200m -jar RDPquery_2_7.jar
```

This command raises the maximum amount of memory Java is allowed to use and is necessary if one wants to run many hundreds or thousands of sequences. See question 1 in the FAQ for more information.

3) You should be prompted with a screen that looks like this:

RDP Options:

(O) Source: Both
(T) Strain: Both
(Z) Size: >=1200
(M) Matches from RDP: 10

(F) File names
(C) Cutoff values
(I) Index new RDP database FastA file
(E) Exclude GenBank accession numbers: false
(G) Group size: 50
(R) Retries to contact RDP: 50
(P) Gap Penalties: 12.0(open) 4.0(extend)
(H) Highest hit sequences to display: 1
(B) Begin

(X) Exit

Select menu option:

RDPquery Options:

To select an option, enter the letter in parenthesis to the left of the option and press enter.

(O) Source:

Options:

(U) Uncultured, (I) Isolates, (B) Both

Compare query sequences against RDP sequences that are uncultured, isolates, or both. To obtain taxonomic assignment, isolates should be used.

(T) Strain:

Options:

(T) Type, (N) Non-type, (B) Both

Compare query sequences against RDP sequences that are type strains, non-type strains, or both. To obtain taxonomic assignments, the type strains should be used.

(Z) Size:

Options:

(G) >=1200, (L) <1200, (B) Both

Compare query sequences against RDP sequences that are (1) greater than or equal to 1200 base pairs, (2) less than 1200 base pairs, or (3) both.

(M) Matches from RDP:

Options:

Number of matches to request(1-20)

Defines the number of matching sequences that RDP will return for each query sequence. Simulations have found that the top ten matches will find the most similar sequence 98 % of the time.

(R) Retries to contact RDP:

Options:

Number of retries to contact RDP

Defines the number of times RDPquery will attempt to retrieve data from RDP. This should generally be set at approximately half of the total number of query sequences.

(F) File names (view/change):

This menu allows the user to change the input/output file names that RDPquery will use. The user will not be able to choose the directory where his/her files are located. The directory containing RDPquery_2_7.jar is used as the base directory and is where one's files must be located.

Options:

1) Query seq(s) FastA file: queryseqs.fa

The path to the FastA file containing the sequences the user wishes to assign.
Note: I recommend using files with 500 query sequences or fewer for two reasons: running more than 500 query sequences at a time requires a very large amount of RAM, and the more sequences one runs at a time, the higher the chance that the run will fail due to the problem discussed below in FAQ #5.

2) RDP database FastA file: RDPdatabaseFile.fasta

The path to the RDP database FastA file.

3) RDP database Index file: RDPindexFile.txt

The path to the RDP database FastA file index file.

4) Output file: output.xls

The path to the RDPquery output file. Remember that if a file with the same name already exists in the directory, it will be overwritten. Also, the actual output file name will have the chosen options inserted into the file name.

5) Go Back

Return to previous menu.

(C) Cutoff values (view/change):

Options:

Current cutoff values:

domain: 0.5
phylum: 0.8
class: 0.85
order: 0.91
family: 0.92
genus: 0.95
species: 1.0

Change cutoff values? (Y/N)

Displays the current cutoff values and allows the user to input new values. The default values were chosen from our survey of usage in Bergeys (see Figure 2). The value of 0.5 was selected as the default for domain to screen for sequences that are not 16S rRNA. In our experience, sequences below this value usually contain vector or some other gene. If these values are changed, they will only be maintained during the current run and will be reset to the default values when the program is exited. The values entered must be decimal numbers (e.g. 0.95).

(I) Index new RDP database FastA file:

Options:

Are you sure you want to index a new RDP database FastA file? (Y/N)

Allows the user to create an index file from an RDP database FastA file. The file name of the RDP database FastA file used will be that of option 2 in the File paths menu and the file name of the index file that is created will be that of option 3 in the File paths menu.

