Package ‘rmelting’

April 2, 2024

Title R Interface to MELTING 5

Version 1.18.0

Description R interface to the MELTING 5 program (https://www.ebi.ac.uk/biomodels/tools/melting/) to compute melting temperatures of nucleic acid duplexes along with other thermodynamic parameters.

Depends R (>= 3.6)

Imports Rdpack, rJava (>= 0.9-8)

Suggests readxl, knitr, rmarkdown, reshape2, pander, testthat

SystemRequirements Java

biocViews BiomedicalInformatics, Cheminformatics,

License GPL-2 | GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.2.1

RdMacros Rdpack

URL https://github.com/aravind-j/rmelting,
     https://aravind-j.github.io/rmelting/

BugReports https://github.com/aravind-j/rmelting/issues

VignetteBuilder knitr

git_url https://git.bioconductor.org/packages/rmelting

git_branch RELEASE_3_18

git_last_commit 0c3fef2

git_last_commit_date 2023-10-24

Repository Bioconductor 3.18

Date/Publication 2024-04-01
Author J. Aravind [aut, cre] (<https://orcid.org/0000-0002-4791-442X>), G. K. Krishna [aut],
Bob Rudis [ctb] (melting5jars),
Nicolas Le Novère [ctb] (MELTING 5 Java Library),
Marine Dumousseau [ctb] (MELTING 5 Java Library),
William John Gowers [ctb] (MELTING 5 Java Library)
Maintainer J. Aravind <j.aravind@icar.gov.in>

R topics documented:

- melting ................................................. 2
- meltingBatch ........................................... 19
- print.melting ........................................... 20
- withWE ................................................. 21

Index 22

melting

Compute melting temperature of a nucleic acid duplex

Description

Compute the enthalpy and entropy of helix-coil transition, and then the melting temperature of a nucleic acid duplex with the MELTING 5 software (Le Novère, 2001; Dumousseau et al., 2012).

Usage

melting(sequence, comp.sequence = NULL,
nucleic.acid.conc,
hybridisation.type = c("dnadna", "rnarna", "dnarna",
  "rnadna", "mrnarna", "rnamrna"),
Na.conc, Mg.conc, Tris.conc, K.conc,
dNTP.conc, DMSO.conc, formamide.conc,
size.threshold = 60, force.self = FALSE, correction.factor,
method.approx = c("ahs91", "che93", "che93corr",
  "schdot", "owe69", "san98",
  "wetdna91", "wetrna91", "wetdnarna91"),
method.nn = c("al197", "bre86", "san04", "san96", "sug96",
  "tan04", "fre86", "xia98", "sug95", "tur06"),
method.GU = c("tur99", "ser12"),
method.singleMM = c("allsanpey", "tur06", "zno07", "zno08", "wat11"),
method.tandemMM = c("allsanpey", "tur99"),
method.single.dangle = c("bom00", "sugdna02", "sugrna02", "ser08"),
method.double.dangle = c("sugdna02", "sugrna02", "ser05", "ser06"),
method.long.dangle = c("sugdna02", "sugrna02"),
method.internal.loop = c("san04", "tur06", "zno07"),
method.single.bulge.loop = c("tan04", "san04", "ser07", "tur06"),
Arguments

**sequence**
Sequence (5’ to 3’) of one strand of the nucleic acid duplex as a character string.

**comp.sequence**
Complementary sequence (3’ to 5’) of the nucleic acid duplex as a character string.

**nucleic.acid.conc**
Concentration of the nucleic acid strand (M or mol L\(^{-1}\)) in excess as a numeric value.

**hybridisation.type**
The hybridisation type. Either "dnadna", "rnarna", "dnarna", "rnadna", "mrnarna" or "rnamrna" (see Hybridisation type options).

**Na.conc**
Concentration of Na ions (M) as a positive numeric value (see Ion and agent concentrations).

**Mg.conc**
Concentration of Mg ions (M) as a positive numeric value (see Ion and agent concentrations).

**Tris.conc**
Concentration of Tris ions (M) as a positive numeric value (see Ion and agent concentrations).

**K.conc**
Concentration of K ions (M) as a positive numeric value (see Ion and agent concentrations).

**dNTP.conc**
Concentration of dNTP (M) as a positive numeric value (see Ion and agent concentrations).

**DMSO.conc**
Concentration of DMSO (%) as a positive numeric value (see Ion and agent concentrations).

**formamide.conc**
Concentration of formamide (M or % depending on correction method) as a positive numeric value (see Ion and agent concentrations).

