BTD

- btd.nwalign — Needleman-Wunsch Global alignment of two nucleotide sequences
- btd.swalign — Smith Waterman Local alignment of two nucleotide sequences
- btd_CpG — CpG Islands (counts 'GC' dinucleotides in a DNA sequence)
- btd_DNAword — Number and positions of a DNA pattern in a DNA sequence
- btd_UpA — UpA Islands : counts 'AU'/'AT' dinucleotides in a DNA/RNA sequence
- btd_at2gcratio — AT/GC Ratio
- btd_atcontent — AT content of a DNA Sequence
- btd_basecomp — Frequency of indicidual nucleotide
- btd_codoncount — Count codons in a nucleotide sequence
- btd_codonusage — Frequency of each codon type
- btd_entropydna — Entropy of DNA
- btd_gccontent — Computes the GC content of a DNA sequence
- btd_isoelectricpoint — Isoelectric Point
- btd_kmer — K-mer counts of number of occurences of given DNA word/substring
- btd_ntdensity — Nucleotide Density
- btd_purinecomp — Purine Composition of a DNA/RNA sequence
- btd_pyrimidinecomp — Pyrimidine Composition
- btd_randntseq — Generates a random nucleotide sequence from finite alphabet
- btd_randprotseq — Generates a random protein sequence from finite alphabet
- btd_rho — Rho
- btd_seqcomplement — Complementary strand of a nucleotide sequence
- btd_seqrcomplement — reverse complementary strand of a nucleotide sequence
- btd_seqshoworfs — Open Reading Frames found in a nucleotide sequence
- Amino Acids
  - btd_aacount — Counts amino acids in sequence
  - btd_aadensity — Amino Acid Density
  - btd_atomiccomp — CHONS Atomic Composition of an amino acid sequence
  - btd_atomicweight — Atomic weight of C,N,O,H,S atoms in an amino acid sequence
  - btd_avg_hydrophobicity — Average Hydrophobicity of amino acid sequence
  - btd_basecount — Count nucleotides in a amino acid sequence
  - btd_cleave — Cleaves amino acid sequence with enzyme
  - btd_entropyamino — Entropy of Amino Acid
  - btd_ext_coefficients — Extinction coefficients of aminoacid sequence
  - btd_exterior_volume — Exterior Volume of amino acid sequence
  - btd_interior_volume — Interior Volume of aminoacid sequence
  - btd_lcdensity — Linear Charge Density of an aminoacid sequence
  - btd_molarabsortivity — Molar absortivity/absorbance of aminoacid sequence
  - btd_molweigth — Computes the molecular Weight of a amino acide sequence
  - btd_packingvolume — Volume of Amino Acid Sequence
  - btd_proteinradius — Radius of Amino Acid Sequence
  - btd_psv — Partial Specific Volume of a amino acid sequence
  - btd_volumeratio — Volume Ratio of a amino acid sequence
- Conversions
  - btd_aa2int — Amino acid to Integers
  - btd_aa2nt — Amino Acid Sequence to Nucleotide Sequence
  - btd_dna2rna — DNA sequence to RNA sequence
  - btd_int2aa — Integer to Amino Acid codes
  - btd_nt2aa — Nucleotide Sequence to Amino Acid
  - btd_rna2dna — Convert RNA sequence to DNA sequence
- Data bases
  - btd_getembl — Gets sequence information from the online EMBL database
  - btd_getgenbank — Gets sequence information from the online GenBank database
  - btd_getgenpept — Retrieve sequence information from GenPept database

- btd_getpdb — Retrieve protein structure data from Protein Data Bank (PDB) database
- btd_getuniprot — Retrieve sequence information from UniProt database

<< Conversions	BTD

BTD >> BTD > Data bases

Data bases

- btd_getembl — Gets sequence information from the online EMBL database
- btd_getgenbank — Gets sequence information from the online GenBank database
- btd_getgenpept — Retrieve sequence information from GenPept database
- btd_getpdb — Retrieve protein structure data from Protein Data Bank (PDB) database
- btd_getuniprot — Retrieve sequence information from UniProt database

btd_getembl

Gets sequence information from the online EMBL database

Calling Function

```
filename = btd_getembl(x)
filename = btd_getembl(x, filepath)
```

Arguments

x

Unique identifier for a sequence record

filepath

local Path + filename where the file returned by the online query must be stored. The directory must be writable. If only a path to directory in provided, the default filename used is built as `filepath+"/embl_"+x+".html"`.

filename

Name of the file returned by the query. By default (if no **`filepath`** is provided), it is named "dbfetch" and is stored in the current directory (that must be writeable).

Description

Retrieves sequence information from the online EMBL database, as a HTML file returned and stored locally.

Examples

```
fn = btd_getembl('X00558', TMPDIR)
winopen(fn);    // Opens the local file in your web browser
fileparts(fn, "fname")
```

```
--> fn = btd_getembl('X00558', TMPDIR)
 fn  =
 C:\Users\Me\AppData\Local\Temp\SCI_TMP_7176_10748\embl_X00558.html

--> winopen(fn);    // Opens the local file in your web browser

--> fileparts(fn, "fname")
 ans  =
 embl_X00558
```

See Also

- btd_getuniprot — Retrieve sequence information from UniProt database
- btd_getgenbank — Gets sequence information from the online GenBank database
- btd_getpdb — Retrieve protein structure data from Protein Data Bank (PDB) database
- btd_getgenpept — Retrieve sequence information from GenPept database
- htmlRead
- getPreferencesValue
- xmlXPath

History

Version	Description
1.1	Input option filepath added.

btd_getgenbank

Gets sequence information from the online GenBank database

Calling Function

```
filename = btd_getgenbank(x)
filename = btd_getgenbank(x, filepath)
```

Arguments

x

Unique identifier for a sequence record

filepath

local Path + filename where the file returned by the online query must be stored. The directory must be writable. If only a path to directory in provided, the default filename used is built as `filepath+"/genbank_"+x+".txt"`.

filename

Name of the file returned by the query. By default (if no **`filepath`** is provided), it is named "efetch.fcgi" and is stored in the current directory (that must be writeable).

Examples

```
fn = btd_getgenbank('M10051', TMPDIR)
// Open the local file in your text editor:
if getos()=="Windows", winopen(fn); else, edit(fn); end
fileparts(fn, "fname")
```

```
--> fn = btd_getgenbank('M10051', TMPDIR)
 fn  =
 C:\Users\Me\AppData\Local\Temp\SCI_TMP_7176_10748\genbank_M10051.txt

--> if getos()=="Windows", winopen(fn); else, edit(fn); end

--> fileparts(fn, "fname")
 ans  =
 genbank_M10051
```

See Also

- btd_getgenpept — Retrieve sequence information from GenPept database
- btd_getuniprot — Retrieve sequence information from UniProt database
- btd_getembl — Gets sequence information from the online EMBL database
- btd_getpdb — Retrieve protein structure data from Protein Data Bank (PDB) database

- mgetl
- grep

History

Version	Description
1.1	Input option filepath added.

btd_getgenpept

Retrieve sequence information from GenPept database

Calling Function

```
btd_getgenpept(x)
```

Arguments

x

Unique identifier for a sequence record

filepath

local Path + filename where the file returned by the online query must be stored. The directory must be writable. If only a path to directory in provided, the default filename used is built as `filepath+"/genpept_"+x+".txt"`.

filename

Name of the file returned by the query. By default (if no **`filepath`** is provided), it is named "efetch.fcgi" and is stored in the current directory (that must be writeable).

