Package 'DBTC'

July 21, 2025

Type Package

Title Dada-BLAST-Taxon Assign-Condense Metabarcode Analysis

Version 0.1.0

Maintainer Robert G Young <rgyoung6@gmail.com>

Description First using 'dada2' R tools to analyse metabarcode data, the 'DBTC' package then uses the BLAST algorithm to search unknown sequences against local databases, and then takes reduced matched results and provides best taxonomic assignments.

License GPL-2 | GPL-3

Encoding UTF-8

URL <https://github.com/rgyoung6/DBTC>

Imports dada2, ggplot2, parallel, pbapply, plyr, ShortRead, stats, taxonomizr, utils

Language en-GB

RoxygenNote 7.3.1

NeedsCompilation no

Author Robert G Young [aut, cre, cph] (ORCID: <https://orcid.org/0000-0002-6731-2506>)

Repository CRAN

Date/Publication 2024-04-27 18:20:07 UTC

Contents

---

combine_assign_output   *Combine Taxa Assignment for Same ASV Using Different Databases*

Description

This function takes a file selection and then uses all 'taxaAssign' files in that directory and combines them into a single output 'taxaAssignCombined.tsv' file.

Usage

```
combine_assign_output(fileLoc = NULL, numCores = 1, verbose = TRUE)
```

Arguments

fileLoc	The location of a file in a directory where all of the 'taxaAssign' files are located (Default NULL).
numCores	The number of cores used to run the function (Default 1, Windows systems can only use a single core)
verbose	If set to TRUE then there will be output to the R console, if FALSE then this reporting data is suppressed (Default TRUE).

Details

The User Input: This function requires a file in a directory where all 'taxaAssign' files in that directory will be combined.

The examples are present to display the syntax for the function. These examples are not run because there are files required to run the functions, in some cases multiple files are necessary and some of these are quite large. To get specific examples please see https://github.com/rgyoung6/DBTCShinyTutorial/blob/main/READ

Value

This function produces a 'YYYY_MM_DD_HHMM_taxaAssignCombined.tsv' and a 'YYYY_MM_DD_HHMM_taxaAssi
file in the selected target directory.

Note

WARNING - NO WHITESPACE!

When running DBTC functions the paths for the files selected cannot have white space! File folder locations should be as short as possible (close to the root as some functions do not process long naming conventions.

Also, special characters should be avoided (including question mark, number sign, exclamation mark). It is recommended that dashes be used for separations in naming conventions while retaining underscores for use as information delimiters (this is how DBTC functions use underscore).

There are several key character strings used in the DBTC pipeline, the presence of these strings in file or folder names will cause errors when running DBTC functions.

The following strings are those used in DBTC and should not be used in file or folder naming: - _BLAST - _combinedDada - _taxaAssign - _taxaAssignCombined - _taxaReduced - _Combine-TaxaReduced

Author(s)

Robert G. Young

References

<https://github.com/rgyoung6/DBTC> Young, R. G., Hanner, R. H. (Submitted October 2023). Dada-BLAST-Taxon Assign-Condense Shiny Application (DBTCShiny). Biodiversity Data Journal.

See Also

dada_implement() combine_dada_output() make_BLAST_DB() seq_BLAST() taxon_assign() reduce_taxa() combine_reduced_output()

Examples

```
## Not run:
combine_assign_output()
combine_assign_output(fileLoc = NULL,    numCores = 1)

## End(Not run)
```

---

combine_dada_output   *Combine Dada Output*

Description

This function uses DBTC dada_implement ASV output files (YYYY_MM_DD_HH_MM_UserInputRunName_Merge, YYYY_MM_DD_HH_MM_UserInputRunName_MergeFwdRev, and/or YYYY_MM_DD_HH_MM_UserInputRunName_ and combines them into a single ASV table with accompanying fasta file. This function also produces a file containing the processing information for the function. The main input argument for this function is the location of a file in a folder containing all ASV tables wanting to be combined. Output files are generated with the naming convention YYYY_MM_DD_HH_MM_combinedDada.

Usage

```
combine_dada_output(fileLoc = NULL, minLen = 100, verbose = TRUE)
```

Arguments

fileLoc	Select a file in the file folder with dada_implement() results you would like to combine (YYYY_MM_DD_HHMM_FileName_MergeFwdRev OR YYYY_MM_DD_HHMM_FileNar both .tsv and .fas files (Default NULL).
minLen	The minimum final desired length of the read (Default 100).
verbose	If set to TRUE then there will be output to the R console, if FALSE then this reporting data is suppressed (Default TRUE).