**size.threshold**
Sequence length threshold to decide approximative or nearest-neighbour approach for computation. Default is 60.
force.self  logical. Enforces that sequence is self complementary and complementary sequence is not required (see Self complementary sequences). Default is FALSE.

correction.factor
Correction factor to be used to modulate the effect of the nucleic acid concentration (nucleic.acid.conc) in the computation of melting temperature (see Correction factor for nucleic acid concentration).

method.approx
Specify the approximative formula to be used for melting temperature calculation for sequences of length greater than size.threshold. Either "ahs01", "che93", "che93corr", "schdot", "owe69", "san98", "wetdna91", "wetrna91" or "wetdnarna91" (see Approximative formulas).

method.nn
Specify the nearest neighbor model to be used for melting temperature calculation for perfectly matching sequences of length lesser than size.threshold. Either "all97", "bre86", "san04", "san96", "sug96", "tan04", "fre86", "xia98", "sug95" or "tur06" (see Perfectly matching sequences).

method.GU
Specify the nearest neighbor model to compute the contribution of GU base pairs to the thermodynamic of helix-coil transition. Either "tur99" or "ser12" (see GU wobble base pairs effect).

method.singleMM
Specify the nearest neighbor model to compute the contribution of single mismatch to the thermodynamic of helix-coil transition. Either "allsanpey", "tur06", "zno07", "zno08" or "wat11" (see Single mismatch effect).

method.tandemMM
Specify the nearest neighbor model to compute the contribution of tandem mismatches to the thermodynamic of helix-coil transition. Either "allsanpey" or "tur99" (see Tandem mismatches effect).

method.single.dangle
Specify the nearest neighbor model to compute the contribution of single dangling end to the thermodynamic of helix-coil transition. Either "bom00", "sugdna02", "sugrna02" or "ser08" (see Single dangling end effect).

method.double.dangle
Specify the nearest neighbor model to compute the contribution of double dangling end to the thermodynamic of helix-coil transition. Either "sugdna02", "sugrna02", "ser05" or "ser06" (see Double dangling end effect).

method.long.dangle
Specify the nearest neighbor model to compute the contribution of long dangling end to the thermodynamic of helix-coil transition. Either "sugdna02" or "sugrna02" (see Long dangling end effect).

method.internal.loop
Specify the nearest neighbor model to compute the contribution of internal loop to the thermodynamic of helix-coil transition. Either "san04", "tur06" or "zno07" (see Internal loop effect).

method.single.bulge.loop
Specify the nearest neighbor model to compute the contribution of single bulge loop to the thermodynamic of helix-coil transition. Either "san04", "tan04", "ser07" or "tur06" (see Single bulge loop effect).
method.long.bulge.loop
Specify the nearest neighbor model to compute the contribution of long bulge loop to the thermodynamic of helix-coil transition. Either "san04" or "tur06" (see Long bulge loop effect).

method.CNG
Specify the nearest neighbor model to compute the contribution of CNG repeats to the thermodynamic of helix-coil transition. Available method is "bro05" (see CNG repeats effect).

method.inosine
Specify the specific nearest neighbor model to compute the contribution of inosine bases (I) to the thermodynamic of helix-coil transition. Either "san05" or "zno07" (see Inosine bases effect).

method.hydroxyadenine
Specify the nearest neighbor model to compute the contribution of hydroxyadenine bases (A*) to the thermodynamic of helix-coil transition. Available method is "sug01" (see Hydroxyadenine bases effect).

method.azobenzenes
Specify the nearest neighbor model to compute the contribution of azobenzenes (X_T for trans azobenzenes and X_C for cis azobenzenes) to the thermodynamic of helix-coil transition. Available method is "asa05" (see Azobenzenes effect).

method.locked
Specify the nearest neighbor model to compute the contribution of single locked nucleic acids (AL, GL, TL and CL) to the thermodynamic of helix-coil transition. Either "owc11" or "mct04" (see Single locked nucleic acids effect).

method.consecutive.locked
Specify the nearest neighbor model to compute the contribution of consecutive locked nucleic acids (AL, GL, TL and CL) to the thermodynamic of helix-coil transition. Available method is "owc11" (see Consecutive locked nucleic acids effect).

method.consecutive.locked.singleMM
Specify the nearest neighbor model to compute the contribution of consecutive locked nucleic acids (AL, GL, TL and CL) with a single mismatch to the thermodynamic of helix-coil transition. Available method is "owc11" (see Consecutive locked nucleic acids with single mismatch effect).

correction.ion
Specify the correction method for ions. Either one of the following:
- Na corrections "ahs01", "kam71", "owc1904", "owc2004", "owc2104", "owc2204", "san96", "san04", "schlif", "tanna06", "wetdna91", "tanna07", "wetrna91" or "wetdnarna91" (see Sodium corrections)
- Mg corrections "owcmg08", "tanmg06" or "tanmg07" (see Magnesium corrections)
- Mixed Na Mg corrections "owcmix08", "tanmix07" or "tanmix07" (see Mixed Sodium and Magnesium corrections)

method.Naeq
Specify the ion correction which gives a sodium equivalent concentration if other cations are present. Either "ahs01", "mit96" or "pey00" (see Sodium equivalent concentration methods).

correction.DMSO
Specify the correction method for DMSO. Specify the correction method for DMSO. Either "ahs01", "mus81", "cul76" or "esc80" (see DMSO corrections).
correction.formamide

Specify the correction method for formamide. Specify the correction method for formamide Either "bla96" or "lincorr" (see Formamide corrections).

