

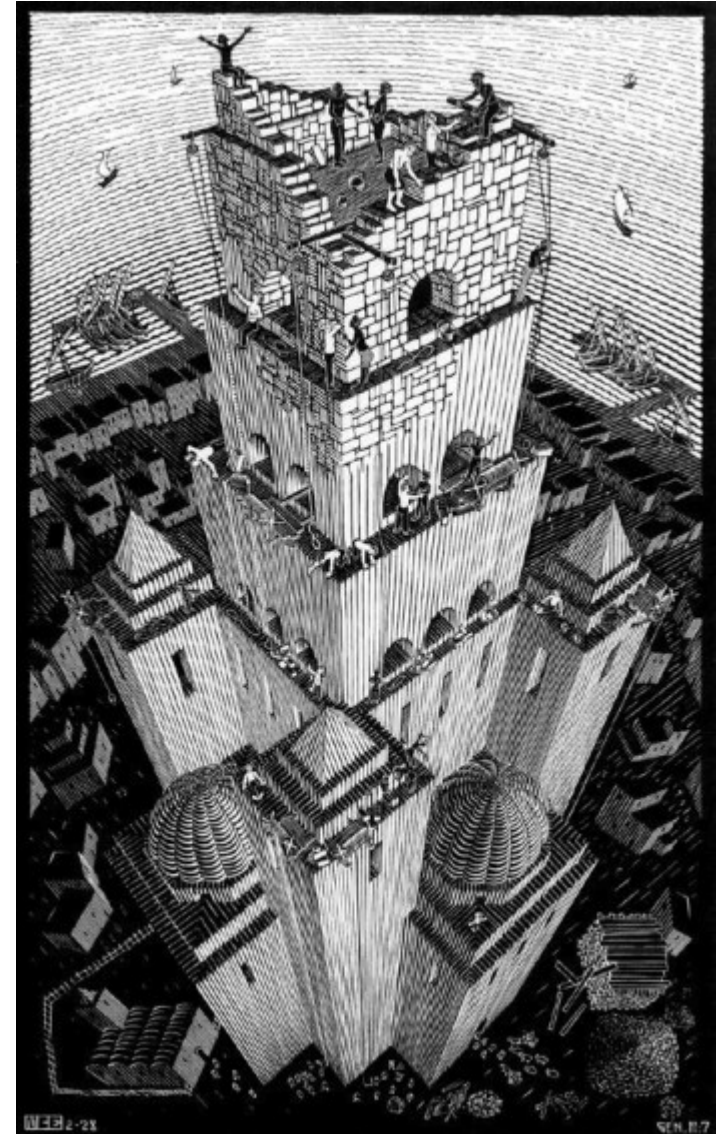
# Species parameter estimation using traits4models

Miquel De Cáceres, Rodrigo Balaguer  
Ecosystem Modelling Facility, CREAM

# Outline

1. Why package **traits4models**?
2. Harmonization of plant trait databases
3. Harmonization of allometry databases
4. Populating species parameter tables for medfate