Description

Retrieve sequence information from GenPept database.

Examples

```
fn = btd_getgenpept('AAA59174', TMPDIR)
// Open the local file in your text editor:
if getos()=="Windows", winopen(fn); else, edit(fn); end
fileparts(fn, "fname")
```

```
--> fn = btd_getgenpept('AAA59174', TMPDIR)
 fn  =
 C:\Users\Me\AppData\Local\Temp\SCI_TMP_6764_11518\genpept_AAA59174.txt

--> if getos()=="Windows", winopen(fn); else, edit(fn); end

--> fileparts(fn, "fname")
 ans  =
 genpept_AAA59174
```

See Also

- btd_getgenbank — Gets sequence information from the online GenBank database

- btd_getuniprot — Retrieve sequence information from UniProt database
- btd_getembl — Gets sequence information from the online EMBL database
- btd_getpdb — Retrieve protein structure data from Protein Data Bank (PDB) database
- mgetl
- grep

History

Version	Description
1.1	Input option filepath added.

btd_getpdb

Retrieve protein structure data from Protein Data Bank (PDB) database

Calling Function

```
filename = btd_getpdb(x)
filename = btd_getpdb(x, filepath)
```

Arguments

x

Unique identifier for a sequence record

filepath

local Path + filename where the file returned by the online query must be stored. The directory must be writable. If only a path to directory in provided, the default filename used is built as `filepath+"/pdb_"+x+".txt"`.

filename

Name of the file returned by the query. By default (if no **`filepath`** is provided), it is named "downloadFile.do" and is stored in the current directory (that must be writeable).

Description

Retrieves protein structure data from the online Protein Data Bank (PDB) database. A live Internet connection is required.

Examples

```
fn = btd_getpdb('5CYT', TMPDIR)
// Open the local file in your text editor:
if getos()=="Windows", winopen(fn); else, edit(fn); end
fileparts(fn, "fname")
```

```
--> fn = btd_getpdb('5CYT', TMPDIR)
 fn  =
 C:\Users\Me\AppData\Local\Temp\SCI_TMP_7176_10748\pdb_5CYT.txt

--> if getos()=="Windows", winopen(fn); else, edit(fn); end

--> fileparts(fn, "fname")
 ans  =
 pdb_5CYT
```

See Also

- btd_getgenbank — Gets sequence information from the online GenBank database
- btd_getuniprot — Retrieve sequence information from UniProt database
- btd_getembl — Gets sequence information from the online EMBL database
- btd_getgenpept — Retrieve sequence information from GenPept database
- mgetl
- grep

History

Version	Description
1.1	Input option filepath added.

btd_getuniprot

Retrieve sequence information from UniProt database

Calling Function

```
filename = btd_getuniprot(x)
filename = btd_getuniprot(x, filepath)
```

Arguments

x

Unique identifier for a sequence record

filepath

local Path + filename where the file returned by the online query must be stored. The directory must be writable. If only a path to directory in provided, the default filename used is built as `filepath+"/uniprot_"+x+".html"`.

filename

Name of the file returned by the query. By default (if no **`filepath`** is provided), it is named as the sequence **`x`** and is stored in the current directory (that must be writeable).

Description

Retrieve sequence information from UniProt database

Examples

```
fn = btd_getuniprot('P31946', TMPDIR);
winopen(fn);    // Opens the local file in your web browser
fileparts(fn, "fname")
```

```
--> fn = btd_getuniprot('P31946', TMPDIR)
 fn  =
 C:\Users\Me\AppData\Local\Temp\SCI_TMP_7176_10748\uniprot_P31946.html

--> winopen(fn);    // Opens the local file in your web browser

--> fileparts(fn, "fname")
 ans  =
uniprot_P31946
```

See Also

- btd_getgenbank — Gets sequence information from the online GenBank database

- btd_getgenpept — Retrieve sequence information from GenPept database
- btd_getembl — Gets sequence information from the online EMBL database
- btd_getpdb — Retrieve protein structure data from Protein Data Bank (PDB) database
- htmlRead
- getPreferencesValue
- xmlXPath

History

Version	Description
1.1	Input option filepath added.

Conversions

- btd_aa2int — Amino acid to Integers
- btd_aa2nt — Amino Acid Sequence to Nucleotide Sequence
- btd_dna2rna — DNA sequence to RNA sequence
- btd_int2aa — Integer to Amino Acid codes
- btd_nt2aa — Nucleotide Sequence to Amino Acid
- btd_rna2dna — Convert RNA sequence to DNA sequence

btd_aa2int

Amino acid to Integers

Calling Function

```
y = btd_aa2int(x)
```

Arguments

x

An amino acid string sequence

y

Row vector of decimal integers representing the Amino acid sequence.

Examples

```
y = btd_aa2int('AGCTTGGCAAT')
```

```
--> y = btd_aa2int('AGCTTGGCAAT')
 y  =

   1.   8.   5.   17.   17.   8.   8.   5.   1.   1.   17.
```

See Also

- btd_int2aa — Integer to Amino Acid codes

btd_aa2nt

Amino Acid Sequence to Nucleotide Sequence

Calling Function

```
y = btd_aa2nt(x)
```

Arguments

x

String of single-letter codes specifying an amino acid sequence

y

Nucleotide sequence specified by a character string of letter codes.

Description

Converts a amino acid sequence to nucleotide sequence

Examples

```
y = btd_aa2nt('KAEPLLQ')
```

```
--> y = btd_aa2nt('KAEPLLQ')
 y  =

 AAAGCGGAACCTCTACTACAG
```

See Also

- btd_nt2aa — Nucleotide Sequence to Amino Acid

btd_dna2rna

DNA sequence to RNA sequence

Calling Function

```
y = btd_dna2rna(x)
```

Arguments

x

String of single-letter codes specifying a DNA sequence

y

RNA sequence

Description

Convert DNA sequence to RNA sequence

Examples

```
y = btd_dna2rna('AGCTTGGCAAT')
```

```
--> y = btd_dna2rna('AGCTTGGCAAT')
 y  =

 AGCUUGGCAAU
```

See Also

- btd_rna2dna — Convert RNA sequence to DNA sequence

btd_int2aa

Integer to Amino Acid codes

Calling Function

```
y = btd_int2aa(x)
```

Arguments

x

Row vector of integers specifying an amino acid sequence.

