

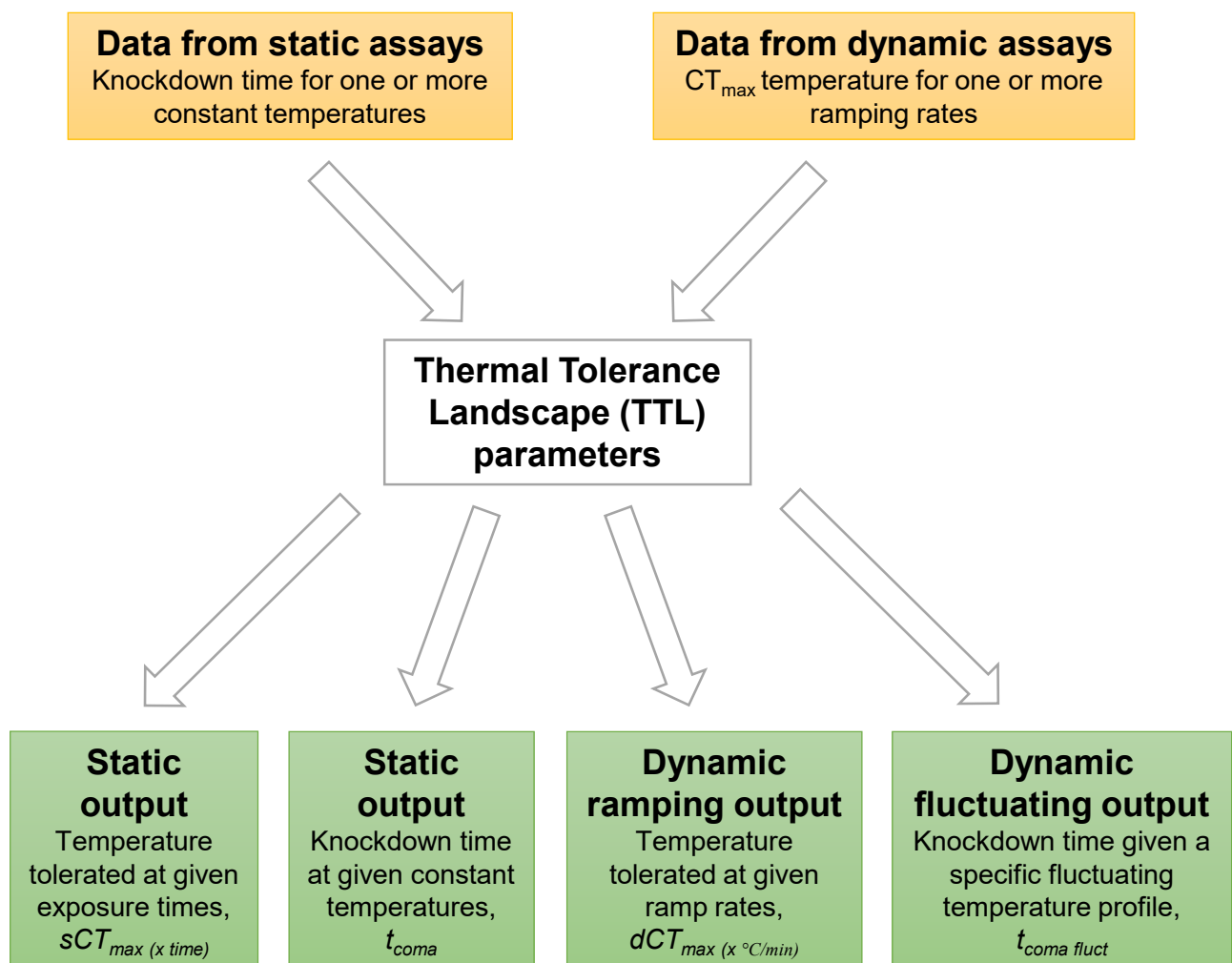
## Guide to R-scripts

As a practical application of the mathematical framework presented in the main text, we here provide R-scripts to derive parameters of the Thermal Tolerance Landscape (TTL) and use these to assess thermal tolerance limits. We present two scripts (**Fig. S2**). Which one you should use depend on the type of input data which depends on the type of experiment conducted:

- 1) **“TTL\_from\_Static.R”**. This script derives TTL parameters from static experiments, where time to failure ( $t_{coma}$ ) is measured at one or more constant temperatures. An input data template is provided (static\_input.csv)
- 2) **“TTL\_from\_Dynamic.R”**. This script derives TTL parameters from dynamic experiments, where the maximal temperatures tolerated (dynamic  $CT_{max}$ ,  $dCT_{max}$ ) are measured using one or more ramping rates. An input data template is provided (dynamic\_input.csv).

Both scripts use the derived TTL parameters to convert between and within static and dynamic measurements and input data of natural temperature fluctuations can be added to assess when failure occurs based on the derived TTL parameters.