Details

Two or more files to be combined are required as input for this function. These files need to be ASV files as outputted from the dada_implement() and can include Merge, MergeFwdRev, or To-talTable.tsv files. In addition, the user can input the desired minimum length of sequences that are wanted in the output combined file.

The examples are present to display the syntax for the function. These examples are not run because there are files required to run the functions, in some cases multiple files are necessary and some of these are quite large. To get specific examples please see https://github.com/rgyoung6/DBTCShinyTutorial/blob/main/READ.

Value

The output from this function includes three files. 1. YYYY_MM_DD_HHMM_combinedDada.tsv - combined ASV table 2. YYYY_MM_DD_HHMM_combinedDada.fas - combined fasta file 3. YYYY_MM_DD_HHMM_combinedDada.txt - Summary file from the combine_dada_output run

Note

WARNING - NO WHITESPACE!

When running DBTC functions the paths for the files selected cannot have white space! File folder locations should be as short as possible (close to the root as some functions do not process long naming conventions.

Also, special characters should be avoided (including question mark, number sign, exclamation mark). It is recommended that dashes be used for separations in naming conventions while retaining underscores for use as information delimiters (this is how DBTC functions use underscore).

There are several key character strings used in the DBTC pipeline, the presence of these strings in file or folder names will cause errors when running DBTC functions.

The following strings are those used in DBTC and should not be used in file or folder naming: - _BLAST - _combinedDada - _taxaAssign - _taxaAssignCombined - _taxaReduced - _Combine-TaxaReduced

Author(s)

Robert G. Young

References

<https://github.com/rgyoung6/DBTC> Young, R. G., Hanner, R. H. (Submitted October 2023). Dada-BLAST-Taxon Assign-Condense Shiny Application (DBTCShiny). Biodiversity Data Journal.

See Also

dada_implement() make_BLAST_DB() seq_BLAST() taxon_assign() combine_assign_output() re-
duce_taxa() combine_reduced_output()

Examples

```
## Not run:
combine_dada_output()
combine_dada_output(fileLoc = NULL, minLen = 100)

## End(Not run)
```

---

combine_reduced_output

Combine Reduce Taxa Files for the Same Biological Samples using
Different Markers

Description

This function takes a file selection and then uses all 'taxaReduced' files in that directory and com-
bines them into a single taxa table file with presence absence results.The output file is named with
the string _CombineTaxaReduced.tsv

Usage

```
combine_reduced_output(fileLoc = NULL, presenceAbsence = TRUE, verbose = TRUE)
```

Arguments

fileLoc   The location of a file in a directory where all of the 'taxa_assign' and/or 'com-
               bined_taxa_assign' files are located (Default NULL).

presenceAbsence

               This setting is a TRUE or FASLE value that indicates if the results will include
               read counts or be reduced to 0/1 presence absence values (Default TRUE)

verbose   If set to TRUE then there will be output to the R console, if FALSE then this
               reporting data is suppressed (Default TRUE).

Details

The User Input: This function requires a file in a directory where all 'taxaReduced' files in that
directory will be combined. The output format will be a taxa table with all taxa from all files
combined into a single table with presence absence (0 or 1) results. The value metrics for the
identification of the taxa from each combined file will remain in a column with the parenthetical
results from the 'taxaReduced' files ("Num_Rec", "Coverage", "Identity", "Max_eVal").

The examples are present to display the syntax for the function. These examples are not run because
there are files required to run the functions, in some cases multiple files are necessary and some of
these are quite large. To get specific examples please see https://github.com/rgyoung6/DBTCShinyTutorial/blob/main/READ

Value

This function produces a single 'YYYY_MM_DD_HHMM_CombineTaxaReduced' file and associated summary file in the target directory.

Note

WARNING - NO WHITESPACE!

When running DBTC functions the paths for the files selected cannot have white space! File folder locations should be as short as possible (close to the root as some functions do not process long naming conventions.

Also, special characters should be avoided (including question mark, number sign, exclamation mark). It is recommended that dashes be used for separations in naming conventions while retaining underscores for use as information delimiters (this is how DBTC functions use underscore).