y

Amino acid sequence specified by a vector of single-letter codes/characters

Description

Convert amino acid sequence from integer to letter representation

Examples

```
y = btd_int2aa([1 8 5 17 17 8 8 5 1 1 17])
```

```
--> y = btd_int2aa([1 8 5 17 17 8 8 5 1 1 17])
 y  =
!A  G  C  T  T  G  G  C  A  A  T  !

--> size(y)
 ans  =
   1.   11.
```

See Also

- btd_aa2int — Amino acid to Integers

btd_nt2aa

Nucleotide Sequence to Amino Acid

Calling Function

```
y = btd_nt2aa(x)
```

Arguments

x

String of single-letter codes specifying a nucleotide sequence

y

Amino acid sequence specified by a single string of single-letter codes

Description

Converts a nucleotide sequence to amino acid sequence

Examples

```
y = btd_nt2aa('AAAGCTGAACCTTTGCTGCAGGA')
```

```
--> y = btd_nt2aa('AAAGCTGAACCTTTGCTGCAGGA')

 transcribed DNA is AAAGCUGAACCUUUGCUGCAGGA
 y  =

 KAEPLLQ
```

See Also

- btd_aa2nt — Amino Acid Sequence to Nucleotide Sequence

btd_rna2dna

Convert RNA sequence to DNA sequence

Calling Function

```
y = btd_rna2dna(x)
```

Arguments

x

String of single-letter codes specifying a RNA sequence

y

DNA sequence

Examples

```
y = btd_rna2dna('AGCUUGGCAAU')
```

```
--> y = btd_rna2dna('AGCUUGGCAAU')
 y  =

 AGCTTGGCAAT
```

See Also

- btd_dna2rna — DNA sequence to RNA sequence

Amino Acids

- btd_aacount — Counts amino acids in sequence
- btd_aadensity — Amino Acid Density
- btd_atomiccomp — CHONS Atomic Composition of an amino acid sequence
- btd_atomicweight — Atomic weight of C,N,O,H,S atoms in an amino acid sequence
- btd_avg_hydrophobicity — Average Hydrophobicity of amino acid sequence
- btd_basecount — Count nucleotides in a amino acid sequence
- btd_cleave — Cleaves amino acid sequence with enzyme
- btd_entropyamino — Entropy of Amino Acid
- btd_ext_coefficients — Extinction coefficients of aminoacid sequence
- btd_exterior_volume — Exterior Volume of amino acid sequence
- btd_interior_volume — Interior Volume of aminoacid sequence
- btd_lcdensity — Linear Charge Density of an aminoacid sequence
- btd_molarabsortivity — Molar absortivity/absorbance of aminoacid sequence
- btd_molweigth — Computes the molecular Weight of a amino acide sequence
- btd_packingvolume — Volume of Amino Acid Sequence
- btd_proteinradius — Radius of Amino Acid Sequence
- btd_psv — Partial Specific Volume of a amino acid sequence
- btd_volumeratio — Volume Ratio of a amino acid sequence

btd_aacount

Counts amino acids in sequence

Calling Function

```
y = btd_aacount(x)
```

Arguments

x

String of single-letter codes specifying an amino acid sequence

y

Frequency of each aminoacid (column of strings)

Description

`y = btd_aacount(x)` counts the number of all individual aminoacid in the given aminoacid sequence b.

Examples

```
y = btd_aacount("AQIILPNMQRRRSTVHPQAABCDBCEFFGHWWIV")
evstr(part(y,5:$))'  // To get the values as numbers
```

```
--> y = btd_aacount("AQIILPNMQRRRSTVHPQAABCDBCEFFGHWWIV")
 y  =
!A  :  3  !
!B  :  2  !
!C  :  2  !
!D  :  1  !
!E  :  1  !
!F  :  2  !
!G  :  1  !
!H  :  2  !
!I  :  3  !
!K  :  0  !
!L  :  1  !
!M  :  1  !
!N  :  1  !
!P  :  2  !
!Q  :  3  !
!R  :  3  !
!S  :  1  !
!T  :  1  !
!V  :  2  !
!W  :  2  !
!Y  :  0  !
!Z  :  0  !
```

```
--> evstr(part(y,5:$))'
 ans  =
         column 1 to 15
   1.    0.    0.    0.    0.    0.    0.    0.    2.    0.    1.    1.    1.    1.    2.

         column 16 to 22
   2.    0.    0.    0.    0.    0.    0.
```

See Also

- btd_codoncount — Count codons in a nucleotide sequence
- btd_basecount — Count nucleotides in a amino acid sequence

btd_aadensity

Amino Acid Density

Calling Function

```
y = btd_aadensity(x)
```

Arguments

x

String of single-letter codes specifying an aminoacid sequence

y

Bar Graph representing density of aminoacids

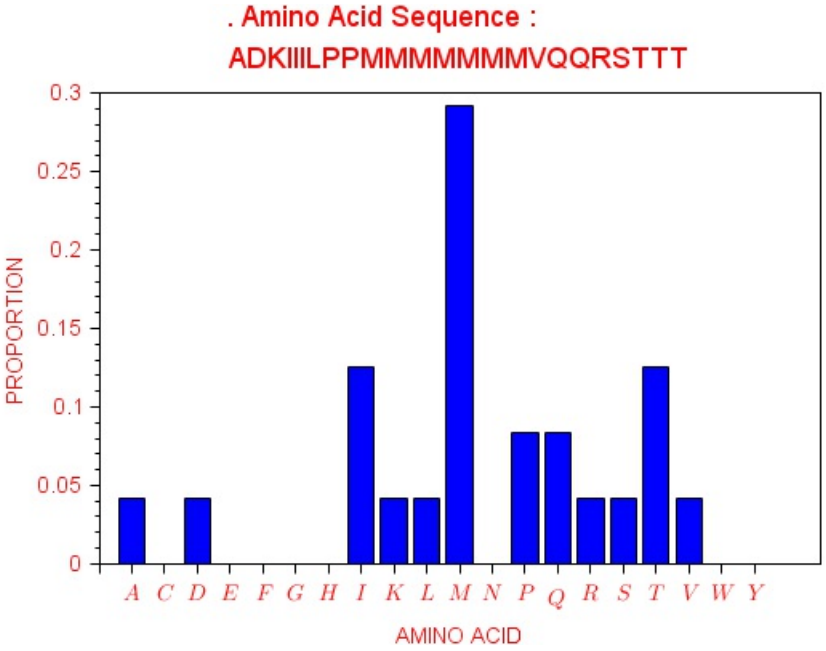
Description

Computes and returns a column vector of decimal numbers as the density of amino acids in the given sequence. The densities are plotted as a bar graph. plot the density of aminoacids along the given sequence x.and display its proportion.