In the contents below, you can click on the appropriate section depending on your type of data for details on the derivation of TTL parameters. Once the TTL parameters have been derived from either static or dynamic input data, you have four options within each script depending on which kind of output is wanted (**Fig. S2**).



**Supplementary Fig. S2.** Schematic workflow of scripts that derive TTL parameters and then allow conversion between and within static and dynamic output data along with prediction of injury accumulation under fluctuating temperatures.

## Guide contents

1	TTL from static input data .....	3
1.1	Output: Tolerable temperature at a given exposure time .....	4
1.2	Output: Knockdown time at a given temperature .....	6
1.3	Output: Dynamic CT <sub>max</sub> in ramping assays.....	6
1.4	Output: Knockdown time under randomly fluctuating temperatures.....	7
2	TTL from dynamic input data .....	9
2.1	Output: Tolerable temperature at a given exposure time .....	12
2.2	Output: Knockdown time at a given temperature .....	12
2.3	Output: Dynamic CT <sub>max</sub> in ramping assays.....	13
2.4	Output: Knockdown time under randomly fluctuating temperatures.....	14

# 1 TTL from static input data

The script “**TTL\_from\_Static.R**” derives TTL parameters from static experiments, where time to failure ( $t_{coma}$ ) is measured at one or more static temperatures. The corresponding input data template “**static\_assay.csv**” contains three columns:

group: Identity label, e.g. treatment group or species binomial name if multiple species have been assessed.

$t_{coma}$ : Time to failure [in minutes], e.g. time to onset of muscle loss. This can be the mean or median time of individuals assessed in each group at a given temperature.

assay\_temp: Temperature [in °C] of the static experiment.

For each unique input group ID, the script will fit the TTL as a linear regression:

$$\log_{10}(t_{coma}) = \beta \cdot \text{assay\_temp} + \alpha$$

Where  $\beta$  and  $\alpha$  is the slope and intercept, respectively. From the slope, the thermal sensitivity coefficient,  $z$  is calculated as  $z = -1/\beta$ . The intercept with the y-axis (i.e. when temperature = 0 °C) has no biological relevance for heat stress, and instead we see this as “a point on the line” which can be substituted by another value. In the main manuscript we present  $sCT_{max(1h)}$  (temperature that causes heat failure after a 1-hour exposure) as such an alternative value that, together with  $z$ , convey the same information of the TTL. The TTL is plotted for each unique group to visualize the spread around a linear relationship and  $R^2$  is provided (see **Fig. S3** for an example of the graphic output).

In the following, the numbered headings of each output type correspond to the same numbers given at the start of the script when selecting desired output types

## BOX 1.

In the case  $t_{coma}$  has only been assessed at one constant temperature, you must provide a guess of the value of  $z$ . For invertebrates and fishes, values of  $z$  in the range 1 to 5 is a reasonable starting point (see **Table S1**). Either supply a general value for all groups or make it group specific with a vector, where  $z$  can be called for each group.

CAUTION: The estimate of  $z$  has extreme consequences for model predictions and excessive extrapolations from the original data point should be treated with considerable caution (see discussion in the main text)

## 1.1 Output: Tolerable temperature at a given exposure time

The script can use TTL parameters to predict  $sCT_{max}$  [in °C] for other exposure durations ( $t_{coma}$ ), e.g. if you want to know what temperature your organisms can survive for e.g. 1 h ( $sCT_{max(1h)}$ ) or 1 week ( $sCT_{max(1week)}$ ). The desired additional exposure durations can be provided in the object 'extra\_t\_coma' [in minutes], note that 1 day = 1440 minutes.  $sCT_{max}$  for each supplied extra  $t_{coma}$  is determined from the TTL parameters  $\beta$  and  $\alpha$ :