Value

A list with the following components:

Environment
A list with details about the melting temperature computation environment.

Options
A list with details about the options (default or user specified) used for melting temperature computation.

Results
A list with the results of the melting temperature computation including the enthalpy and entropy in case of nearest neighbour methods.

Message
Error and/or Warning messages, if any.

Mandatory arguments

The following are the arguments which are mandatory for computation.

sequence 5’ to 3’ sequence of one strand of the nucleic acid duplex as a character string. Recognises A, C, G, T, U, I, X_C, X_T, A*, AL, TL, GL and CL. U and T are not considered identical (see Recognized nucleotides).

comp.sequence Mandatory if there are mismatches, inosine(s) or hydroxyadenine(s) between the two strands. If not specified, it is computed as the complement of sequence. Self-complementarity in sequence is detected even though there may be (are) dangling end(s) and comp. sequence is computed (see Self complementary sequences).

nucleic.acid.conc See Correction factor for nucleic acid concentration.

Na.conc, Mg.conc, Tris.conc, K.conc At least one cation (Na, Mg, Tris, K) concentration is mandatory, the other agents(dNTP, DMSO, formamide) are optional (see Ion and agent concentrations).

hybridisation.type See Hybridisation type options.

Recognized nucleotides

<table>
<thead>
<tr>
<th>Code</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Adenine</td>
</tr>
<tr>
<td>C</td>
<td>Cytosine</td>
</tr>
<tr>
<td>G</td>
<td>Guanine</td>
</tr>
<tr>
<td>T</td>
<td>Thymine</td>
</tr>
<tr>
<td>U</td>
<td>Uracil</td>
</tr>
<tr>
<td>I</td>
<td>Inosine</td>
</tr>
<tr>
<td>X_C</td>
<td>Trans azobenzenes</td>
</tr>
<tr>
<td>X_T</td>
<td>Cis azobenzenes</td>
</tr>
<tr>
<td>A*</td>
<td>Hydroxyadenine</td>
</tr>
<tr>
<td>AL</td>
<td>Locked nucleic acid</td>
</tr>
<tr>
<td>TL</td>
<td>&quot;</td>
</tr>
<tr>
<td>GL</td>
<td>&quot;</td>
</tr>
</tbody>
</table>
U and T are not considered identical.

**Hybridisation type options**

The details of the possible options for hybridisation type specified in the argument `hybridisation.type` are as follows:

<table>
<thead>
<tr>
<th>Option</th>
<th>Sequence</th>
<th>Complementary sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>dndna</td>
<td>DNA</td>
<td>DNA</td>
</tr>
<tr>
<td>rnrna</td>
<td>RNA</td>
<td>RNA</td>
</tr>
<tr>
<td>dnrna</td>
<td>DNA</td>
<td>RNA</td>
</tr>
<tr>
<td>rnrna</td>
<td>RNA</td>
<td>DNA</td>
</tr>
<tr>
<td>mnrnrna</td>
<td>2-o-methyl RNA</td>
<td>RNA</td>
</tr>
<tr>
<td>rnmrnrna</td>
<td>RNA</td>
<td>2-o-methyl RNA</td>
</tr>
</tbody>
</table>

This parameter determines the nature of the sequences in the arguments `sequence` and `comp.sequence`.

**Ion and agent concentrations**

Ion concentrations are specified by the arguments `Na.conc`, `Mg.conc`, `Tris.conc` and `K.conc`, while agent concentrations are specified by the arguments `dNTP.conc`, `DMSO.conc` and `formamide.conc`.

These values are used for different correction functions which approximately adjusts for effects of these ions (Na, Mg, Tris, K) and/or agents (dNTP, DMSO, formamide) on on thermodynamic stability of nucleic acid duplexes. Their concentration limits depends on the correction method used. All the concentrations must be in M, except for the DMSO (%) and formamide (% or M depending on the correction method). Note that `[Tris+]` is about half of the total tris buffer concentration.

**Self complementary sequences**

Self complementarity for perfect matching sequences or sequences with dangling ends is detected automatically. However it can be enforced by the argument `force.self = TRUE`.

**Correction factor for nucleic acid concentration**

For self complementary sequences (Auto detected or specified by `force.self`) it is 1. Otherwise it is 4 if the both strands are present in equivalent amount and 1 if one strand is in excess.