This function will only need to be executed once for a given release by RDP. For instance, the current RDP version is 26. If RDP were to release version 27, one would need to download the new RDP database FastA file and execute this function to create the new index file.

(E) Exclude GenBank accession numbers:

Options:

Y) Yes, exclude GenBank accession numbers

(R) Exclude Range or (C) Custom regular expression

R) Input range(s) of GenBank accession numbers to exclude

C) Input custom regular expression

N) No, do not exclude GenBank accession numbers

Allows the user to exclude one or more ranges of GenBank accession numbers from those sequences which are considered in the assignment process. This option is helpful for those who have submitted sequences to GenBank and do not want those sequences to match the query sequence(s). If a range of accession numbers spans more than one letter (e.g. AY7502-AZ1503), the user will need to enter two or more ranges separated by commas (e.g. AY7502-AY9999, AZ0000-AZ1503). The user also has the

option to input a custom regular expression. A description of how to create regular expressions can be found at:
<http://java.sun.com/docs/books/tutorial/extra/regex/>

(G) Group size:

Options:

Number of sequences to submit at a time (1-500)

Allows the user to define the number of sequences that are sent at a time to RDP for assignment. Numbers must be within the range 1-500 and can be larger than the number of total query sequences (e.g. if the query sequence file has 121 sequences and a group size of 200 is specified, RDPquery will simply submit all 121 sequences at once).

(P) Gap Penalties:

Options:

Input open gap penalty (e.g. 12.0)

Input extend gap penalty (e.g. 4.0)

Allows the user to define the open and extend gap penalties for sequence alignments by JAligner.

(H) Highest hit sequences to display:

Options:

Number of best RDP match sequences to display per query sequence (0 to Cancel)

Defines the number of RDP match sequences out of the total number of matches sequences that will be displayed in the results file for each query sequence. If, however, one of the hit sequences to be displayed matches an excluded GenBank accession number, this sequence will be removed from consideration, and the next non-excluded hit sequence will take its place. Because of this, it is possible that fewer than the requested number of hit sequences will be displayed in the results file. For instance, if the user requests 20 RDP match sequences and tells RDPquery to display all 10 of them and 12 of the 20 RDP match sequences are excluded, only the 8 remaining hit sequences will be displayed in the results file – rather than the requested 10.

(B) Begin

Starts the program.

(X) Exit

Exits the program. All options defined will be reset to default values.

Using RDPquery:

If this is the first time you have run RDPquery or RDP has released a new database version, go to step 1 – otherwise, skip to step 2.

1) Indexing the RDP database FastA file:

If you are running RDPquery for the first time or RDP has released a new database version, a new index file must be created. The index file helps RDPquery efficiently sort through the enormous number of sequences in the RDP database FastA file.

To create the index file, first define the name of the RDP database FastA file to be indexed and the RDP database Index file to be created. To do so, select the File names menu option and change the “RDP database FastA file” to the file name of the RDP database FastA file and the “RDP database Index file” to the name of the index file you wish to create. File names are changed by selecting the corresponding number in the menu, pressing enter, then entering the new file name. The RDP database FastA file *must* be located in the same directory as the RDPquery_2_7.jar file. The index file will be created in that directory as well.

Indexing an RDP database FastA file only needs to be done once for each new release of the RDP database and does *not* need to be done every time the program is executed. If one is running RDPquery for the first time on a computer or RDP releases a new revision of its database, index the new RDP database FastA file.

Note: For more information on indexing, see FAQ question #2.

2) Define input/output file names:

By this point, one should have 3 input files: an RDP database FastA file, an RDP database Index file, and a query sequence(s) FastA file. RDPquery needs to know the name of each these files. To tell RDPquery the names of your input files and to define the name of your output file, go to the File names menu and change the necessary file names by selecting the corresponding number in the menu, pressing enter, then entering the new file name.