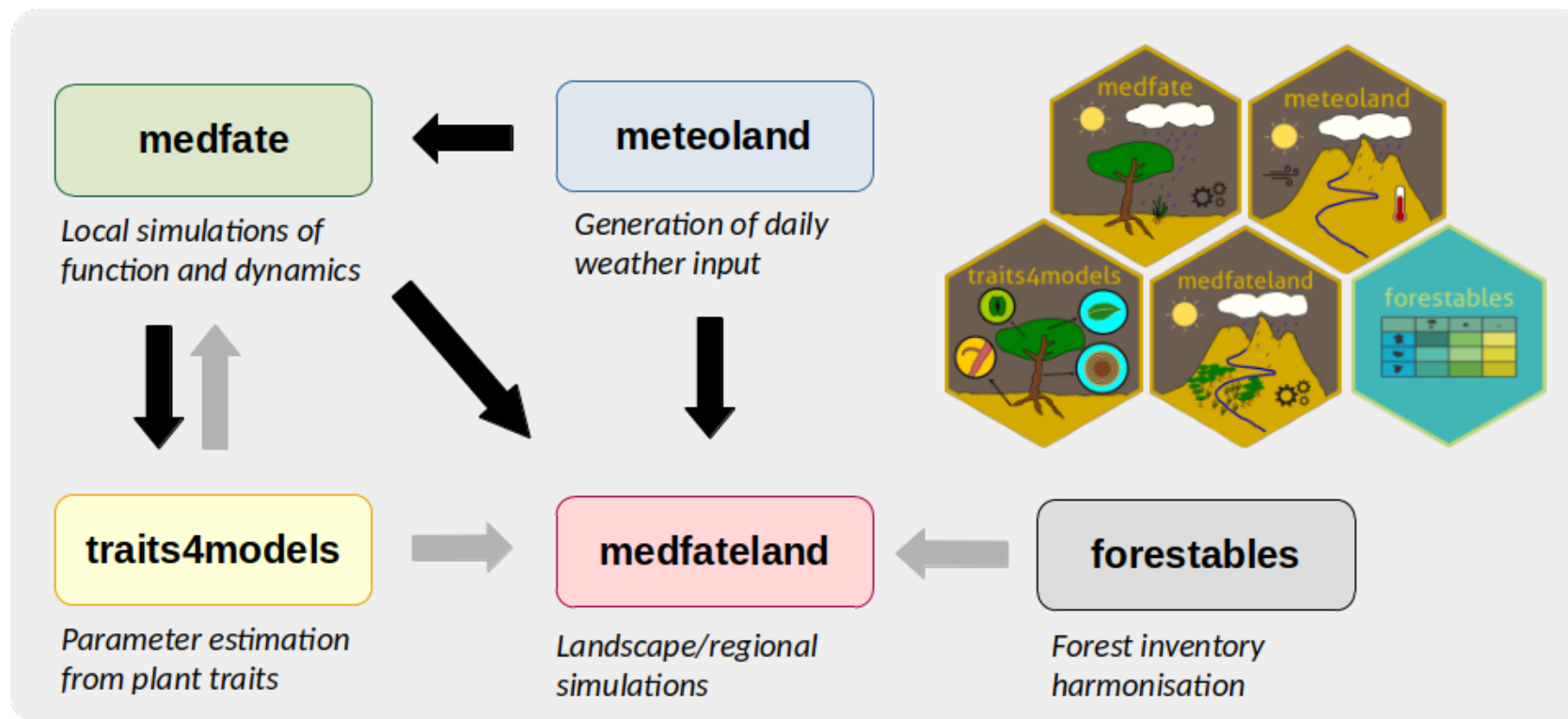
M.C. Escher - Babel tower, 1928



# 1. Why package `traits4models`?

# Purpose

- Process-based models like **medfate** are challenging to parameterize for new areas!
- Package **traits4models** is designed to facilitate creating plant species parameter tables for process-based models of forest function and/or dynamics.
- Utility functions are provided to:
  - a. Harmonize *plant trait* databases.
  - b. Harmonize *plant allometry* databases.
  - c. Populate model *species parameter tables* from harmonized data.



## 2. Harmonization of plant trait databases

## Why harmonization?

1. Species parameter values cannot be drawn from a single data source.
2. The availability of plant trait data **progressively increases**, as additional efforts are made in observational or experimental studies.
3. Multiple data sources need to be harmonized before species parameter tables are build.

# Harmonization procedure

## General procedure

For each trait data source, harmonization needs to be conducted in terms of:

- **Task a** - Definition and nomenclature of measured traits
- **Task b** - Trait measurement units
- **Task c** - Taxonomy of the plant entities

## How it is done in traits4models?

- The package provides a data table called `HarmonizedTraitDefinition` that presents plant trait definitions and their required notation and units.
- Tasks **a** and **b** should be performed manually using **tidyverse** functions.
- Task **c** is conducted with the help of the taxonomic backbone provided by [World Flora Online](#)<sup>1</sup> and using function `harmonize_taxonomy_WFO()`.
- Function `check_harmonized_trait()` allows checking that the harmonized source is acceptable.
- Results are stored as `csv` or `rds` (compressed) format.
- The procedure is illustrated in vignette [Trait database harmonization](#).