**Approximative estimation formulas**

<table>
<thead>
<tr>
<th>Formula</th>
<th>Type</th>
<th>Limits/Remarks</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>ahs01</td>
<td>DNA</td>
<td>No mismatch</td>
<td>von Ahsen et al., 2001</td>
</tr>
<tr>
<td>che93</td>
<td>DNA</td>
<td>No mismatch; Na=0, Mg=0.0015, Tris=0.01, K=0.05</td>
<td>Marmur and Doty, 1962</td>
</tr>
<tr>
<td>che93corr</td>
<td>DNA</td>
<td>No mismatch; Na=0, Mg=0.0015, Tris=0.01, K=0.05</td>
<td>Marmur and Doty, 1962</td>
</tr>
<tr>
<td>schdot</td>
<td>DNA</td>
<td>No mismatch</td>
<td>Wetmur, 1991; Marmur and</td>
</tr>
</tbody>
</table>
Doty, 1962; Chester and Marshak, 1993; Schildkraut and Lifson, 1965; Wahl et al., 1987; Britten et al., 1974; Hall et al., 1980
Owen et al., 1969; Frank-Kamenetskiii, 1971; Blake, 1996; Blake and Delcourt, 1998
SantaLucia, 1998; von Ahsen et al., 2001
Wetmur, 1991
Wetmur, 1991
Wetmur, 1991

* Default formula for computation.

Note that calculation is increasingly incorrect when the length of the duplex decreases. Further, it does not take into account nucleic acid concentration.

Nearest neighbor models

Perfectly matching sequences:

<table>
<thead>
<tr>
<th>Model</th>
<th>Type</th>
<th>Limits/Remarks</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>owe69</td>
<td>DNA</td>
<td>No mismatch</td>
<td></td>
</tr>
<tr>
<td>san98</td>
<td>DNA</td>
<td>No mismatch</td>
<td></td>
</tr>
<tr>
<td>wetdna91*</td>
<td>DNA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wetrna91*</td>
<td>RNA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wetdnarna91*</td>
<td>DNA/RNA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Default model for computation.

GU wobble base pairs effect:

<table>
<thead>
<tr>
<th>Model</th>
<th>Type</th>
<th>Limits/Remarks</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>tur99</td>
<td>RNA</td>
<td></td>
<td>Mathews et al., 1999</td>
</tr>
<tr>
<td>ser12*</td>
<td>RNA</td>
<td></td>
<td>Chen et al., 2012</td>
</tr>
</tbody>
</table>
* Default model for computation.
GU base pairs are not taken into account by the approximative mode.

**Single mismatch effect:**

<table>
<thead>
<tr>
<th>Model</th>
<th>Type</th>
<th>Limits/Remarks</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>allsanpey*</td>
<td>DNA</td>
<td></td>
<td>Allawi and SantaLucia, 1997; Allawi and SantaLucia, 1998; Peyret et al., 1999</td>
</tr>
<tr>
<td>wat11*</td>
<td>DNA/RNA</td>
<td></td>
<td>Watkins et al., 2011</td>
</tr>
<tr>
<td>tur06</td>
<td>RNA</td>
<td></td>
<td>Lu et al., 2006</td>
</tr>
<tr>
<td>zno07*</td>
<td>RNA</td>
<td>At least one adjacent GU base pair.</td>
<td>Davis and Znosko, 2007</td>
</tr>
<tr>
<td>zno08</td>
<td>RNA</td>
<td></td>
<td>Davis and Znosko, 2008</td>
</tr>
</tbody>
</table>

* Default model for computation.
Single mismatches are not taken into account by the approximative mode.

**Tandem mismatches effect:**

<table>
<thead>
<tr>
<th>Model</th>
<th>Type</th>
<th>Limits/Remarks</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>allsanpey*</td>
<td>DNA</td>
<td>Only GT mismatches and TA/TG mismatches.</td>
<td>Allawi and SantaLucia, 1997; Allawi and SantaLucia, 1998; Peyret et al., 1999</td>
</tr>
<tr>
<td>tur99*</td>
<td>RNA</td>
<td>No adjacent GU or UG base pairs.</td>
<td>Mathews et al., 1999; Lu et al., 2006</td>
</tr>
</tbody>
</table>

* Default model for computation.
Tandem mismatches are not taken into account by the approximative mode. Note that not all the mismatched Crick’s pairs have been investigated.

**Single dangling end effect:**

<table>
<thead>
<tr>
<th>Model</th>
<th>Type</th>
<th>Limits/Remarks</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>bom00*</td>
<td>DNA</td>
<td>Only terminal poly A self complementary sequences.</td>
<td>Bommarito et al., 2000</td>
</tr>
<tr>
<td>sugdna02</td>
<td>DNA</td>
<td>Only terminal poly A self complementary sequences.</td>
<td>Ohmichi et al., 2002</td>
</tr>
<tr>
<td>sugrna02</td>
<td>RNA</td>
<td>Only terminal poly A self complementary sequences.</td>
<td>Ohmichi et al., 2002</td>
</tr>
<tr>
<td>ser08*</td>
<td>RNA</td>
<td>Only 3’ UA, GU and UG terminal base pairs only 5’ UG and GU terminal base pairs.</td>
<td>O’Toole et al., 2006; Miller et al., 2008</td>
</tr>
</tbody>
</table>