There are several key character strings used in the DBTC pipeline, the presence of these strings in file or folder names will cause errors when running DBTC functions.

The following strings are those used in DBTC and should not be used in file or folder naming: - _BLAST - _combinedDada - _taxaAssign - _taxaAssignCombined - _taxaReduced - _CombineTaxaReduced

Author(s)

Robert G. Young

References

<https://github.com/rgyoung6/DBTC> Young, R. G., Hanner, R. H. (Submitted October 2023). Dada-BLAST-Taxon Assign-Condense Shiny Application (DBTCShiny). Biodiversity Data Journal.

See Also

dada_implement() combine_dada_output() make_BLAST_DB() seq_BLAST() taxon_assign() combine_assign_output() reduce_taxa()

Examples

```
## Not run:
combine_reduced_output()
combine_reduced_output(fileLoc = NULL, presenceAbsence = TRUE)

## End(Not run)
```

dada_implement   *Dada Implement*

Description

This function requires a main directory containing a folder(s) representing sequencing runs which in-turn contain fastq files (the location of one of the fastq files in one of the sequencing run folders is used as an input argument). All sequencing folders in the main directory need to represent data from sequencing runs that have used the same primers and protocols. Output from this function includes all processing files and final main output files in the form of fasta files and amplicon sequencing variant (ASV) tables.

Usage

```
dada_implement(
  runFolderLoc = NULL,
  primerFile = NULL,
  fwdIdent = "_R1_001",
  revIdent = "_R2_001",
  unidirectional = FALSE,
  bidirectional = TRUE,
  printQualityPdf = TRUE,
  maxPrimeMis = 2,
  fwdTrimLen = 0,
  revTrimLen = 0,
  maxEEVal = 2,
  truncQValue = 2,
  truncLenValueF = 0,
  truncLenValueR = 0,
  error = 0.1,
  nbases = 1e+80,
  maxMismatchValue = 0,
  minOverlapValue = 12,
  trimOverhang = FALSE,
  minFinalSeqLen = 100,
  verbose = TRUE
)
```

Arguments

runFolderLoc	Select a file in the one of the run folders with the fastq files of interest (Default NULL).
primerFile	Select a file with the primers for this analysis (Default NULL).
fwdIdent	Forward identifier naming string (Default '_R1_001').
revIdent	Reverse identifier naming string (Default '_R2_001').
unidirectional	Selection to process files independently (Default FALSE).

bidirectional	Selection to process paired forward and reverse sequence for analysis (Default TRUE).
printQualityPdf	
	Selection to process save image files showing quality metrics (Default TRUE).
maxPrimeMis	Maximum number of mismatches allowed when pattern matching trimming the primers from the ends of the reads for the ShortRead trimLRPatterns() function (Default 2).
fwdTrimLen	Select a forward trim length for the Dada filterAndTrim() function (Default 0).
revTrimLen	Select a reverse trim length for the Dada filterAndTrim() function (Default 0).
maxEEVal	Maximum number of expected errors allowed in a read for the Dada filterAndTrim() function (Default 2).
truncQValue	Truncation value use to trim ends of reads, nucleotides with quality values less than this value will be used to trim the remainder of the reads for the Dada filterAndTrim() function (Default 2).
truncLenValueF	Dada forward length trim value for the Dada filterAndTrim() function. This function is set to 0 when the pattern matching trim function is enabled (Default 0).
truncLenValueR	Dada reverse length trim value for the Dada filterAndTrim() function. This function is set to 0 when the pattern matching trim function is enabled (Default 0).
error	Percent of fastq files used to assess error rates for the Dada learnErrors() function (Default 0.1).
nbases	The total number of bases used to assess errors for the Dada learnErrors() function (Default 1e80) NOTE: this value is set very high to get all nucleotides in the error present file subset. If the error is to be assessed using total reads and not specific fastq files then set the error to 1 and set this value to the desired number of reads.
maxMismatchValue	
	Maximum number of mismatches allowed when merging two reads for the Dada mergePairs() function (Default 2).
minOverlapValue	
	Minimum number of overlapping nucleotides for the forward and reverse reads for the Dada mergePairs() function (Default 12).
trimOverhang	Trim merged reads past the start of the complimentary primer regions for the Dada mergePairs() function (Default FALSE).
minFinalSeqLen	The minimum final desired length of the read (Default 100).
verbose	If set to TRUE then there will be output to the R console, if FALSE then this reporting data is suppressed (Default TRUE).