$$sCT_{max(t_{coma})} = \frac{\log_{10}(t_{coma}) - \alpha}{\beta}$$

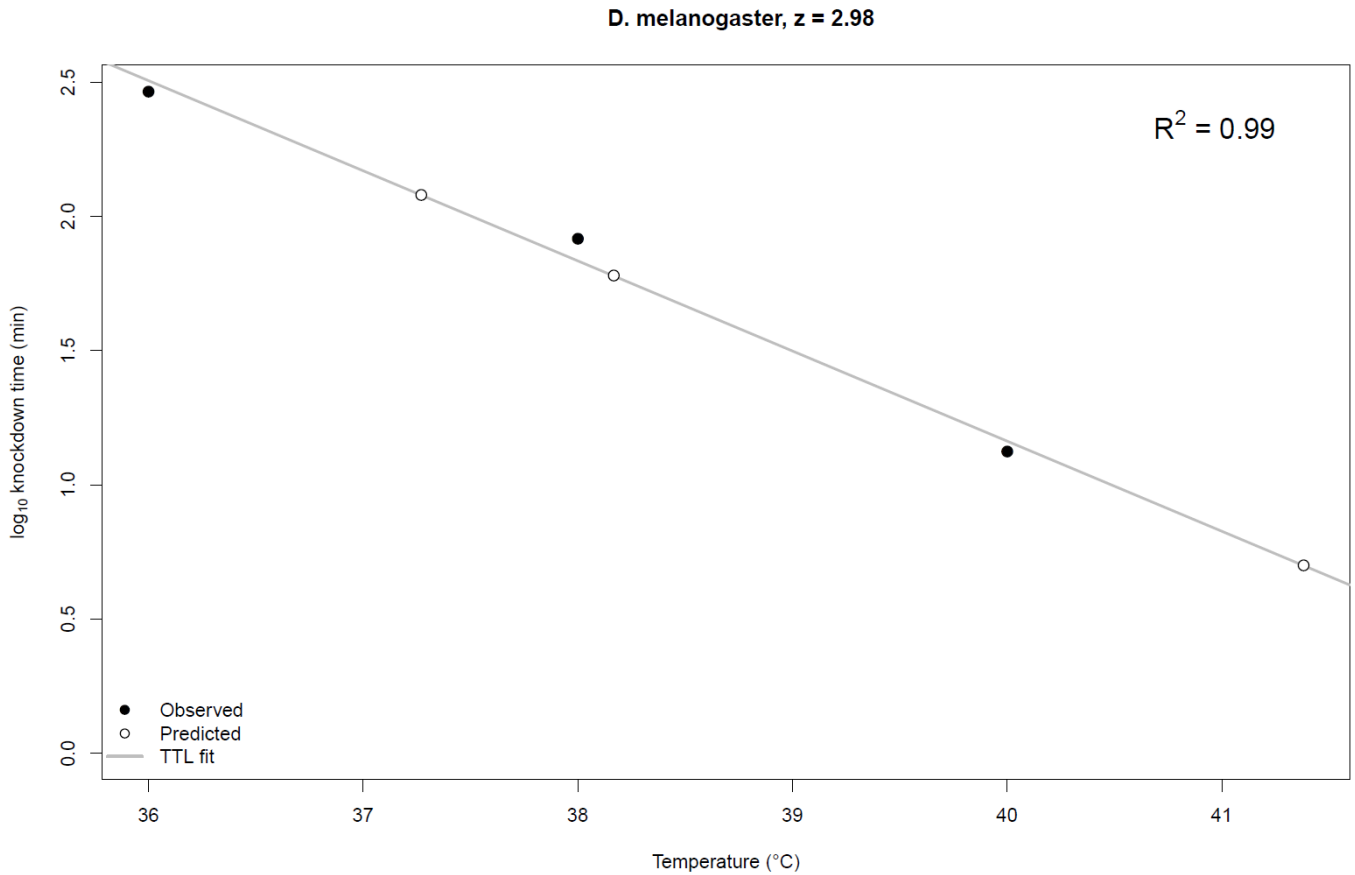
Alternatively, if a point on the line (e.g.  $sCT_{max(1h)}$  with  $t_{coma} = 1$  h) and  $z$  is available, a similar calculation can be made:

$$sCT_{max(t_{coma})} = sCT_{max(1h)} - z \cdot \log_{10}\left(\frac{t_{coma}}{t_{coma(1h)}}\right)$$

Where  $sCT_{max(1h)}$  and the corresponding  $t_{coma(1h)}$  can be substituted by any point on the line, including the intercept (0, $\alpha$ ).

If additional  $t_{coma}$  times are provided, a csv table named "extra\_t\_coma.csv" is produced containing the input data along with the predicted  $sCT_{max}$  at the given  $t_{coma}$  times.

**CAUTION:** Severe extrapolation outside the time and temperature domain that was used to parameterize the TTL increases uncertainty of predictions, i.e. if observed  $t_{coma}$  is within minutes to hours, the model cannot confidently predict  $sCT_{max}$  in tests that last days (See main text).



**Supplementary Fig. S3.** Example of the graphic output of the “TTL\_from\_Static.R” script, which plots observed static tolerances (filled circles). Here we show an empirical example with *Drosophila melanogaster* from Jørgensen et al. (2019b), with three observed  $t_{\text{coma}}$  durations at 36, 38, and 40 °C ( $t_{\text{coma}} = 292.0, 82.25$ , and 13.25 minutes, respectively). The TTL is fitted as the regression of  $\log_{10}(t_{\text{coma}})$  on assay temperature (grey solid line), and the  $R^2$  value is provided. From the slope of the regression,  $z$  is calculated ( $z = -1/\text{slope}$ ) and is provided in the plot title to allow an easy comparison of species specific  $z$  values. If additional  $t_{\text{coma}}$  times are provided (section 1.1) or additional temperatures are provided (section 1.2), these will be predicted from the TTL parameters and plotted alongside the observed static tolerances (open circles, note they are positioned exactly on the TTL line). Here we provide an example of three additional  $t_{\text{coma}}$  durations (5, 60, and 120 minutes), for which the tolerable temperature is predicted (41.38, 38.17, and 37.27 °C, respectively)