* Default model for computation.
Single dangling ends are not taken into account by the approximative mode.
### Double dangling end effect:

<table>
<thead>
<tr>
<th>Model</th>
<th>Type</th>
<th>Limits/Remarks</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>sugdna02*</td>
<td>DNA</td>
<td>Only terminal poly A self complementary sequences.</td>
<td>Ohmichi et al., 2002</td>
</tr>
<tr>
<td>sugrna02</td>
<td>RNA</td>
<td>Only terminal poly A self complementary sequences.</td>
<td>Ohmichi et al., 2002</td>
</tr>
<tr>
<td>ser05</td>
<td>RNA</td>
<td>Depends on the available thermodynamic parameters for single dangling end.</td>
<td>O’Toole et al., 2005</td>
</tr>
<tr>
<td>ser06*</td>
<td>RNA</td>
<td></td>
<td>O’Toole et al., 2006</td>
</tr>
</tbody>
</table>

* Default model for computation.

Double dangling ends are not taken into account by the approximative mode.

### Long dangling end effect:

<table>
<thead>
<tr>
<th>Model</th>
<th>Type</th>
<th>Limits/Remarks</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>sugdna02*</td>
<td>DNA</td>
<td>Only terminal poly A self complementary sequences.</td>
<td>Ohmichi et al., 2002</td>
</tr>
<tr>
<td>sugrna02</td>
<td>RNA</td>
<td>Only terminal poly A self complementary sequences.</td>
<td>Ohmichi et al., 2002</td>
</tr>
</tbody>
</table>

* Default model for computation.

Long dangling ends are not taken into account by the approximative mode.

### Internal loop effect:

<table>
<thead>
<tr>
<th>Model</th>
<th>Type</th>
<th>Limits/Remarks</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>san04*</td>
<td>DNA</td>
<td>Missing asymmetry penalty. Not tested with experimental results.</td>
<td>SantaLucia and Hicks, 2004</td>
</tr>
<tr>
<td>tur06</td>
<td>RNA</td>
<td>Not tested with experimental results.</td>
<td>Lu et al., 2006</td>
</tr>
<tr>
<td>zno07*</td>
<td>RNA</td>
<td>Only for 1x2 loop.</td>
<td>Badhwar et al., 2007</td>
</tr>
</tbody>
</table>

* Default model for computation.

Internal loops are not taken into account by the approximative mode.

### Single bulge loop effect:

<table>
<thead>
<tr>
<th>Model</th>
<th>Type</th>
<th>Limits/Remarks</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>tan04*</td>
<td>DNA</td>
<td>Missing closing AT penalty.</td>
<td>Tan and Chen, 2007</td>
</tr>
<tr>
<td>san04</td>
<td>DNA</td>
<td>Missing closing AT penalty.</td>
<td>SantaLucia and Hicks, 2004</td>
</tr>
<tr>
<td>ser07</td>
<td>RNA</td>
<td>Less reliable results. Some missing parameters.</td>
<td>Blose et al., 2007</td>
</tr>
<tr>
<td>tur06*</td>
<td>RNA</td>
<td></td>
<td>Lu et al., 2006</td>
</tr>
</tbody>
</table>

* Default model for computation.

Single bulge loops are not taken into account by the approximative mode.
Long bulge loop effect:

<table>
<thead>
<tr>
<th>Model</th>
<th>Type</th>
<th>Limits/Remarks</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>san04*</td>
<td>DNA</td>
<td>Missing closing AT penalty.</td>
<td>SantaLucia and Hicks, 2004</td>
</tr>
<tr>
<td>tur06*</td>
<td>RNA</td>
<td>Not tested with experimental results.</td>
<td>Mathews et al., 1999; Lu et al., 2006</td>
</tr>
</tbody>
</table>

* Default model for computation.
Long bulge loops are not taken into account by the approximative mode.

CNG repeats effect:

<table>
<thead>
<tr>
<th>Model</th>
<th>Type</th>
<th>Limits/Remarks</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>bro05*</td>
<td>RNA</td>
<td>Self complementary sequences.</td>
<td>Broda et al., 2005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 to 7 CNG repeats.</td>
<td></td>
</tr>
</tbody>
</table>

* Default model for computation.
CNG repeats are not taken into account by the approximative mode. The contribution of CNG repeats to the thermodynamic of helix-coil transition can be computed only for 2 to 7 CNG repeats. N represents a single mismatch of type N/N.