Details

Two file types are required as input for the dada_implement() function. The first are the fastq files in the appropriate folder structure (see below) and the second is a file containing the primers used for the amplification of the sequence reads.

Fastq File Folder Structure

Parent Directory | | ————————— | | | | Run1 Directory Run2 Directory -Fastq -Fastq -Fastq -Fastq
... ...

Format of the primer file

| Forward | Reverse | | AGTGTGTAGTGATTG | CGCATCGCTCAGACTGACTGC | | GAGCC-
CTCGATCGCT | GGTCGATAGCTACGCGCGCATACGACT | | | GGTTCACATCGCATTCAT |

The examples are present to display the syntax for the function. These examples are not run because
there are files required to run the functions, in some cases multiple files are necessary and some of
these are quite large. To get specific examples please see https://github.com/rgyoung6/DBTCShinyTutorial/blob/main/READ

Value

The output from this function includes four folders. A_Qual - Contains quality pdf files for the input
fastq files (if printQualityPdf set to TRUE). B_Filt - Contains dada filtered fastq files and a folder
with the end trimmed fastq files before quality filtering. C_FiltQual - Contains quality pdf files
for the filtered fastq files (if printQualityPdf set to TRUE). D_Output - This folder contains output
files including and analysis summary, an analysis summary table of processing values, forward
and reverse error assessments, and finally the output ASV and fasta files of obtained sequences.
-TotalTable.tsv

Note

WARNING - NO WHITESPACE!

When running DBTC functions the paths for the files selected cannot have white space! File folder
locations should be as short as possible (close to the root as some functions do not process long
naming conventions.

Also, special characters should be avoided (including question mark, number sign, exclamation
mark). It is recommended that dashes be used for separations in naming conventions while retaining
underscores for use as information delimiters (this is how DBTC functions use underscore).

There are several key character strings used in the DBTC pipeline, the presence of these strings in
file or folder names will cause errors when running DBTC functions.

The following strings are those used in DBTC and should not be used in file or folder naming: -
_BLAST - _combinedDada - _taxaAssign - _taxaAssignCombined - _taxaReduced - _Combine-
TaxaReduced

Author(s)

Robert G. Young

References

<https://github.com/rgyoung6/DBTC> Young, R. G., Hanner, R. H. (Submitted October 2023).
Dada-BLAST-Taxon Assign-Condense Shiny Application (DBTCShiny). Biodiversity Data Jour-
nal.

See Also

combine_dada_output() make_BLAST_DB() seq_BLAST() taxon_assign() combine_assign_output()
reduce_taxa() combine_reduced_output()

Examples

```
## Not run:
dada_implement()
dada_implement(runFolderLoc = NULL, primerFile = NULL,fwdIdent = "_R1_001",
revIdent = "_R2_001",unidirectional = FALSE, bidirectional = TRUE, printQualityPdf = TRUE,
maxPrimeMis = 2, fwdTrimLen = 0, revTrimLen = 0,maxEEVal=2, truncQValue = 2,
truncLenValueF = 0, truncLenValueR = 0,error = 0.1, nbases = 1e80,
maxMismatchValue = 0, minOverlapValue = 12,trimOverhang = FALSE,
minFinalSeqLen = 100)

## End(Not run)
```

---

make_BLAST_DB   *Make a BLAST Database*

Description

This function takes a fasta file (in MACER format) and establishes a database upon which a BLAST search can be completed.

Usage

```
make_BLAST_DB(
  fileLoc = NULL,
  makeblastdbPath = "makeblastdb",
  taxaDBLoc = NULL,
  dbName = NULL,
  minLen = 100,
  verbose = TRUE
)
```

Arguments

fileLoc	The location of a file in a directory where all fasta files will be used to construct a BLASTable database (Default NULL).
makeblastdbPath	
	The local path for the blast+ makeblastdbPath program (Default 'makeblastdb').
taxaDBLoc	The location of the NCBI taxonomic data base (Default NULL; for accession-Taxa.sql see the main DBTC page for details).
dbName	A short 6-8 alpha character name used when building a database (Default NULL).
minLen	The minimum sequence length used to construct the BLAST database (Default 100).
verbose	If set to TRUE then there will be output to the R console, if FALSE then this reporting data is suppressed (Default TRUE).