Examples

```
y = btd_aadensity("ADKIIILPPMMMMMMMVQQRSTTT")
```

```
--> y = btd_aadensity("ADKIIILPPMMMMMMMVQQRSTTT")
 y  =
   0.0416667
   0.
   0.0416667
   0.
   0.
   0.
   0.
   0.125
   0.0416667
   0.0416667
   0.2916667
   0.
   0.0833333
   0.0833333
   0.0416667
   0.0416667
   0.125
   0.0416667
   0.
   0.
```



See Also

- btd_ntdensity — Nucleotide Density

btd_atomiccomp

CHONS Atomic Composition of an amino acid sequence

Calling Function

```
y = btd_atomiccomp(x)
```

Arguments

x

Amino acid sequence (text)

y

Type and number of atoms (C, H, N, O, and S) in an amino acid sequence, as a (Nx2) matrix of text.

Description

`atomiccomp(y)` calculates the atomic composition such as number of C,N,O,H,S atoms in the given amino acid sequence b.

Examples

```
y = btd_atomiccomp("AHILLLCKQRSSPA")
evstr(y(:,2))'
```

```
--> y = btd_atomiccomp("AHILLLCKQRSSPA")
 y  =
!H_atoms                117  !
!N_atoms                21   !
!O_atoms                18   !
!C_atoms                67   !
!S_atoms                1    !
!Total number of atoms  224  !

--> evstr(y(:,2))'
 ans  =
   117.   21.   18.   67.   1.   224.
```

See Also

- btd_basecomp — Frequency of indicidual nucleotide

btd_atomicweight

Atomic weight of C,N,O,H,S atoms in an amino acid sequence

Calling Function

```
y = btd_atomicweight(x)
```

Arguments

x

String of single-letter codes specifying a amino acid sequence

y

Atomic weight of amino acids as a (N,3) matrix of strings.

Examples

```
y = btd_atomicweight('ACDEFGHIKLMNPQRSTVWY')
evstr(y(:,2))
```

```
--> y = btd_atomicweight('ACDEFGHIKLMNPQRSTVWY')
 y  =
!atomic weight :    2395.742  amu  !
!number of atoms :  327            !

--> evstr(y(:,2))
 ans  =
   2395.742
   327.
```

See Also

- btd_molweight — Computes the molecular Weight of a amino acide sequence

btd_avg_hydrophobicity

Average Hydrophobicity of amino acid sequence

Calling Function

```
y = btd_avg_hydrophobicity(x)
```

Arguments

x

String of single-letter codes specifying a amino acid sequence

y

Average hydrophobicity of amino acid sequence (decimal number)

Description

Grand Average Hydrophobicity value of aminoacid sequence based on Kyte-Doolittle hydropathy values

Examples

```
y = btd_avg_hydrophobicity('ACDCEFGHIKLMNPQRSTVWY')
```

```
--> y = btd_avg_hydrophobicity('ACDCEFGHIKLMNPQRSTVWY')
 y  =

  -0.3476190
```

See Also

- btd_molarabsortivity — Molar absortivity/absorbance of aminoacid sequence

btd_basecount

Count nucleotides in a amino acid sequence

Calling Function

```
y = btd_basecount(x)
```

Arguments

x

String of codes specifying a nucleotide sequence

y

A, T, G, C with their counts (column of text)

Description

`y = btd_basecount(x)` counts the number of all individual aminoacid in the given aminoacid sequence x.

Examples

```
y = btd_basecount("AATGCTTGGAA")
```

```
--> y = btd_basecount("AATGCTTGGAA")
 y  =
!A  :  4  !
!T  :  3  !
!G  :  3  !
!C  :  1  !
```

See Also

- btd_aacount — Counts amino acids in sequence

btd_cleave

Cleaves amino acid sequence with enzyme

Calling Function

```
y = btd_cleave(x1, x2)
```

Arguments

x1

String of single-letter codes specifying an amino acid sequence.

x2

Enzyme

y

Fragments of aminoacid sequence from cleaveage

Description

To cleave aminoacid sequence at specific sites using enzymes.

`y = btd_cleave(x1, x2)` cuts x1, an aminoacid sequence, in to fragments at the cleavage sites specific for x2(Enzyme),a string specifying a name or abbreviation code for an enzyme. It returns ouput as a cell array of strings representing the fragments from the cleavage.

Examples

```
y = btd_cleave("ADIIILLLAKLKKKRPAIMMM", "Glu C")
```

```
--> y = btd_cleave("ADIIILLLAKLKKKRPAIMMM", "Glu C")
 y  =

!A                  !
!IILLLAKLKKKRPAIMMM  !
```

See Also

- btd_aa2nt — Amino Acid Sequence to Nucleotide Sequence
- btd_nt2aa — Nucleotide Sequence to Amino Acid

btd_entropyamino

Entropy of Amino Acid

Calling Function

```
y = btd_entropyamino(x)
```

Arguments

x

String of single-letter codes specifying an amino acid sequence

y

Entropy value of amino acid sequence (decimal number)

Description

The statistical parameter Entropy of given aminoacid sequence

Examples

```
y = btd_entropyamino("ACDEFGHIKLMNPQRSTVWY")
```

```
--> y = btd_entropyamino("ACDEFGHIKLMNPQRSTVWY")
 y  =
   4.3219281
```

See Also

- btd_entropydna — Entropy of DNA

btd_ext_coefficients

Extinction coefficients of aminoacid sequence

Calling Function

```
y = btd_ext_coefficients(x)
```

Arguments

x

String of single-letter codes specifying a amino acid sequence

y

Extinction Coefficients of amino acid sequence, at the 280 nm wavelength, in $m^{-1}cm^{-1}$ (row vector of 3 strings). The coefficient's value is in y(2). `evstr(y(2))` gives it as decimal number.

Description

Extinction Coefficients of aminoacid sequence, for proteins in water measured at 280 nm.

Examples

```
y = btd_ext_coefficients('ACDEFGHIKLMNPQRSTVWY')
evstr(y(2))
```

```
--> y = btd_ext_coefficients('ACDEFGHIKLMNPQRSTVWY')
 y  =
!Extinction Coefficients:  699  M-1·cm-1  !

--> evstr(y(2))
 ans  =
   699.
```

See Also

- btd_molarabsortivity — Molar absortivity/absorbance of aminoacid sequence

btd_exterior_volume

Exterior Volume of amino acid sequence

Calling Function

```
y = btd_exterior_volume(x)
```

Arguments

x

String of single-letter codes specifying a amino acid sequence

y

Exterior volume of amino acid sequence, in [Å^3](row vector of 3 strings). The volume's value is in y(2). `evstr(y(2))` gives it as decimal number.

Examples

```
y = btd_exterior_volume('ACDEFGHIKLMNPQRSTVWY')
```

```
--> y = btd_exterior_volume('ACDEFGHIKLMNPQRSTVWY')
 y  =
!Approximate Exterior Volume:  2066.501  Å^3  !

--> evstr(y(2))
 ans  =
   2066.501
```

See Also

- btd_packingvolume — Volume of Amino Acid Sequence
- btd_interior_volume — Interior Volume of aminoacid sequence
- btd_volumeratio — Volume Ratio of a amino acid sequence

btd_interior_volume

Interior Volume of aminoacid sequence

Calling Function

```
y = btd_interior_volume(x)
```

Arguments

x

String of single-letter codes specifying a amino acid sequence

y

Interior volume of amino acid sequence, in [Å^3] (row vector of 3 strings). The volume's value is in y(2). `evstr(y(2))` gives it as decimal number.