## 1.2 Output: Knockdown time at a given temperature

The script can use TTL parameters to predict  $t_{coma}$  [in minutes] at other static temperatures, e.g. if you want to know how long your organisms can survive at a stressful static temperature. The desired extra static temperatures can be provided in the object 'extra\_sCTmax' [in °C]. The model can only handle positive temperatures (but in cases where cold stress is considered and subzero temperatures are relevant it is easy to convert all measures to the kelvin scale, however models assumptions and accuracy have not presently been tested for cold stress).  $t_{coma}$  for each additionally supplied  $sCT_{max}$  is determined from the TTL parameters  $\beta$  and  $\alpha$ :

$$t_{coma} = 10^{\alpha + (\beta \cdot sCT_{max})}$$

Alternatively, if a point on the line (e.g.  $sCT_{max(1h)}$  with  $t_{coma} = 1$  h) and  $z$  is available, a similar calculation can be made:

$$t_{coma} = t_{coma(1h)} \cdot 10^{\frac{sCT_{max(1h)} - sCT_{max}(t_{coma})}{z}}$$

Where  $sCT_{max(1h)}$  and the corresponding  $t_{coma(1h)}$  can be substituted by any point on the line, including the intercept (0, $\alpha$ ).

If additional temperatures are provided, these will also be plotted in the TTL plot alongside observed static tolerances (see **Fig. S3** for an example), and a csv table named "extra\_sCTmax.csv" is produced containing the input data along with the predicted  $t_{coma}$  at the given  $sCT_{max}$  temperatures.

**CAUTION:** Severe extrapolation outside the time and temperature domain that was used to parameterize the TTL increases uncertainty of predictions (see main text).

## 1.3 Output: Dynamic $CT_{max}$ in ramping assays

The script can use TTL parameters to predict  $dCT_{max}$  in dynamic assays where temperature is changed at a constant rate (ramp rate). In the object 'ramprates', you can supply the rates of temperature change [in °C/min] for which  $dCT_{max}$  should be predicted. Rates often range between 0.01 to 1.00 °C/min, but any positive numerical input is allowed.

**CAUTION:** consider the use of very slow or very fast ramping rates in the model only with caution as time or thermal equilibrium may be problematic (See main text and section 1.1).

The parameters used to estimate  $dCT_{max}$  are:

- $z$ : temperature sensitivity coefficient [dimensionless] from TTL or provided (see Box 1)
- $t_{LS}$ : the time where the critical amount of injury has accumulated resulting in coma, i.e.  $t_{coma}$ , time of static  $CT_{max}$  ( $sCT_{max}$ ) [min]
- $sCT_{max}$ :  $sCT_{max}$  [°C] at  $t_{LS}$  ( $t_{coma}$ )
- $T_0$ : ramp start temperature [°C]
- $T_{c*}$ :  $T_c$  is the temperature above which net injury accumulation starts. As the ‘true’  $T_c$  is rarely known, we recommend using some convenient value  $T_{c*}$  below the true  $T_c$ , e.g. the rearing temperature. See main text for a discussion on this and **Fig. S1 in supplementary information** for further justification.

The script sets  $t_{LS}$  to the highest  $t_{coma}$  and the corresponding  $sCT_{max}$  is used. Then for each ramping rate,  $b$ ,  $dCT_{max}$  [in °C] is estimated (equation 7a in main text):

$$dCT_{max} = T_0 + \frac{z}{\ln(10)} \ln \left[ \frac{\ln(10) \cdot b \cdot t_{LS}}{z} \cdot e^{\frac{\ln(10)}{z}(sCT_{max}-T_0)} + e^{\frac{\ln(10)}{z}(T_{c*}-T_0)} \right]$$

Here we substituted  $k$  in equation 7a with  $\ln(10)/z$ . Note that in  $R \log()$  is the natural logarithm, whereas  $\log_{10}()$  is the base 10 or “common” logarithm.

If other ramp rates are provided, a csv table named “ $dCT_{max}$ \_predictions.csv” is produced with the predicted  $dCT_{max}$  for each supplied ramp rate.

## 1.4 Output: Knockdown time under randomly fluctuating temperatures

From the TTL parameters ( $\beta$  and  $\alpha$ ) estimated from the static assays (observed  $sCT_{max}$  temperatures and  $t_{coma}$  durations), the script can predict time to failure for experiments in which temperature fluctuates either randomly or predictably, from a specific fluctuating temperature profile. An input template is provided (“fluctuating\_temperature\_profile.csv”). Time must be provided in minutes, e.g. 10 seconds as 1/6 min, and temperature at that time must be supplied in °C.