Note: The default names for the three input files are RDPdatabaseFile.fasta, RDPindexFile.txt, and queryseqs.fa. To avoid having to change the input file names each time RDPquery is run, one can rename his/her files to match.

The output file for RDPquery is a tab-delimited text file, so opening it with a spreadsheet application is a convenient way to read it. Keep in mind that if the output file name that you designate already exists, the existing file will be overwritten.

Note: If the existing output file that is being overwritten is larger than the output file overwriting it, the existing file will only be partially overwritten.

3) Set options:

The user will now define the options to use for this run of RDPquery. For descriptions of each of the options, see the “RDPquery Options” section above.

4) Begin:

Select the begin option from the main option menu and press enter. In a test run of RDPquery, 574 sequences with ~650 bases/sequence took 16 minutes and 46 seconds (8:15 of which was waiting for RDP, the rest was performing local comparisons).

Note: This was performed on a Windows XP machine with a Pentium 4 2GHz and 1GB of RAM.

Output file descriptors:

The output file name is only partially definable by the user. The text within the parenthesis of the output file name is automatically generated and inserted into the file name by RDPquery. The automatically inserted text contains the pertinent options that were set when the program was executed.

E.g.: “(gt1200,10,both,both,12.0,4.0,EXC)”

The each element of the text in the parenthesis means the following (from left to right):

- 1) Size of RDP match sequences:
 - gt1200 – Greater than or equal to 1200 base pairs
 - lt1200 – Less than 1200 base pairs
 - both – Any size
- 2) Number of RDP matches to request (within the range 1-20)
- 3) Source (isolates, uncultured, or both)
- 4) Strain (type, nontype, or both)
- 5) Open gap penalty
- 6) Extend gap penalty
- 7 (optional)) Whether range(s) of accession numbers or a custom regular expression was used to exclude GenBank accession numbers(‘EXC’ if Yes, blank if No).

Output file column headers:

- Exclusions* (optional): Excluded range(s)/custom regular expression used to exclude GenBank accessions.
- Heading*: The FastA header for the query sequence.
- Hit_Rank*: The rank, based on *Similarity*, of the RDP sequence out of the number of RDP sequences displayed for the query sequence.
- Query_Seq*: The sequence of the query.
- Query_Seq_Bases*: The number of bases for the query sequence.
- Compare_Date*: The date and time the comparison between the RDP sequence and the query sequence was performed.
- RDP_id*: The unique ID of the RDP sequence.
- RDP_SAB*: The SAB value RDP returned for the comparison between the RDP and the query sequence.
- Similarity*: The similarity value given by the local comparison.
- RDP_Name*: The name of the RDP sequence.
- GenBank_Accession*: The GenBank accession number of the RDP sequence.
- RDP_Seq*: The sequence of the RDP sequence.
- RDP_Bases*: The number of bases for the RDP sequence.
- The *domain*, *phylum*, *class*, *order*, *family*, *genus*, and *species* columns contain the taxonomic information for the RDP sequence at their respective taxonomic levels. The *species* column, however, is not necessarily the species of the RDP sequence. Rather, it is the descriptive text from the *RDP_Name* prior to the various accession numbers.
- The *domain(>n)*, *phylum(>n)*, *class(>n)*, *order(>n)*, *family(>n)*, *genus(>n)*, and *species(>=n)* columns contain the taxonomic information for the RDP sequence at their respective

taxonomic levels only if the Similarity between the RDP sequence and the query sequence meets the cutoff value for the taxonomic level. The species column, however, is not necessarily the species of the RDP sequence. Rather, it is the descriptive text from the RDP_Name prior to the various accession numbers.

FAQ:

1) **Q:** I get a "java.lang.OutOfMemoryError: Java heap space" error and the program crashes. What gives?

A: Try using a FastA file with fewer sequences. When one loads 500+ sequences, the results can become too large for the space in memory allotted to Java and crash the JVM.