Details

The user inputs the location of a file in a directory that contains a properly formatted fasta file which can be used to construct a BLASTable database. The NCBI blast+ program, makeblastdb and the NCBI taxonomic database (accessionTaxa.sql) are required to run this script (see readme instructions for details).

The examples are present to display the syntax for the function. These examples are not run because there are files required to run the functions, in some cases multiple files are necessary and some of these are quite large. To get specific examples please see https://github.com/rgyoung6/DBTCShinyTutorial/blob/main/READ

Value

The output from this function includes a folder with the BLAST database named according to the submitted dbName

Note

WARNING - NO WHITESPACE!

When running DBTC functions the paths for the files selected cannot have white space! File folder locations should be as short as possible (close to the root as some functions do not process long naming conventions.

Also, special characters should be avoided (including question mark, number sign, exclamation mark). It is recommended that dashes be used for separations in naming conventions while retaining underscores for use as information delimiters (this is how DBTC functions use underscore).

There are several key character strings used in the DBTC pipeline, the presence of these strings in file or folder names will cause errors when running DBTC functions.

The following strings are those used in DBTC and should not be used in file or folder naming: - _BLAST - _combinedDada - _taxaAssign - _taxaAssignCombined - _taxaReduced - _Combine-TaxaReduced

Author(s)

Robert G. Young

References

<https://github.com/rgyoung6/DBTC> Young, R. G., Hanner, R. H. (Submitted October 2023). Dada-BLAST-Taxon Assign-Condense Shiny Application (DBTCShiny). Biodiversity Data Journal.

See Also

dada_implement() combine_dada_output() seq_BLAST() taxon_assign() combine_assign_output() reduce_taxa() combine_reduced_output()

Examples

```
## Not run:
make_BLAST_DB()
make_BLAST_DB(fileLoc = NULL, makeblastdbPath = "makeblastdb", taxaDBLoc = NULL,
inputFormat = NULL, dbName = NULL, minLen = 100)

## End(Not run)
```

---

reduce_taxa   *Reduce Taxa Assignment*

Description

This function takes a file selection and then uses all '_taxaAssign_YYYY_MM_DD_HHMM.tsv'
and/or 'YYYY_MM_DD_HHMM_taxaAssignCombined.tsv' files in that directory and reduces all
ASV with the same taxonomic assignment into a single taxonomic result for each submitted file.
The results are then placed in to a '_taxaReduced_YYYY_MM_DD_HHMM.tsv' file for each of
the target files in the directory.

Usage

```
reduce_taxa(fileLoc = NULL, numCores = 1, verbose = TRUE)
```

Arguments

fileLoc	The location of a file in a directory where all of the 'taxaAssign' and/or 'taxaAssignCombine' files are located (Default NULL).
numCores	The number of cores used to run the function (Default 1, Windows systems can only use a single core)
verbose	If set to TRUE then there will be output to the R console, if FALSE then this reporting data is suppressed (Default TRUE).

Details

This function requires a file in a directory where all '_taxaAssign_YYYY_MM_DD_HHMM.tsv'
and/or 'YYYY_MM_DD_HHMM_taxaAssignCombined.tsv' files in that directory will be com-
bined. All records with the same taxonomic result will be combined. The BLAST values in paren-
theses ("Num_Rec", "Coverage", "Identity", "Max_eVal") are combine by the mean number of
records, the mean of the minimum coverage and identity values, and the mean of the maximum
eValues.

The examples are present to display the syntax for the function. These examples are not run because
there are files required to run the functions, in some cases multiple files are necessary and some of
these are quite large. To get specific examples please see https://github.com/rgyoung6/DBTCShinyTutorial/blob/main/READ

Value

This function produces a 'taxa_reduced' file for every 'taxaAssign' or 'taxaAssignCombine' present in the target directory.

Note

WARNING - NO WHITESPACE!

When running DBTC functions the paths for the files selected cannot have white space! File folder locations should be as short as possible (close to the root as some functions do not process long naming conventions.

Also, special characters should be avoided (including question mark, number sign, exclamation mark). It is recommended that dashes be used for separations in naming conventions while retaining underscores for use as information delimiters (this is how DBTC functions use underscore).