Assuming additivity of thermal injury, time to failure is predicted based on the sum of accumulated thermal injury:

$$\text{Accumulated injury} = \sum_{i=1}^{t_e} \frac{100 \cdot (t_{i+1} - t_i)}{10^{(\beta \cdot \max(T_i, T_{i+1}) + \alpha)}}$$

Which sums the additive injury over each time interval  $i$  until  $t_e$  which is the time interval for which the total accumulated injury is calculated. The accumulated injury for which the tolerable exposure duration should be predicted (corresponding to  $t_e$ , the time interval where summation of cumulative injury should stop) can be set to any percent lethal damage above 0 and up to 100. In the case of 100 % accumulated injury  $t_e$  equals  $t_{coma}$ . At each time step  $t_i$ , the interval in minutes until the next measurement is determined regardless of whether temperatures have been recorded with equal intervals. The denominator is the fraction of the tolerable exposure duration for the maximum temperature (of  $T_i$  and  $T_{i+1}$ ) in each time interval. The time where accumulated injury is closest to the set percent lethal damage is returned for each group provided. If a fluctuating temperature profile is provided a csv table named “fluctemp\_predictions.csv” is produced.

**CAUTION:** Temperatures below the damage accumulation threshold might “repair” thermal injury (see main text).



## 2 TTL from dynamic input data

The script “**TTL\_from\_Dynamic.R**” derives TTL parameters from dynamic experiments, where temperature is changed at a constant rate (ramp rate) and the maximal temperatures tolerated ( $dCT_{max}$ ) are measured using one or more ramping rates. The corresponding input data template “**dynamic\_input.csv**” contains three columns:

group:	Identity label, e.g. treatment group or species binomial name if multiple species have been assessed
ramprate:	Temperature change [in °C/min]
$dCT_{max}$ :	Temperature at time of failure [in °C] of the dynamic experiment

The parameters used to estimate  $sCT_{max}$  are:

$t_{Ls}$ :	The time where the critical amount of injury has accumulated resulting in coma, i.e. $t_{coma}$ , time of static $CT_{max}$ ( $sCT_{max}$ ) [min] given in the object ‘t_coma’ (see below)
ramp rate:	Temperature change [in °C/min]
$dCT_{max}$ :	Temperature at time of failure [in °C] at a given ramp rate
$T_0$ :	ramp start temperature [°C]
$T_{c*}$ :	$T_c$ is the temperature where damage accumulation starts. As the ‘true’ $T_c$ is rarely known, however, we recommend using some convenient value $T_{c*}$ below the true $T_c$ (e.g. the rearing temperature). See main text for a discussion on this and <b>Fig. S1 in supplementary information</b> for further justification.

**NOTE:** In order to get a starting point from which to build the TTL, at least one  $t_{coma}$  value MUST be supplied (a point on the line) in the object ‘t\_coma’. The script will stop and print an error message if it is not provided. We recommend a  $t_{coma}$  value of 1 h, at this will accommodate the duration of most thermal assays. If you have dynamic assays than span a duration much shorter or much longer than 1 h, consider using the average duration as a starting  $t_{coma}$  point for parameterizing the TTL. If several tolerable static temperatures should be predicted as the output (section 2.1), you can supply several  $t_{coma}$  values in this object [in mins]. The script will derive TTL parameters and predict  $sCT_{max}$  for each of the supplied  $t_{coma}$  ( $t_{Ls}$ ) durations. There are three different methods depending on how many ramp rates  $dCT_{max}$  has been assessed for:

### From dCT<sub>max</sub> at three or more ramp rates

In the case of three or more available ramp rates, the script fits a non-linear model on dCT<sub>max</sub> and the corresponding ramp rate  $b$  via the nonlinear least squares (nls) function in R to estimate  $z$  and sCT<sub>max</sub> for each supplied  $t_{coma}$  duration (equation 7a in main text):

$$dCT_{max} = T_0 + \frac{z}{\ln(10)} \ln \left[ \frac{\ln(10) \cdot b \cdot t_{Ls}}{z} \cdot e^{\frac{\ln(10)}{z}(sCT_{max}-T_0)} + e^{\frac{\ln(10)}{z}(T_c-T_0)} \right]$$

Here we substituted  $k$  in equation 7a with  $\ln(10)/z$ . Note that in R  $\log()$  is the natural logarithm, whereas  $\log10()$  is the base 10 or “common” logarithm. Also note,  $z$  is independent of  $t_{coma}$  ( $t_{Ls}$ ).  $nls()$  iterates from a user supplied starting position for each variable, with the argument ‘start = list(sCTmax=35, z=2.5)’ (example starting values). Note the order of variables in this list must be maintained. This can be made group specific with a starting value vector before the loop over groups and calling the individual values at each iteration,  $i$ , but generally  $nls$  performs well with reasonable guesses across groups.

### From dCT<sub>max</sub> at two ramp rates

In the case of two ramp rates, the script sets up two equations with two unknowns with the two ramp rates and corresponding values of dCT<sub>max</sub> and solves for  $z$  and sCT<sub>max</sub> with the ‘rootSolve’ R package (this is installed and loaded initially if not already present; Soetaert 2009; Soetaert & Herman 2009). The multiroot() function must be supplied with starting values (start guesses) with the argument ‘start = c(2.5,35)’ (example starting values for  $z$  and sCT<sub>max</sub>, respectively, note the order of variables must be maintained). Only positive solutions are allowed.