Inosine bases effect:

<table>
<thead>
<tr>
<th>Model</th>
<th>Type</th>
<th>Limits/Remarks</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>san05*</td>
<td>DNA</td>
<td>Missing parameters for tandem base pairs containing inosine bases.</td>
<td>Watkins and SantaLucia, 2005</td>
</tr>
<tr>
<td>zno07*</td>
<td>RNA</td>
<td>Only IU base pairs.</td>
<td>Wright et al., 2007</td>
</tr>
</tbody>
</table>

* Default model for computation.
Inosine bases (I) are not taken into account by the approximative mode.

Hydroxyadenine bases effect:

<table>
<thead>
<tr>
<th>Model</th>
<th>Type</th>
<th>Limits/Remarks</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>sug01*</td>
<td>DNA</td>
<td>Only 5’ GA<em>C 3’ and 5’ TA</em>A 3’ contexts.</td>
<td>Kawakami et al., 2001</td>
</tr>
</tbody>
</table>

* Default model for computation.
Hydroxyadenine bases (A*) are not taken into account by the approximative mode.

Azobenzenes effect effect:

<table>
<thead>
<tr>
<th>Model</th>
<th>Type</th>
<th>Limits/Remarks</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>asa05*</td>
<td>DNA</td>
<td>Less reliable results when the number of cis azobenzene increases.</td>
<td>Asanuma et al., 2005</td>
</tr>
</tbody>
</table>

* Default model for computation.
Azobenzenes (X_T for trans azobenzenes and X_C for cis azobenzenes) are not taken into account by the approximative mode.
Single locked nucleic acids effect:

<table>
<thead>
<tr>
<th>Model</th>
<th>Type</th>
<th>Limits.Remarks</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>mct04</td>
<td>DNA</td>
<td></td>
<td>McTigue, Peterson, and Kahn, 2004</td>
</tr>
<tr>
<td>owc11*</td>
<td>DNA</td>
<td></td>
<td>Owczarzy, You, Groth, and Tataurov, 2011</td>
</tr>
</tbody>
</table>

* Default model for computation.
Locked nucleic acids (AL, GL, TL and CL) are not taken into account by the approximative mode.

Consecutive locked nucleic acids effect:

<table>
<thead>
<tr>
<th>Model</th>
<th>Type</th>
<th>Limits.Remarks</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>owc11*</td>
<td>DNA</td>
<td></td>
<td>Owczarzy et al., 2011</td>
</tr>
</tbody>
</table>

* Default model for computation.
Locked nucleic acids (AL, GL, TL and CL) are not taken into account by the approximative mode.

Consecutive locked nucleic acids with single mismatch effect:

<table>
<thead>
<tr>
<th>Model</th>
<th>Type</th>
<th>Limits.Remarks</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>owc11*</td>
<td>DNA</td>
<td></td>
<td>Owczarzy et al., 2011</td>
</tr>
</tbody>
</table>

* Default model for computation.
Locked nucleic acids (AL, GL, TL and CL) are not taken into account by the approximative mode.

Ion corrections

Sodium corrections:

<table>
<thead>
<tr>
<th>Correction</th>
<th>Type</th>
<th>Limits.Remarks</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>ahs01</td>
<td>DNA</td>
<td>Na&gt;0.</td>
<td>von Ahsen et al., 2001</td>
</tr>
<tr>
<td>schlif</td>
<td>DNA</td>
<td>Na&gt;=0.07; Na&lt;=0.12.</td>
<td>Schildkraut and Lifson, 1965</td>
</tr>
<tr>
<td>tanna06</td>
<td>DNA</td>
<td>Na&gt;=0.001; Na&lt;=1.</td>
<td>Tan and Chen, 2006</td>
</tr>
<tr>
<td>tanna07</td>
<td>RNA</td>
<td>Na&gt;=0.003; Na&lt;=1.</td>
<td>Tan and Chen, 2007</td>
</tr>
<tr>
<td>wet91</td>
<td>RNA, DNA and RNA/DNA</td>
<td>Na&gt;0.</td>
<td>Wetmur, 1991</td>
</tr>
<tr>
<td>kam71</td>
<td>DNA</td>
<td>Na&gt;0; Na&gt;=0.069; Na&lt;=1.02.</td>
<td>Frank-Kamenetskii, 1971</td>
</tr>
<tr>
<td>marschdot</td>
<td>DNA</td>
<td>Na&gt;=0.069; Na&lt;=1.02.</td>
<td>Marmur and Doty, 1962; Blake and Delcourt, 1998</td>
</tr>
<tr>
<td>owc1904</td>
<td>DNA</td>
<td>Na&gt;0. (equation 19)</td>
<td>Owczarzy et al., 2004</td>
</tr>
<tr>
<td>owc2004</td>
<td>DNA</td>
<td>Na&gt;0. (equation 20)</td>
<td>Owczarzy et al., 2004</td>
</tr>
<tr>
<td>owc2104</td>
<td>DNA</td>
<td>Na&gt;0. (equation 21)</td>
<td>Owczarzy et al., 2004</td>
</tr>
</tbody>
</table>
**Magnesium corrections:**