There are several key character strings used in the DBTC pipeline, the presence of these strings in file or folder names will cause errors when running DBTC functions.

The following strings are those used in DBTC and should not be used in file or folder naming: - _BLAST - _combinedDada - _taxaAssign - _taxaAssignCombined - _taxaReduced - _Combine-TaxaReduced

Author(s)

Robert G. Young

References

<https://github.com/rgyoung6/DBTC> Young, R. G., Hanner, R. H. (Submitted October 2023). Dada-BLAST-Taxon Assign-Condense Shiny Application (DBTCShiny). Biodiversity Data Journal.

See Also

dada_implement() combine_dada_output() make_BLAST_DB() seq_BLAST() taxon_assign() combine_assign_output() combine_reduced_output()

Examples

```
## Not run:
reduce_taxa()
reduce_taxa(fileLoc = NULL,   numCores = 1)

## End(Not run)
```

---

seq_BLAST   *BLAST Query File Against Local Database*

Description

This function takes fasta files as input along with a user selected NCBI formatted database to
BLAST sequences against. The outcome of the function are two files, a BLAST run file and a
single file containing all of the BLAST results in tab delimited format (Note: there are no headers
but the columns are, query sequence ID, search sequence ID, search taxonomic ID, query to se-
quence coverage, percent identity, search scientific name, search common name, query start, query
end, search start, search end, e-value.

Usage

```
seq_BLAST(
  databasePath = NULL,
  querySeqPath = NULL,
  blastnPath = "blastn",
  minLen = 100,
  BLASTResults = 200,
  numCores = 1,
  verbose = TRUE
)
```

Arguments

databasePath	The location of a file in a directory where the desired BLAST database is located (Default NULL).
querySeqPath	The location of a file in a directory containing all of the fasta files wishing to be BLASTed (Default NULL).
blastnPath	The location of the NCBI blast+ blastn program (Default 'blastn').
minLen	The minimum length of the sequences that will be BLASTed (Default 100).
BLASTResults	The number of returned results, or the depth of the reported results, saved from the BLAST (Default 200).
numCores	The number of cores used to run the function (Default 1, Windows systems can only use a single core).
verbose	If set to TRUE then there will be output to the R console, if FALSE then this reporting data is suppressed (Default TRUE).

Details

The user input provides a location for the BLAST database you would like to use by selecting a
file in the target directory. Then provide the location of the query sequence file(s) by indicating a
file in a directory that contains the fasta file(s) of interest. Provide the path for the blast+ blastn
program. Finally, provide the minimum query sequence length to BLAST (Default 100), the depth

of the BLAST returned results (default 200), and finally the number of cores to process the function
(Default 1, Windows implementation will only accept this value as 1).

The examples are present to display the syntax for the function. These examples are not run because
there are files required to run the functions, in some cases multiple files are necessary and some of
these are quite large. To get specific examples please see https://github.com/rgyoung6/DBTCShinyTutorial/blob/main/READ

Value

Two files are produced from this function, a BLAST run file and a BLAST results file for each of
the fasta files in the target directory.

Note

WARNING - NO WHITESPACE!

When running DBTC functions the paths for the files selected cannot have white space! File folder
locations should be as short as possible (close to the root as some functions do not process long
naming conventions.

Also, special characters should be avoided (including question mark, number sign, exclamation
mark). It is recommended that dashes be used for separations in naming conventions while retaining
underscores for use as information delimiters (this is how DBTC functions use underscore).

There are several key character strings used in the DBTC pipeline, the presence of these strings in
file or folder names will cause errors when running DBTC functions.

The following strings are those used in DBTC and should not be used in file or folder naming: -
_BLAST - _combinedDada - _taxaAssign - _taxaAssignCombined - _taxaReduced - _Combine-
TaxaReduced

Author(s)

Robert G. Young

References

<https://github.com/rgyoung6/DBTC> Young, R. G., Hanner, R. H. (Submitted October 2023).
Dada-BLAST-Taxon Assign-Condense Shiny Application (DBTCShiny). Biodiversity Data Jour-
nal.