Examples

```
y = btd_interior_volume('ACDEFGHIKLMNPQRSTVWY')
evstr(y(2))
```

```
--> y = btd_interior_volume('ACDEFGHIKLMNPQRSTVWY')
 y  =
!Approximate Interior Volume:  772.399  Å^3  !

--> evstr(y(2))
 ans  =
   772.399
```

See Also

- btd_packingvolume — Volume of Amino Acid Sequence
- btd_exterior_volume — Exterior Volume of amino acid sequence
- btd_volumeratio — Volume Ratio of a amino acid sequence

btd_lcdensity

Linear Charge Density of an aminoacid sequence

Calling Function

```
y = btd_lcdensity(x)
```

Arguments

x

String of single-letter codes specifying a amino acid sequence

y

Linear Charge Density of amino acid sequence (decimal number)

Examples

```
y = btd_lcdensity('ACDEFGHIKLMNPQRSTVWY')
```

```
--> y = btd_lcdensity('ACDEFGHIKLMNPQRSTVWY')
 y  =
   0.35
```

See Also

- btd_isoelectricpoint — Isoelectric Point

btd_molarabsortivity

Molar absortivity/absorbance of aminoacid sequence

Calling Function

```
y = btd_molarabsortivity(x)
```

Arguments

x

String of single-letter codes specifying an amino acid sequence

y

Molar absortivity of amino acid sequence (decimal number).

Examples

```
y = btd_molarabsortivity('ACDEFGHIKLMNPQRSTVWY')
```

```
--> y = btd_molarabsortivity('ACDEFGHIKLMNPQRSTVWY')
 y  =
   2.9176971
```

See Also

- btd_ext_coefficients — Extinction coefficients of aminoacid sequence

btd_molweigth

Computes the molecular Weight of a amino acide sequence

Calling Function

```
y = btd_molweight(x)
```

Arguments

x

String of single-letter codes specifying an amino acid sequence.

y

molecular weight of the amino acid sequence, as a string like `"2526.235 g/mol"`

Description

`molweight(x)` calculates the molecular weight for the amino acid sequence x, in g/mol.

Examples

```
y = btd_molweight("ADIIILLLAKLKKKRPAIMMM")
```

```
--> y = btd_molweight("ADIIILLLAKLKKKRPAIMMM")
 y  =
 2411.15 g/mol
```

See Also

- btd_molweight — Computes the molecular Weight of a amino acide sequence

btd_packingvolume

Volume of Amino Acid Sequence

Calling Function

```
y = btd_packingvolume(x)
```

Arguments

x

String of single-letter codes specifying a amino acid sequence

y

Volume of amino acid sequence (row vector of 3 strings). The radius value is in y(2). `evstr(y(2))` gives it as decimal number.

Description

Approximate Volume of aminoacid sequence and its unit is Å^3.

Examples

```
y = btd_packingvolume('ACDEFGHIKLMNPQRSTVWY')
evstr(y(2))
```

```
--> y = btd_packingvolume('ACDEFGHIKLMNPQRSTVWY')
 y  =
!Approximate Packing Volume:  2838.9  Å^3  !

--> evstr(y(2))
 ans  =
   2838.9
```

See Also

- btd_interior_volume — Interior Volume of aminoacid sequence
- btd_exterior_volume — Exterior Volume of amino acid sequence
- btd_volumeratio — Volume Ratio of a amino acid sequence

btd_proteinradius

Radius of Amino Acid Sequence

Calling Function

```
y = btd_proteinradius(x)
```

Arguments

x

String of single-letter codes specifying a amino acid sequence

y

Radius of amino acid sequence, in Å (decimal number).

Examples

```
y = btd_proteinradius('ACDEFGHIKLMNPQRSTVWY')
```

```
--> y = btd_proteinradius('ACDEFGHIKLMNPQRSTVWY')
 y  =

  10.50787
```

See Also

- btd_packingvolume — Volume of Amino Acid Sequence

btd_psv

Partial Specific Volume of a amino acid sequence

Calling Function

```
y = btd_psv(x)
```

Arguments

x

String of single-letter codes specifying a amino acid sequence

y

Partial specific volume of amino acid sequence, and its [ml/g] unit (as a row vector of 3 strings). The value can be retrieved as a decimal number with `evstr(y(2))`.

Examples

```
y = btd_psv('ACDEFGHIKLMNPQRSTVWY')
evstr(y(2))
```

```
--> y = btd_psv('ACDEFGHIKLMNPQRSTVWY')
 y  =
!Approximate Partial Specific Volume:  1.1849858  ml/g  !

--> evstr(y(2))
 ans  =
   1.1849858
```

See Also

- btd_interior_volume — Interior Volume of aminoacid sequence
- btd_exterior_volume — Exterior Volume of amino acid sequence

btd_volumeratio

Volume Ratio of a amino acid sequence

Calling Function

```
y = btd_volumeratio(x)
```

Arguments

x

String of single-letter codes specifying an amino acid sequence

y

Volume ratio of amino acid sequence (decimal number).

Description

Approximate Volume ratio between Interior Volume and Exterior Volume of aminoacid sequence and its unit is [$Å^3$]

Examples

```
y = btd_volumeratio('ACDEFGHIKLMNPQRSTVWY')
```

```
--> y = btd_volumeratio('ACDEFGHIKLMNPQRSTVWY')
 y  =

   2.675432
```

See Also

- btd_interior_volume — Interior Volume of aminoacid sequence
- btd_exterior_volume — Exterior Volume of amino acid sequence
- btd_packingvolume — Volume of Amino Acid Sequence

BTD

- BTD
  - btd.nwalign — Needleman-Wunsch Global alignment of two nucleotide sequences
  - btd.swalign — Smith Waterman Local alignment of two nucleotide sequences
  - btd_CpG — CpG Islands (counts 'GC' dinucleotides in a DNA sequence)
  - btd_DNAword — Number and positions of a DNA pattern in a DNA sequence
  - btd_UpA — UpA Islands : counts 'AU'/'AT' dinucleotides in a DNA/RNA sequence
  - btd_at2gcratio — AT/GC Ratio
  - btd_atcontent — AT content of a DNA Sequence
  - btd_basecomp — Frequency of indicidual nucleotide
  - btd_codoncount — Count codons in a nucleotide sequence
  - btd_codonusage — Frequency of each codon type
  - btd_entropydna — Entropy of DNA
  - btd_gccontent — Computes the GC content of a DNA sequence
  - btd_isoelectricpoint — Isoelectric Point
  - btd_kmer — K-mer counts of number of occurences of given DNA word/substring
  - btd_ntdensity — Nucleotide Density
  - btd_purinecomp — Purine Composition of a DNA/RNA sequence
  - btd_pyrimidinecomp — Pyrimidine Composition
  - btd_randntseq — Generates a random nucleotide sequence from finite alphabet
  - btd_randprotseq — Generates a random protein sequence from finite alphabet
  - btd_rho — Rho
  - btd_seqcomplement — Complementary strand of a nucleotide sequence
  - btd_seqrcomplement — reverse complementary strand of a nucleotide sequence
  - btd_seqshoworfs — Open Reading Frames found in a nucleotide sequence

