## Estimating assimilation rates by cells in Matlab

Use the following steps to determine the best estimates and errors (uncertainties) of the carbon-specific (*k*, Δ*k*) and cell-specific (*r*, Δ*r*) rates of C assimilation in cells from nanoSIMS data. The input data is organized in a spreadsheet (e.g., in Excel), the calculations are implemented in a custom-made Matlab function ***RatesCells.m***. Although the description here assumes assimilation of C, the function is applicable for any other element.

In the Monte-Carlo method used for the calculation of the rate, the C content of the cell is sampled from a distribution described by the probability distribution function θ (Eq. 18), with and *Cmax­* calculated as described above and *C0 = C*. Thus, the cell is assumed to originate from a population of cells with partially synchronized cell cycles. If Δ*V* is set to 0, assimilation rates are calculated assuming that the measured cell originates from a population of cellswith *perfectly unsynchronized* cell cycles, where each cell assimilates C at a constant cell-specific rate, *r* (see Approach A, Eq. 21). In this case, the value of *V* is irrelevant, as only the value of is used to calculate *r*. If Δ*V* is set to a *positive* value, Approach B is used to calculate the assimilation rates (Eq. 22). In this case, it is assumed that the measured cell originates from a population of *partially synchronized* cells characterized by known values of *C0* and Δ*C* (see Eq. 18 and **Supplementary Figure S1 and S2**). Hence, Δ*V* should be calculated as Δ*V* = Δ*C*/ *ρ* before entering the value in the spreadsheet cell.

1. **Execute the function *RatesCells* in Matlab.**
   1. Copy the spreadsheet with the input data to the same folder as the file ***RatesCells.m*.**
   2. Start Matlab and open the file ***RatesCells.m*** in the Matlab editor. If necessary, change function’s default parameters (see comments within the file for explanation). Execute the function by clicking the green “Run” arrow (**Supplementary Figure S3b**).
   3. The output is exported in the spreadsheet ***ResultsCells1*.*xlsx*** (**Supplementary Figure S3c**), including the following values:

## Estimating uncertainty of assimilation rates

The approaches to estimate the error (uncertainty) of the calculated assimilation rates, Δ*k* and Δ*r*, depend on the model used to approximate C assimilation. For the first-order kinetics model, Δ*k* and Δ*r* are estimated using the well-known rules of error propagation (Fitzsimons et al., 2000). Specifically, by combining the derivatives , and calculated from equations 10, 20 and 21, respectively, we obtain the following expressions:

where Δ*x* is the analytical error of *x*. The value of Δ*x* depends on the counts of the 13C and 12C ions detected from the measured cell and is exported by programs used for nanoSIMS data processing (e.g., Look@NanoSIMS; Polerecky et al., 2012).

For the zero-order kinetics model of C assimilation, the errors are estimated through Monte-Carlo simulations. First, a large number of values of *xj* and *Cj* are randomly generated. When applying approaches A and B, the values of *Cj* are assumed to be distributed according to Eq. 16 and 18, respectively, while for both approaches the values of *xj* are assumed to be normally distributed with the standard deviation equal to Δ*x*. The numerical procedure described for approaches A and B is then used to calculate *rj* for each pair of values *xj* and *Cj*. Finally, Δ*r* is defined as the standard deviation of the calculated values *rj*, and Δ*k* is calculated based on Eq. 20, i.e., This procedure to estimate the errors Δ*k* and Δ*r* is given as part of the same Matlab function used for finding the best estimates of the rates (**Supplementary Methods, Section 1.2**).

## Graphical user interface, table, Excel Description automatically generatedText Description automatically generatedTable Description automatically generatedSupplementary Figure S3.

**(c)**

**(b)**

**(a)**

Calculation of substrate assimilation rates by cells in Matlab. Shown are screenshots of **(a)** exampleinput data organized in a spreadsheet; **(b)** Matlab editor with the function **RatesCells.m** open (the function is executed by clicking on the green “Run” arrow); **(c)** example of the output data organized in a spreadsheet. See **Supplementary Methods, Section 1.1,** for the explanation of the different input and output data columns.