If you absolutely need to run all sequences at once, and you have a moderately beefy computer, you can use this command to execute the program:

```
java -Xmx200m -jar RDPquery_2_7.jar
```

This command increases the amount of memory that Java is allowed to take up to 200MB. One could raise it even further if necessary (I use `-Xmx400m`), but the program will become an increasingly large drain on your computer. The primary reason that this program requires so much memory is the fact that the HTML trees returned by RDP are quite large (300 trees = ~12MB) and become even larger if one increases the number of RDP matches.

To decrease the amount of memory used, one could also reduce the number of RDP match sequences. This, however, may reduce RDPquery's effectiveness in identifying the highest %S match if the number of RDP matches is lowered too much (below 10).

2) **Q:** What is indexing, and why do I need to index the RDP database FastA file?

A: Indexing a file creates a series of pointers to desired locations within the file – making searching much more efficient. The RDP database FastA file is over 100MB and, therefore, is difficult to load into memory all at once. For this reason, it is necessary to create a series of pointers to specific regions of the FastA file so that one can go directly to the desired location without loading the entire file. The indexing step does, however, take a fair bit of time. Because it is a time consuming step, the indices are stored in the RDP database Index file for future use. This file, which is about 60 times smaller than the full RDP database FastA file, can be loaded fairly quickly when the program executes and is used as a reference when the program needs to retrieve a sequence file from the RDP database FastA file. The RDP database Index file must be periodically recreated when RDP releases new versions of its database. One will likely not know that RDP has released a new version of its database until RDPquery begins failing to finish running – when this happens, repeat the “Before you begin” process and Step 1 of “Running RDPquery”.

3) **Q:** Why is it that not all of the non-limited taxonomic columns contain identifications for all rows?

A: The non-limited taxonomic columns contain only the data provided by the RDP tree. If a sequence is unclassified at, for instance, the family level, it will have data up to the family column, then a blank genus column. It will, however, have species information because the species column is filled by the RDP name, which always exists in the RDP tree (e.g. “uncultured gamma proteobacterium”).

4) **Q:** Why can't I open the results file?

A: The most likely cause is that the combined path and file name of the results file is too long. In Windows, the combined path/file name can be no longer than 244 characters. To retrieve this file, one must cut the folder containing it, and paste that folder into a folder that is higher on the directory tree. For example, assuming that my path/file name is “C:\TaxonomicPrograms\LongShots\RDPquery\results.xls” and that this path/file name combination is too long, one should cut the folder \RDPquery\ and paste it into “C:\TaxonomicPrograms\”. The new path/file name is now “C:\TaxonomicPrograms\RDPquery\results.xls”, which may now be short enough to open.

5) **Q:** What does it mean when RDPquery says that “RDP dropped job.”?

A: Occasionally the RDP stops working on jobs and they need to be resubmitted. RDPquery should automatically resubmit the group. However, if the run fails, try rerunning the program. In our experience, small jobs may run more smoothly than large ones.

6) **Q:** How can I exit the program while I am in the middle of a run?

A: To exit a java application while it is running, press and hold the “Ctrl” key, then press “c”. Doing so will exit the application, and any data stored in RAM pertaining to the current run will be lost.

References:

¹Ahmed Moustafa, *JAligner: Open source Java implementation of Smith-Waterman*, <http://jaligner.sourceforge.net> September 23, 2004.

²Garrity, G.M., and J.G. Holt (2001) The road map to the Manual, in Bergey's Manual of Systematic Bacteriology, second edition, vol. 1., p. 119-166.

³Cole JR, Chai B, Farris RJ, Wang Q, Kulam SA, McGarrell DM, Garrity GM, Tiedje JM. The Ribosomal Database Project (RDP-II): sequences and tools for high-throughput rRNA analysis. *Nucleic Acids Res* 2005 Jan 1;33(Database Issue):D294-D296. doi: 10.1093/nar/gki038.