btd.nwalign

Needleman-Wunsch Global alignment of two nucleotide sequences

Calling Sequence

```
y = btd.nwalign(x1, x2)
```

Arguments

x1

String of single-letter codes specifying a query nucleotide sequence

x2

String of single-letter codes specifying a target nucleotide sequence

y

Alligned sequences

Description

Needleman-Wunsch Global alignment of two nucleotide sequences

y= btd.nwalign(x1, x2) returns a 2-by-N character array showing the two sequences, Query and Target which are aligned using Needleman Wunsch Algorithm. The symbol _ indicates the gap introduced in the query or target..

Examples

```
jimport btd
```

```
y = btd.nwalign("ATGTCGAT","ATGCCAT")
```

See Also

- btd.swalign — Smith Waterman Local alignment of two nucleotide sequences

btd.swalign

Smith Waterman Local alignment of two nucleotide sequences

Calling Sequence

```
y = btd.swalign(x1, x2)
```

Arguments

x1

String of single-letter codes specifying a query nucleotide sequence

x2

String of single-letter codes specifying a target nucleotide sequence

y

Alligned sequences

Description

`y = btd.swalign(x1, x2)` returns a 2-by-N character array showing the two sequences, Query and Target which are aligned using Smith Waterman Algorithm. The symbol _ indicates the gap introduced in the query or target..

Examples

```
jimport btd;
```

```
y = btd.swalign("ATGTCGAT","ATGCCAT")
```

See Also

- btd.nwalign — Needleman-Wunsch Global alignment of two nucleotide sequences

btd_CpG

CpG Islands (counts 'GC' dinucleotides in a DNA sequence)

Calling Function

```
y = btd_CpG(x)
```

Arguments

x

String of single-letter codes specifying a DNA sequence

y

Number of 'GC' dinucleotides (decimal integer)

Description

`y = btd_CpG(x)` count of all dinucleotides for which a Cytosine nucleotide occurs next to a Guanine nucleotide.

Examples

```
y = btd_CpG("ATCGATCGATCGTATATGCTAGCTA")
```

```
--> y = btd_CpG("ATCGATCGATCGTATATGCTAGCTA")
 y  =
   2.
```

See Also

- btd_UpA — UpA Islands : counts 'AU'/'AT' dinucleotides in a DNA/RNA sequence

btd_DNAword

Number and positions of a DNA pattern in a DNA sequence

Calling Function

```
y = btd_DNAword(x1, x2)
```

Arguments

x1

String of single-letter codes specifying a DNA sequence

x2

DNA pattern (string)

y

Number and positions of given DNA word (2x1 column of strings)

Description

`y = btd_DNAword(x1, x2)` counts the number of DNA pattern x2, and locates its position in the given DNA sequence x1.

Examples

```
y = btd_DNAword("ACTCTAGCTACT","ACT")
```

```
--> y = btd_DNAword("ACTCTAGCTACT","ACT")
 y  =
!No of DNA words        :  2     !
!Positions of DNA words :  1 10  !
```

See Also

- btd_basecomp — Frequency of indicidual nucleotide
- btd_atomiccomp — CHONS Atomic Composition of an amino acid sequence

btd_UpA

UpA Islands : counts 'AU'/'AT' dinucleotides in a DNA/RNA sequence

Calling Function

```
y = btd_UpA(x)
```

Arguments

x

String of single-letter codes specifying a DNA/RNA sequence

y

Count of 'AT'/'AU' dinucleotides (decimal integer)

Description

`y = btd_UpA(x)` count of all dinucleotides for which a Uracil/Thymine nucleotide occurs next to a Adenine nucleotide.

Examples

```
y = btd_UpA("ATCGATCGATCGTATATGCTAGCTA")
```

```
--> y = btd_UpA("ATCGATCGATCGTATATGCTAGCTA")
 y  =
   5.
```

See Also

- btd_CpG — CpG Islands (counts 'GC' dinucleotides in a DNA sequence)

btd_at2gcratio

AT/GC Ratio

Calling Function

```
y = btd_at2gcratio(x)
```

Arguments

x

String of single-letter codes specifying a DNA/RNA sequence

y

Ratio of AT/GC (decimal number).

Description

Computes the ratio between AT and GC

Examples

```
y = btd_at2gcratio("ACTCTAGCTACT")
```

```
--> y = btd_at2gcratio("ACTCTAGCTACT")
 y  =
   1.4
```

See Also

- btd_atcontent — AT content of a DNA Sequence
- btd_gccontent — Computes the GC content of a DNA sequence

btd_atcontent

AT content of a DNA Sequence

Calling Function

```
y = btd_atcontent(x)
```

Arguments

x

input DNA sequence (string)

y

holds output AT content (percentage, as a string)

Description

`y = btd_atcontent(b)` calulates the number of A and T in a given DNA sequence b.

Examples

```
y = btd_atcontent("ATCGATCGATCGTATATGCTAGCTA")
```

```
--> y = btd_atcontent("ATCGATCGATCGTATATGCTAGCTA")
 y  =
 60 %
```

See Also

- btd_gccontent — Computes the GC content of a DNA sequence
- btd_CpG — CpG Islands (counts 'GC' dinucleotides in a DNA sequence)

btd_basecomp

Frequency of indicidual nucleotide

Calling Function

```
y = btd_basecomp(x1, x2)
```

Arguments

x1

String of single-letter codes specifying a DNA sequence

x2

Nucleotide Base

y

Frequency of given nucleotide base (positive decimal integer).

Description

Computes frequency of indicidual nucleotide

`y = btd_basecomp(x1, x2)` count the number of nucleotide x2 in the given DNA sequence x1.

Examples

```
y = btd_basecomp("ACTCTAGCTACT","A")
```

```
--> y = btd_basecomp("ACTCTAGCTACT","A")
 y  =
   3.
```

See Also

- btd_atomiccomp — CHONS Atomic Composition of an amino acid sequence

btd_codoncount

Count codons in a nucleotide sequence

Calling Function

```
y = btd_codoncount(x)
```

Arguments

x

String of codes specifying a nucleotide sequence

y

Matrix of size [n,2] of text, one row per identified codon: y(:,1) are condons codes. y(:,2) are respective numbers: they can be converted into decimal numbers with `evstr(y(:,2))`.

Description

To count the number of codon of each type

`y = btd_codoncount(x)` counts the number of all possible three letter codons in the given DNA sequence x.