1. Kindt, Roeland. 2020. "WorldFlora: An R Package for Exact and Fuzzy Matching of Plant Names Against the World Flora Online Taxonomic Backbone Data." *Applications in Plant Sciences* 8 (9): e11288. <https://doi.org/10.1002/app.211288>

### **3. Harmonization of allometry databases**



# Harmonization procedure

## General procedure

For each allometry data source, harmonization needs to be conducted in terms of:

- **Task a** - Definition and nomenclature of **response** and **predictor** variables
- **Task b** - Formulation of allometric equation
- **Task c** - Taxonomy of the plant entities

## How it is done in traits4models?

- Tasks **a** and **b** should be performed manually using **tidyverse** functions but following a **harmonized vocabulary** to describe the *equation*, the *response variable*, the *predictor variables* and the *parameter names*.
- Task **c** is conducted with the help of the taxonomic backbone provided by [World Flora Online](#)<sup>1</sup> and using function `harmonize_taxonomy_WFO()`.
- Function `check_harmonized_allometry()` allows checking that the harmonized source is acceptable.
- Results are stored as `csv` or `rds` (compressed) format.
- The procedure is illustrated in vignette [Allometry database harmonization](#).

1. Kindt, Roeland. 2020. "WorldFlora: An R Package for Exact and Fuzzy Matching of Plant Names Against the World Flora Online Taxonomic Backbone Data." *Applications in Plant Sciences* 8 (9): e11288. <https://doi.org/10.1002/app2.11288>

## 4. Populating species parameter tables for medfate

# Parameter estimation procedure

The following sections describe the main steps used to obtain the species parameter table for **medfate** and the functions in **traits4models** that assist the process:

#	Step	Function(s)
1	Initialize the parameter table with target taxonomic entities	<code>init_medfate_params()</code>
2	Populate species parameters from forest inventory data	<code>fill_medfate_inventory_traits()</code>
3	Populate plant allometric coefficients from suitable databases	<code>fill_medfate_allometries()</code>
4	Populate plant functional traits from harmonized data bases	<code>fill_medfate_traits()</code>
5	Checking the final parameter table	<code>check_medfate_params()</code>
6	If necessary, completing strict parameters	<code>complete_medfate_strict()</code>

The procedure is illustrated in vignette [Building species parameter tables for medfate](#).

## Warning

- Unfortunately, this procedure is not sufficient, as it does not include **metamodelling** and **calibration** exercises that are needed for some specific parameters!
- In the future, function `check_medfate_params()` should include verification of the relationships between physiological traits.

# SpParamsMED vs. traits4models

Species trait parameter table [SpParamsMED](#) was created before **traits4models** and is provided in **medfate** as default parameter table:

```
1 dim(medfate::SpParamsMED)
```

```
[1] 217 157
```

However, the taxonomy of [SpParamsMED](#) is a **simplification** of that of the Spanish National Forest Inventory (IFN).

Package **traits4models** provides [SpParamsES](#) which has more taxonomic entities, matching IFN taxonomy and ready to be used when parsing forest plot data with package **forestables**:

```
1 dim(traits4models::SpParamsES)
```

```
[1] 530 156
```

Data sets [SpParamsES](#), [SpParamsFR](#) and [SpParamsUS](#) provided in **traits4models** follow the populating procedure explained here.

[SpParamsMED](#) is still supported, but [SpParamsES](#), [SpParamsFR](#) and [SpParamsUS](#) are more likely to be **maintained** and **curated** in the long term.

M.C. Escher - Babel tower, 1928