### From dCT<sub>max</sub> at one ramp rate

In the case where dCT<sub>max</sub> has been determined in an experiment with only a single ramp rate, you need to supply a value of  $z$ . **CAUTION: The estimate of  $z$  has extreme consequences for model predictions and excessive extrapolation from the original data-point should be treated with considerable consideration (see discussion in the main text and Box 1 for advice and considerations on selecting appropriate values of  $z$ )**

sCT<sub>max</sub> is determined for each  $t_{coma}$  duration ( $t_{Ls}$ ) (equation 7b in main text):

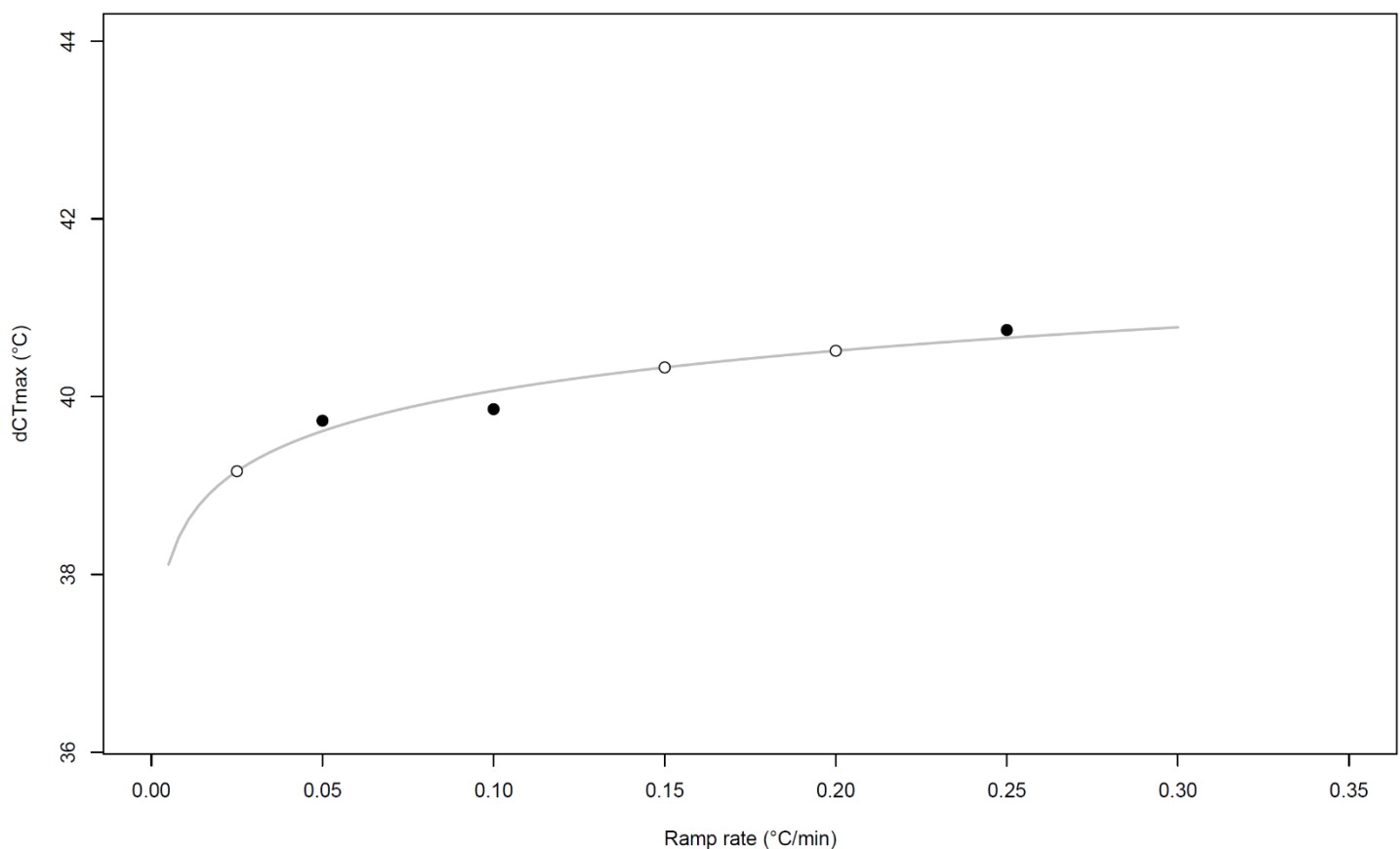
$$sCT_{max} = T_0 + \frac{z}{\ln(10)} \ln \left[ \frac{z}{\ln(10) \cdot b \cdot t_{Ls}} \left( e^{\frac{\ln(10)}{z}(dCT_{max}-T_0)} - e^{\frac{\ln(10)}{z}(T_c-T_0)} \right) \right]$$

Here we substituted  $k$  with  $\ln(10)/z$ . Note that in R  $\log()$  is the natural logarithm, whereas  $\log10()$  is the base 10 or “common” logarithm.