<table>
<thead>
<tr>
<th>Correction</th>
<th>Type</th>
<th>Limits/Remarks</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>owcmg08*</td>
<td>DNA</td>
<td>Mg&gt;=0.0005; Mg&lt;=0.6.</td>
<td>Owczarzy et al., 2008</td>
</tr>
<tr>
<td>tanmg06</td>
<td>DNA</td>
<td>Mg&gt;=0.0001; Mg&lt;=1; Oligomer length superior to 6 base pairs.</td>
<td>Tan and Chen, 2006</td>
</tr>
<tr>
<td>tanmg07*</td>
<td>RNA</td>
<td>Mg&gt;=0.1; Mg&lt;=0.3.</td>
<td>Tan and Chen, 2007</td>
</tr>
</tbody>
</table>

* Default correction method for computation.

**Mixed Sodium and Magnesium corrections:**

<table>
<thead>
<tr>
<th>Correction</th>
<th>Type</th>
<th>Limits/Remarks</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>owcmix08*</td>
<td>DNA</td>
<td>Mg&gt;=0.0005; Mg&lt;=0.6; Na+K+Tris/2&gt;0.</td>
<td>Owczarzy et al., 2008</td>
</tr>
<tr>
<td>tanmix07</td>
<td>DNA, RNA, or 2’-O-MeRNA/RNA</td>
<td>Mg&gt;=0.1; Mg&lt;=0.3; Na+K+Tris/2&gt;=0.1; Na+K+Tris/2&lt;=0.3.</td>
<td>Tan and Chen, 2007</td>
</tr>
</tbody>
</table>

* Default correction method for computation.

The ion correction by Owczarzy et al. (2008) is used by default according to the 
\[
\frac{[\text{Mg}^2+]}{[\text{Mon}^+]^{0.5}}
\]
ratio, where 
\[
[\text{Mon}^+] = [\text{Na}^+] + [\text{Tris}^+] + [\text{K}^+].
\]

If,

\[
[\text{Mon}^+] = 0 \quad \text{Default sodium correction is used.}
\]

**Ratio < 0.22,** Default sodium correction is used.

**0.22 <= Ratio < 6** Default mixed Na and Mg correction is used.

**Ratio >= 6** Default magnesium correction is used.

Note that [Tris+] is about half of the total tris buffer concentration.

**Sodium equivalent concentration methods:**

<table>
<thead>
<tr>
<th>Correction</th>
<th>Type</th>
<th>Limits/Remarks</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>ahs01*</td>
<td>DNA</td>
<td></td>
<td>von Alsen et al., 2001</td>
</tr>
<tr>
<td>mit96</td>
<td>DNA</td>
<td></td>
<td>Mitsuhashi, 1996</td>
</tr>
<tr>
<td>pey00</td>
<td>DNA</td>
<td></td>
<td>Peyret, 2000</td>
</tr>
</tbody>
</table>
* Default correction method for computation.

For the other types of hybridization, the DNA default correction is used. If there are other cations when an approximative approach is used, a sodium equivalence is automatically computed. In case of nearest neighbor approach, the sodium equivalence will be used only if a sodium correction is specified by the argument correction.ion.

### Denaturing agent corrections

#### DMSO corrections:

<table>
<thead>
<tr>
<th>Correction</th>
<th>Type</th>
<th>Limits/Remarks</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>ahs01*</td>
<td>DNA</td>
<td>Not tested with experimental results.</td>
<td>von Ahsen et al., 2001</td>
</tr>
<tr>
<td>cu176</td>
<td>DNA</td>
<td>Not tested with experimental results.</td>
<td>Cullen and Bick, 1976</td>
</tr>
<tr>
<td>esc80</td>
<td>DNA</td>
<td>Not tested with experimental results.</td>
<td>Escara and Hutton, 1980</td>
</tr>
<tr>
<td>mus81</td>
<td>DNA</td>
<td>Not tested with experimental results.</td>
<td>Musielksi et al., 1981</td>
</tr>
</tbody>
</table>

* Default correction method for computation.

For the other types of hybridization, the DNA default correction is used. If there is DMSO when an approximative approach is used, a DMSO correction is automatically computed. In case of nearest neighbor approach and approximative approach, the DMSO correction will be used only if a sodium correction is specified by the argument correction.ion.

#### Formamide corrections:

<table>
<thead>
<tr>
<th>Correction</th>
<th>Type</th>
<th>Limits/Remarks</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>bla96*</td>
<td>DNA</td>
<td>With formamide concentration in mol/L.</td>
<td>Blake, 1996</td>
</tr>
<tr>
<td>lincorr</td>
<td>DNA</td>
<td>With a formamide volume.</td>
<td>McConaughy et al., 1969; Record, 1967; Casey and Davidson, 1977; Hutton, 1977</td>
</tr>
</tbody>
</table>

* Default correction method for computation.