Examples

```
y = btd_codoncount("AAAAAACTC")
evstr(y(:,2))
```

```
--> y = btd_codoncount("AAAAAACTC")
 y  =
!AAA  2  !
!CTC  1  !

--> evstr(y(:,2))
 ans  =
   2.
   1.
```

See Also

- btd_aacount — Counts amino acids in sequence

btd_codonusage

Frequency of each codon type

Calling Function

```
y = btd_codonusage(x1, x2)
```

Arguments

x1
: String of single-letter codes specifying a DNA/RNA sequence

x2
: integer: Reading Frame (1/2/3/4/5/6)

y
: Frequency of each codon type. Matrix of size [n,2] of text, one row per identified codon: y(:,1) are condons codes. y(:,2) are respective numbers: they can be converted into decimal numbers with `evstr(y(:,2))`

Description

To count the frequency of each codon type

y = btd_codonusage(x1, x2) count the frequency of each codon type in the given nucleotide sequence x1 in a given Reading Frame x2.

Examples

```
y = btd_codonusage("ATCGATCGATCGTATATGCTAGCTA",1)
evstr(y(:,2))'
```

```
--> y = btd_codonusage("ATCGATCGATCGTATATGCTAGCTA",1)
 y  =
!UCG  1  !
!UAU  1  !
!GCU  1  !
!GAU  1  !
!CUA  1  !
!CGA  1  !
!AUG  1  !
!AUC  1  !

--> evstr(y(:,2))'
 ans  =
   1.   1.   1.   1.   1.   1.   1.   1.
```

See Also

- btd_codoncount — Count codons in a nucleotide sequence

btd_entropydna

Entropy of DNA

Calling Function

```
y = btd_entropydna(x)
```

Arguments

x

String of single-letter codes specifying a nucleotide sequence

y

Entropy value of DNA sequence (decimal number)

Description

The statistical parameter Entropy of given DNA sequence

Examples

```
y = btd_entropydna("ACTGATCGAATATGCGAGCGA")
```

```
--> y = btd_entropydna("ACTGATCGAATATGCGAGCGA")
 y  =
   1.956067
```

See Also

- btd_entropyamino — Entropy of Amino Acid

btd_gccontent

Computes the GC content of a DNA sequence

Calling Function

```
y = btd_gccontent(x)
```

Arguments

x

String of single-letter codes specifying a DNA sequence

y

Fraction of G+C bases of given DNA sequence (string)

Description

`y = btd_gccontent(x)` calculates the number of G and C in a given DNA sequence x.

Examples

```
y = btd_gccontent("ATCGATCGATCGTATATGCTAGCTA")
```

```
--> y = btd_gccontent("ATCGATCGATCGTATATGCTAGCTA")
 y  =
 4 %
```

See Also

- btd_atcontent — AT content of a DNA Sequence btd_UpA — UpA Islands : counts 'AU'/'AT' dinucleotides in a DNA/RNA sequence

btd_isoelectricpoint

Isoelectric Point

Calling Function

```
y = btd_isoelectricpoint(x)
```

Arguments

x

String of single-letter codes specifying a amino acid sequence

y

Isoelectricpoint of amino acid sequence (1x2 row of text: y(2) is the result, as a string representing a decimal number).

Description

Isoelectric point represents a pH value where given protein's net charge becomes zero

Examples

```
y = btd_isoelectricpoint('ACDEFGHIKLMNPQRSTVWY')
evstr(y(2))
```

```
--> y = btd_isoelectricpoint('ACDEFGHIKLMNPQRSTVWY')
 y  =
!Approximate pI value:  6.892  !

--> evstr(y(2))
 ans  =
   6.892
```

See Also

- btd_lcdensity — Linear Charge Density of an aminoacid sequence
- strtod

btd_kmer

K-mer counts of number of occurences of given DNA word/substring

Calling Function

```
y = btd_kmer(x1, x2)
```

Arguments

x1

String of single-letter codes specifying a DNA sequence

x2

DNA word/substring length.

y

Counts of DNA word/substring (Nx2 matrix of text: y(:1) are words names; y(:,2) are respective count numbers).

Description

`y = btd_kmer(x1, x2)` count the number of occurence of a particular substring that are 1,2,3...n-1 nucleotides long in a given DNA sequence b.

Examples

```
y = btd_kmer("ATCGATCGATCGTATATGCTAGCTA", 2)
evstr(y(:,2))'
```

```
--> y = btd_kmer("ATCGATCGATCGTATATGCTAGCTA",2)
 y  =
!CG  3  !
!AT  3  !
!TA  2  !
!CT  2  !
!TG  1  !
!AG  1  !

--> evstr(y(:,2))'
 ans  =
   3.   3.   2.   2.   1.   1.
```

See Also

- btd_rho — Rho

btd_ntdensity

Nucleotide Density

Calling Function

```
y = btd_ntdensity(x)
```

Arguments

x
: String of single-letter codes specifying a nucleotide sequence

y
: column vector of densities of nucleotides (decimal numbers)

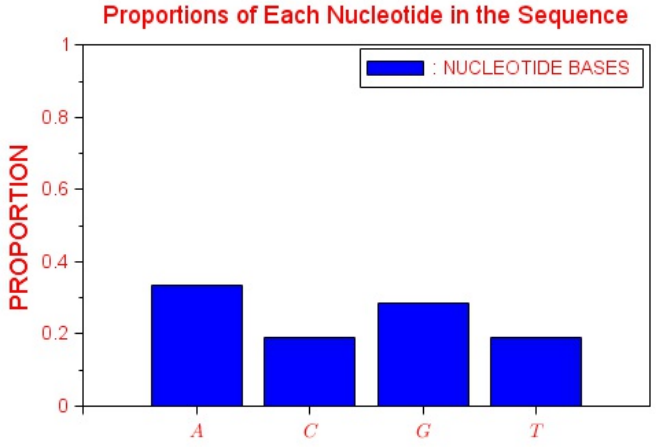
Description

computes, and returns the density of nucleotides along the given sequence x, and plots them as a bar graph.

Examples

```
y = btd_ntdensity("ACTGATCGAATATGCGAGCGA")
```

```
--> y = btd_ntdensity("ACTGATCGAATATGCGAGCGA")
 y  =
   0.3333333
   0.1904762
   0.2857143
   0.1904762
```



See Also

- btd_aadensity — Amino Acid Density

btd_purinecomp

Purine Composition of a DNA/RNA sequence

Calling Function

```
y = btd_purinecomp(x)
```

Arguments

x

String of single-letter codes specifying a DNA/RNA sequence

y

Percentage composition of adenine and guanine bases (as a string)

Description

`y = btd_purinecomp(x)` calulates composition of a purine in DNA sequence,that is number of A and G.