See Also

dada_implement() combine_dada_output() make_BLAST_DB() taxon_assign() combine_assign_output()
reduce_taxa() combine_reduced_output()

Examples

```
## Not run:
seq_BLAST()
seq_BLAST(databasePath = NULL, querySeqPath = NULL,  blastnPath = "blastn",
minLen = 100, BLASTResults = 200, numCores = 1)

## End(Not run)
```

---

taxon_assign   *Assign Taxa using BLAST Results*

Description

This function takes a BLAST result file and associated fasta files (either on their own or with accompanying ASV files generated from the dada_implement function) and collapses the multiple BLAST results into as single result for each query sequence. When an ASV table is present the taxonomic results will be combined with the ASV table.

Usage

```
taxon_assign(
  fileLoc = NULL,
  taxaDBLoc = NULL,
  numCores = 1,
  coverage = 95,
  ident = 95,
  propThres = 0.95,
  coverReportThresh = 0,
  identReportThresh = 0,
  includeAllDada = TRUE,
  verbose = TRUE
)
```

Arguments

fileLoc	The location of a file in a directory where all of the paired fasta and BLAST (and potentially ASV) files are located (Default NULL).
taxaDBLoc	The location of the NCBI taxonomic data base (Default NULL; for accession-Taxa.sql see the main DBTC page for details).
numCores	The number of cores used to run the function (Default 1, Windows systems can only use a single core).
coverage	The percent coverage used for taxonomic assignment for the above threshold results (Default 95).
ident	The percent identity used for the taxonomic assignment for above threshold results (Default 95).
propThres	The proportional threshold flags the final result based on the preponderance of the data. So if the threshold is set to 0.95, results will be flagged if the taxa directly below the assigned taxa has fewer than 0.95 percent of the records causing the upward taxonomic placement (Default 0.95).
coverReportThresh	
	The percent coverage threshold used for reporting flags below this threshold (Default 95).

identReportThresh

> The percent identity threshold used for reporting flags below this threshold (Default 95).

includeAllDada   When paired Dada ASV tables are present, when set to FALSE, this will exclude records without taxonomic assignment (Default TRUE).

verbose   If set to TRUE then there will be output to the R console, if FALSE then this reporting data is suppressed (Default TRUE).

Details

This function requires a BLAST output file and an associated fasta file. In addition, if present an ASV file will also be used and combined with the taxonomic results when present. The BLAST results are reduced to a single result for each read. At each taxonomic level there may be one or more taxonomic assignments. Each assignment has quality metrics in parentheses after the name. These values ("Num_Rec", "Coverage", "Identity", "Max_eVal") represent the number of records with this taxonomic placement, the minimum coverage and identity, and the maximum eValue for the reported taxa.

The examples are present to display the syntax for the function. These examples are not run because there are files required to run the functions, in some cases multiple files are necessary and some of these are quite large. To get specific examples please see https://github.com/rgyoung6/DBTCShinyTutorial/blob/main/READ

Value

This function produces a taxa_reduced file for each submitted BLAST-fasta submission.

Note

WARNING - NO WHITESPACE!

When running DBTC functions the paths for the files selected cannot have white space! File folder locations should be as short as possible (close to the root as some functions do not process long naming conventions.

Also, special characters should be avoided (including question mark, number sign, exclamation mark). It is recommended that dashes be used for separations in naming conventions while retaining underscores for use as information delimiters (this is how DBTC functions use underscore).

There are several key character strings used in the DBTC pipeline, the presence of these strings in file or folder names will cause errors when running DBTC functions.

The following strings are those used in DBTC and should not be used in file or folder naming: - _BLAST - _combinedDada - _taxaAssign - _taxaAssignCombined - _taxaReduced - _CombineTaxaReduced

Author(s)

Robert G. Young

References

<https://github.com/rgyoung6/DBTC> Young, R. G., Hanner, R. H. (Submitted October 2023). Dada-BLAST-Taxon Assign-Condense Shiny Application (DBTCShiny). Biodiversity Data Journal.

See Also

dada_implement() combine_dada_output() make_BLAST_DB() seq_BLAST() combine_assign_output() reduce_taxa() combine_reduced_output()

Examples

```
## Not run:
taxon_assign()
taxon_assign(fileLoc = NULL, taxaDBLoc = NULL, numCores = 1, coverage = 95,
ident = 95, propThres = 0.95, coverReportThresh=0, identReportThresh=0, includeAllDada=TRUE)

## End(Not run)
```

Index