For the other types of hybridization, the DNA default correction is used. If there is formamide when an approximative approach is used, a formamide correction is automatically computed. In case of nearest neighbor approach and approximative approach, the formamide correction will be used only if a sodium correction is specified by the argument correction.ion.

### References


See Also

For more details about algorithm, formulae and methods, see the documentation for MELTING 5.

Examples

```r
# Basic usage
melting(sequence = "CAGTGACAGCAATGGTCG", nucleic.acid.conc = 2e-06, 
          hybridisation.type = "dnadna", Na.conc = 1)

# For more detailed examples refer the vignette.
## Not run:
browseVignettes(package = 'rmelting')

## End(Not run)
```
meltingBatch

Compute melting temperature of multiple nucleic acid duplexes in batch

Description
Compute the enthalpy and entropy of helix-coil transition, and then the melting temperature of multiple nucleic acid duplexes in batch.

Usage
meltingBatch(
  sequence,
  comp.sequence = NULL,
  environment.out = TRUE,
  options.out = TRUE,
  message.out = TRUE,
  ...
)

Arguments
sequence         A character vector of 5' to 3' sequences of one strand of the nucleic acid duplex
                  (Note: Uridine and thymidine are not considered as identical).
comp.sequence     A character vector of 3' to 5' complementary sequences of the nucleic acid duplex. Complementary sequences are computed by default, but need to be specified in case of mismatches, inosine(s) or hydroxyadenine(s) between the two strands.
environment.out   logical. If TRUE, gives the melting temperature computation environment details in the output. Default is TRUE.
options.out       logical. If TRUE, gives the details about the options (default or user specified) used for melting temperature computation in the output. Default is TRUE.
message.out       logical. If TRUE, gives the error and/or warning messages, if any in the output. Default is TRUE.
...               Arguments for melting temperature computation (See melting).

Value
A data frame of the melting temperature computation results along with the details of environment, options and messages if specified by the arguments environment.out, options.out and message.out respectively.

See Also
melting
Examples

```r
sequence <- c("CAAAAAG", "CAAAAAAG", "TTTTATAATAAA", "CCATCGCTACC",
                "CAACAAAAG", "CCATTGCTACC", "CAAAAAAAG", "GTGAAC", "AAAAAACA",
                "CAACTTGATATATA", "CAAAATAAAG", "GCCAGC", "GGGACC",
                "CAAGAAAG", "CTGACAAAGTGC", "GCGAAAAGGCC")

meltingBatch(sequence, nucleic.acid.conc = 0.0004,
              hybridisation.type = "dnadna", Na.conc = 1)

seq <- c("GCAUACG", "CAGUAGGUC", "GCUCUCGC", "GAGUGGAG", "GACAGGCUG",
         "CAGUACGUC", "GACAUCCUG", "GACCACCCUG", "GACAAUGCUG", "GCCGUCGC",
         "CGUCGGCG", "GACUCUGCC", "CAGCUGGUC", "GACUAGCGUC", "CUCUGUCUC",
         "GCGUCCGG", "GUCCGCCG", "GAUCCAC", "GACUACCUG", "GACUAUCUG")

comp.seq <- c("CGUUUGC", "GUCGGCCAG", "GCGUGGCG", "CUCUUUCUC", "CUGUCCGAC",
               "GUCGGCCAG", "CUGUUGGAC", "GUCUUGGAC", "GUGUGGCGAG", "CGUGGCC",
               "CGUGCAGC", "CUGUJUACG", "CUGUJUCAG", "CUGAGCCAG", "GACUJUGAG",
               "CUGUGGC", "CUUGGCG", "GCCUCGAG", "CUGCCAGAC", "CUGCCAGAC")

meltingBatch(sequence = seq, comp.seq = comp.seq, nucleic.acid.conc = 0.0004,
              hybridisation.type = "rnarna", nucleic.acid.conc = 0.0004,
              method.singleMM = "tr06")
```

print.melting

```
Prints melting temperature from a melting object
```

Description

`print.melting` prints to console the melting temperature value from an object of class `melting`.

Usage

```r
## S3 method for class 'melting'
print(x, ...)  
```

Arguments

- `x` An object of class `melting`.
- `...` Unused

Value

The melting temperature value (degree Celsius) in the console.

See Also

`melting`
Description
Not exported. Strictly internal

Usage
withWE(expr)

Arguments
expr The expression to be evaluated.

Value
• In case of Warning(s) Returns the value along with the warning message(s).
• In case of Error Returns NA as the value along with the error message.

Examples
foo <- function(){
  warning("oops")
  1
}

foo <- function(){
  warning("oops")
  warning("again oops")
  warning("again oops")
  1
}

foo <- function(){
  warning("oops")
  log("a")
}
Index

* internal
  withWE, 21

melting, 2, 19, 20
meltingBatch, 19

print.melting, 20

withWE, 21