For each unique group,  $dCT_{max}$  is plotted as a function of ramp rate to visualize the goodness of fit. If additional dynamic tolerances are desired (see section 2.3 below), these will also be plotted in the same plot (see **Fig. S4** for an example of the graphic output).

In the following, the numbered headings of each output type correspond to the same numbers given at the start of the script when selecting desired outputs. Note that Output 2.1: Tolerable temperature at a given exposure time, is not optional as  $t_{coma}$  must be supplied (see above).

**D. melanogaster, z = 2.28**



**Supplementary Fig. S4.** Example of the graphic output of the “TTL\_from\_Dynamic.R” script, which plots observed dynamic tolerances (filled circles), as a function of ramp rate [°C/min]. Here we show an empirical example with *Drosophila melanogaster* from Jørgensen et al. (2019b), with three observed dynamic  $CT_{max}$  ( $dCT_{max}$ ) temperatures from experiments employing ramp rates of 0.05, 0.10, and 0.25 °C/min ( $dCT_{max}$  = 39.73, 39.86, and 40.75°C, respectively). The grey solid line represents the fitted relationship between  $dCT_{max}$  and ramp rate (in this example via nonlinear least squares (nls) function). The derived value of  $z$  is provided in the plot title to allow an easy comparison of species-specific values of  $z$ . If additional ramp rates are provided (section 2.3),  $dCT_{max}$  at these rates will be predicted from the TTL parameters and plotted as well (open circles, note they are positioned exactly on the dynamic TTL line). Here we provide an example of three additional ramp rates (0.025, 0.15, and 0.20 °C/min), for which  $dCT_{max}$  is predicted (39.16, 40.33, and 40.52°C, respectively).

## 2.1 Output: Tolerable temperature at a given exposure time

The script can use TTL parameters to predict  $sCT_{max}$  in static assays, with constant temperature exposure. In the object 't\_coma', you can supply a range of exposure durations [in minutes] for which  $sCT_{max}$  should be predicted (note that for TTL parameterization from dynamic assays already at least one  $t_{coma}$  has been supplied, see section 2 TTL from dynamic input data). Any positive input is allowed.

**CAUTION:** Severe extrapolation outside the time and temperature domain that was used to parameterize the TTL model increases uncertainty of predictions, i.e. if observed  $t_{coma}$  is within minutes to hours, the model cannot confidently predict  $sCT_{max}$  in tests that last days.

Regardless of the method for estimating  $sCT_{max}$  and  $z$  (see the three methods above), the output of the script is a csv table named "sCTmax\_predictions.csv" with the predicted  $sCT_{max}$  and  $z$  for each  $t_{coma}$  for each unique group ID.

## 2.2 Output: Knockdown time at a given temperature

The script can also use TTL parameters to predict  $t_{coma}$  times [in minutes] at other static temperatures, e.g. if you want to know how long your organisms can survive at a specific stressful temperature. The desired additional static temperatures can be provided in the object 'extra\_sCTmax' [in °C]. The model can only handle positive temperatures (but in cases where cold stress is considered and subzero temperatures are relevant it is easy to convert all measures to the kelvin scale, however models assumptions and accuracy have not presently been tested for cold stress).  $t_{coma}$  for each additionally supplied  $sCT_{max}$  is determined from the TTL parameters  $\beta$  and  $\alpha$ :

$$t_{coma} = 10^{\alpha + (\beta \cdot sCT_{max})}$$

Alternatively, if a point on the line (e.g.  $sCT_{max(1h)}$  with  $t_{coma} = 1$  h) and  $z$  is available, a similar calculation can be made:

$$t_{coma} = t_{coma(1h)} \cdot 10^{\frac{sCT_{max(1h)} - sCT_{max}(t_{coma})}{z}}$$

Where  $sCT_{max(1h)}$  and the corresponding  $t_{coma(1h)}$  can be substituted by any point on the line, including the intercept  $(0, \alpha)$ .

**CAUTION:** Severe extrapolation outside the time and temperature domain that was used to parameterize the TTL model increases uncertainty of predictions, i.e. if observed  $sCT_{max}$  is within 36-40 °C, the model cannot confidently predict  $t_{coma}$  for temperatures far outside this range, e.g. 50 °C.

If additional temperatures are provided, a csv table named “extra\_sCTmax.csv” is produced containing the predicted sCT<sub>max</sub> for each supplied t<sub>coma</sub> (see section 2.1) along with the predicted t<sub>coma</sub> times at additionally supplied sCT<sub>max</sub> temperatures.

## 2.3 Output: Dynamic CT<sub>max</sub> in ramping assays

The script can predict dCT<sub>max</sub> for additional ramp rates not included in the input data. In the object ‘extra\_ramprates’, you can supply additional rates of temperature change [in °C/min] for which dCT<sub>max</sub> should be predicted. Rates often range between 0.01 to 1.00 °C/min, but any positive numerical input is allowed.