Examples

```
y = btd_purinecomp("ATCGATCGATCGTATATGCTAGCTA")
```

```
--> y = btd_purinecomp("ATCGATCGATCGTATATGCTAGCTA")
 y  =

 48 %
```

See Also

- btd_pyrimidinecomp — Pyrimidine Composition

btd_pyrimidinecomp

Pyrimidine Composition

Calling Function

```
y = btd_pyrimidinecomp(x)
```

Arguments

x

String of single-letter codes specifying a DNA/RNA sequence

y

Percentage composition of cytosine and thymine bases (as a string)

Description

To calculate percentage composition of cytosine and thymine bases

`y = btd_pyrimidinecomp(x)` calulates composition of a pyrimidine in DNA sequence,that is number of T and C.

Examples

```
y = btd_pyrimidinecomp("ATCGATCGATCGTATATGCTAGCTA")
```

```
--> y = btd_pyrimidinecomp("ATCGATCGATCGTATATGCTAGCTA")
 y  =

 52 %
```

See Also

- btd_purinecomp — Purine Composition of a DNA/RNA sequence
- btd_atcontent — AT content of a DNA Sequence

btd_randntseq

Generates a random nucleotide sequence from finite alphabet

Calling Function

```
y = btd_randntseq(x)
```

Arguments

x

Integer that specifies the number of nucleotides in the random sequence

y

A random nucleotide sequence with a length specified x

Examples

```
y = btd_randntseq(10)
```

```
--> y = btd_randntseq(10)
 y  =

 TCGTGGTTGC
```

See Also

- btd_randprotseq — Generates a random protein sequence from finite alphabet

btd_randprotseq

Generates a random protein sequence from finite alphabet

Calling Function

```
y = btd_randprotseq(x)
```

Arguments

x

Integer that specifies the number of proteins in the random sequence.

y

A random protein sequence with a length specified x.

Examples

```
y = btd_randprotseq(10)
```

```
--> y = btd_randprotseq(10)
 y  =

 VRGMVFRYSD
```

See Also

- btd_randntseq — Generates a random nucleotide sequence from finite alphabet

btd_rho

Rho

Calling Function

```
y = btd_rho(x1, x2)
```

Arguments

x1

String of single-letter codes specifying a DNA sequence

x2

DNA word/substring

y

Proportion of DNA word (decimal number in [0,1])

Description

To measure the proportion of a particular DNA word against the product of frequencies of each base that corresponds to the given DNA word.

`y = btd_rho(x1, x2)` calcualte the proportion of a c against the product of frequencies of each base that correspondes to the DNA word,x2. in a given DNA sequence x1.

Examples

```
y = btd_rho("ATACCGATA","TA")
```

```
--> y = btd_rho("ATACCGATA", "TA")
 y  =

   0.25
```

See Also

- btd_kmer — K-mer counts of number of occurences of given DNA word/substring

btd_seqcomplement

Complementary strand of a nucleotide sequence

Calling Function

```
y = btd_seqcomplement(x)
```

Arguments

x

String of single-letter codes specifying a nucleotide sequence

y

Complementary strand of a DNA or RNA nucleotide sequence.

Description

`y = btd_seqcomplement(x)` returns array string of complementary strand of a given DNA sequence x.

Examples

```
y = btd_seqcomplement("ATGC")
```

```
--> y = btd_seqcomplement("ATGC")
 y  =

 TACG
```

See Also

- btd_seqrcomplement — reverse complementary strand of a nucleotide sequence

btd_seqrcomplement

reverse complementary strand of a nucleotide sequence

Calling Function

```
y = btd_seqrcomplement(x)
```

Arguments

x

String of single-letter codes specifying a nucleotide sequence

y

Reverse complementary strand of a DNA or RNA nucleotide sequence

Description

`y = btd_seqrcomplement(x)` returns array string of reverse complementary strand of a given DNA sequence b.

Examples

```
y = btd_seqrcomplement("ATGC")
```

```
--> y = btd_seqrcomplement("ATGC")
 y  =
 GCAT
```

See Also

- btd_seqcomplement — Complementary strand of a nucleotide sequence

btd_seqshoworfs

Open Reading Frames found in a nucleotide sequence

Calling Function

```
y = btd_seqshoworfs(x)
```

Arguments

y

String of single-letter codes specifying a nucleotide sequence

x

All open reading frames using the standard or an alternative genetic code (column of text: one frame per row).

Examples

```
nuSeq = 'TTTTTATTCTTTCTGCCAGGTACATCAGATCCATCAGGTCCGAGCTGTGTTGA'+..
        'CTACCACTGCTTTTCCCTTCGTCTCAGTTATGTCTTGGAAGAAGGCTTT'+..
        'GCGGATCCCCGGAGACCTTCGGGTAGCAACTGTCACCTTGATGCTGGCGATGCTGAGC';
y = btd_seqshoworfs(nuSeq)
// To get only the frames:
stripblanks(part(y,15:$))
```

```
--> y= btd_seqshoworfs(nuSeq)
 y  =
!Frame 1 :       FLFFLPGTSDPSGPSCVDYHCFSLRLSYVLEEGFADPRRPSGSNCHLDAGDAE   !
!Frame 2 :       FYSFCQVHQIHQVRAVLTTTAFPFVSV*SWKKALRIPGDLRVATVTL*LA*LS   !
!Frame 3 :       FILSARYIRSIRSELC_LPLLFPSSQLCLGRRLCGSPETFG_QLSP_CWRC_    !
!Frame 4 :       RVVAVVVPLST*GFQRPLGVSEEGSVLTLLPFSSPSVVSSLDYLDY*DRLSYF   !
!Frame 5 :       ES_RS_FHCQRWASRGP_AFRKKVLY_LCFPFRHHQLCRAWTT_TTWTVFLIF   !
!Frame 6 :       SRSGRSSTVNDGLPEAPRRFGRRFCIDSASLFVTISCVEPGLPRLHGPSFLF    !

--> // To get only the frames:
--> stripblanks(part(y,15:$))
 ans  =
!FLFFLPGTSDPSGPSCVDYHCFSLRLSYVLEEGFADPRRPSGSNCHLDAGDAE  !
!FYSFCQVHQIHQVRAVLTTTAFPFVSV*SWKKALRIPGDLRVATVTL*LA*LS  !
!FILSARYIRSIRSELC_LPLLFPSSQLCLGRRLCGSPETFG_QLSP_CWRC_   !
!RVVAVVVPLST*GFQRPLGVSEEGSVLTLLPFSSPSVVSSLDYLDY*DRLSYF  !
!ES_RS_FHCQRWASRGP_AFRKKVLY_LCFPFRHHQLCRAWTT_TTWTVFLIF  !
!SRSGRSSTVNDGLPEAPRRFGRRFCIDSASLFVTISCVEPGLPRLHGPSFLF   !
```

Data bases

- btd_getembl — Gets sequence information from the online EMBL database
- btd_getgenbank — Gets sequence information from the online GenBank database
- btd_getgenpept — Retrieve sequence information from GenPept database
- btd_getpdb — Retrieve protein structure data from Protein Data Bank (PDB) database
- btd_getuniprot — Retrieve sequence information from UniProt database