**CAUTION: consider the use of very slow or very fast ramping rates in the model only with caution as time or thermal equilibrium may be problematic (See main text and section 1.1).**

From the derived z and sCT<sub>max</sub> at given t<sub>coma</sub> times (at least one), dCT<sub>max</sub> is predicted for each supplied ramp rate (equation 7a in main text):

$$dCT_{max} = T_0 + \frac{z}{\ln(10)} \ln \left[ \frac{\ln(10) \cdot b \cdot t_{Ls}}{z} \cdot e^{\frac{\ln(10)}{z}(sCT_{max}-T_0)} + e^{\frac{\ln(10)}{z}(T_c-T_0)} \right]$$

where *b* is the ramp rate. Here we substituted *k* in equation 7a with ln(10)/z. Note that in *R* log() is the natural logarithm, whereas log10() is the base 10 or “common” logarithm.

If additional ramp rates were provided, a csv table named “dCTmax\_extra\_ramprates.csv” is produced containing the predicted dCT<sub>max</sub> for each supplied ramp rate.

## 2.4 Output: Knockdown time under randomly fluctuating temperatures

From the TTL parameters ( $\beta$  and  $\alpha$ ) estimated from the dynamic ramping assay (predicted  $sCT_{max}$  at specified  $t_{coma}$  durations), time to failure can be predicted for experiments in which temperature fluctuates either randomly or predictably. If desired, you must provide a fluctuating temperature profile. An input template is provided (“fluctuating\_temperature\_profile.csv”). Time must be provided in minutes, e.g. 10 seconds must be given as 1/6 min, and temperature at that time must be supplied in °C.

Assuming additivity of thermal injury, time to failure can be predicted based on the sum of accumulated thermal injury:

$$\text{Accumulated injury} = \sum_{i=1}^{t_e} \frac{100 \cdot (t_{i+1} - t_i)}{10^{(\beta \cdot \max(T_i; T_{i+1}) + \alpha)}}$$

Which sums the additive injury over each time interval  $i$  until  $t_e$  which is the time interval for which the total accumulated injury is calculated. The accumulated injury for which the tolerable exposure duration should be predicted (corresponding to  $t_e$ , the time interval where summation of cumulative injury should stop) can be set to any percent lethal damage above 0 and up to 100. In the case of 100 % accumulated injury  $t_e$  equals  $t_{coma}$ . At each time step  $t_i$ , the interval in minutes until the next measurement is determined regardless of whether temperatures have been recorded with equal intervals. The denominator is the fraction of the tolerable exposure duration for the maximum temperature (of  $T_i$  and  $T_{i+1}$ ) in each time interval. The time where accumulated injury is closest to the set percent lethal damage is returned for each group provided. If a fluctuating temperature profile is provided a csv table named “fluctemp\_predictions.csv” is produced.

**CAUTION:** Temperatures below the damage accumulation threshold might “repair” thermal injury. (see main text).

**Supplementary Table S1.** Overview of the values of  $z$  used for each species in cases where only a single data entry was available for TTL parameterization. These mean values of  $z$  for each species are based on TTL parameterization where sufficient (i.e. at least two) data entries were available and may thus include both TTLs based on static and dynamic data. The range of the values of parameterized  $z$  is given along with the number ( $n$ ) of TTLs that the mean value of  $z$  is calculated from (this number corresponds to the number of triangle points in Fig. 5 in the main text).

Group	Species	Mean parameterized $z$	$z$ range	$n$
Insects	<i>Drosophila melanogaster</i>	2.55	1.5 - 3.1	7
	<i>Drosophila subobscura</i>	3.40	2.6 - 4.7	9
	<i>Glossina pallidipes</i>	5.15	4.7 - 5.8	3
	<i>Tenebrio molitor</i>	0.90	-	1
Springtails	<i>Orchesella cincta</i>	1.80	1.8 - 1.8	2
	<i>Folsomia candida</i>	3.30	-	1
Waterfleas	<i>Daphnia magna</i>	2.20	0.8 - 3.4	3
Fishes	<i>Gambusia affinis</i>	1.80	1.1 - 2.4	7
	<i>Salmo salar</i>	1.78	1.4 - 2.2	8
	All species	2.54	0.8 - 5.8	41

## Supplementary references

- Jørgensen, L.B., Malte, H. & Overgaard, J. (2019b). Data from: How to assess *Drosophila* heat tolerance: unifying static and dynamic tolerance assays to predict heat distribution limits. Dryad Digital Repository. <https://doi.org/10.5061/dryad.840j728>.
- Soetaert, K. (2009). rootSolve: Nonlinear root finding, equilibrium and steady-state analysis of ordinary differential equations. R-package version 1.6.
- Soetaert, K. & Herman, P. (2009). *A Practical Guide to Ecological Modelling. Using R as a Simulation Platform*. Springer, New York, NY, USA.