

Calculating biomass from a tree list

The goal of this package is to compute the biomass of trees based on allometric equations developed by Lambert *et al* (2005) and Ung *et al* (2008). These equations were derived using data from Canadian trees.

The starting point is a *tree list*, where each tree has a species, diameter, and height (optional).

```
library(tibble)

tree_df <- tribble(
  ~spec, ~diam, ~ht,
  'Jack pine', 21, 15.2,
  'Trembling aspen', 32.1, 28,
  'Black spruce', 31, 22.6,
  'Black spruce', 21, 20.8,
  'Eucalyptus', 14, 10.2
)

tree_df
#> # A tibble: 5 x 3
#>   spec      diam      ht
#>   <chr>    <dbl> <dbl>
#> 1 Jack pine      21  15.2
#> 2 Trembling aspen 32.1  28
#> 3 Black spruce    31  22.6
#> 4 Black spruce    21  20.8
#> 5 Eucalyptus     14  10.2
```

The first three species appear in the database of tree species, as we can verify as follows:

```
library(biomasscan)
show_species()
#> [1] "All" "American beech"
#> [3] "Balsam fir" "Balsam poplar"
#> [5] "Basswood" "Black ash"
#> [7] "Black cherry" "Black spruce"
#> [9] "Conifers" "Deciduous"
#> [11] "Douglas fir" "Eastern hemlock"
#> [13] "Eastern white cedar" "Eastern white pine"
#> [15] "Engelmann spruce" "Grey birch"
#> [17] "Hickory" "Hop-Hornbeam"
#> [19] "Jack pine" "Largetooth aspen"
#> [21] "Lodgepole pine" "Pacific silver fir"
#> [23] "Red alder and Black cottonwood" "Red ash"
#> [25] "Red Maple" "Red oak"
#> [27] "Red pine" "Red spruce"
#> [29] "Silver maple" "Sitka spruce"
#> [31] "Subalpine fir" "Sugar maple"
#> [33] "Tamarack larch" "Trembling aspen"
#> [35] "Western hemlock" "Western red cedar"
```

```
#> [37] "White ash"           "White birch"
#> [39] "White elm"           "White oak"
#> [41] "White spruce"        "Yellow birch"
```

However, Eucalyptus does not appear. Nonetheless, we can still compute the biomass of the other four trees. The function `compute_biomass` takes four main arguments:

- **data**: The dataset (i.e. `data.frame`) containing the tree list.
- **species**: A character string giving the name of the column in `data` containing the species information.
- **diameter**: A character string giving the name of the column in `data` containing the diameter information. We assume the diameter is given in **centimeters**.
- **height**: A character string giving the name of the column in `data` containing the height information. We assume the height is given in **meters**.

```
compute_biomass(tree_df, species = 'spec', diameter = 'diam',
                 height = 'ht')
#> # A tibble: 5 x 8
#>   spec      diam    ht  Bark Branches Foliage  Wood Total
#>   <chr>      <dbl> <dbl> <dbl>    <dbl>    <dbl> <dbl> <dbl>
#> 1 Jack pine      21   15.2  9.29    14.2     7.51  101.  132.
#> 2 Trembling aspen 32.1   28   78.1    43.9     7.36  402.  531.
#> 3 Black spruce    31   22.6  34.3    32.8    20.0  282.  369.
#> 4 Black spruce    21   20.8  16.5    10.6     8.29  132.  167.
#> 5 Eucalyptus     14   10.2  NA      NA      NA    NA    NA
```

As we can see, we get a biomass estimate (in kilograms) for four different tree components, along with the total biomass. We can also see that no biomass estimate is given for the Eucalyptus tree.

Note that the height information is *optional*. If it is omitted, a different set of allometric equations are used.

```
compute_biomass(tree_df, species = 'spec', diameter = 'diam')
#> # A tibble: 5 x 8
#>   spec      diam    ht  Bark Branches Foliage  Wood Total
#>   <chr>      <dbl> <dbl> <dbl>    <dbl>    <dbl> <dbl> <dbl>
#> 1 Jack pine      21   15.2  10.1    12.3     7.52  121.  151.
#> 2 Trembling aspen 32.1   28   68.5    50.2     7.59  324.  450.
#> 3 Black spruce    31   22.6  33.5    36.2    21.6  267.  358.
#> 4 Black spruce    21   20.8  13.9    16.1    12.4  101.  143.
#> 5 Eucalyptus     14   10.2  NA      NA      NA    NA    NA
```

Different naming conventions

In order to allow for different naming conventions (e.g. “Black Spruce” vs “Spruce, Black”), we use fuzzy matching on the species name. Specifically, we compute the cosine distance between the species name in our database and the species name in the user’s dataset, and we select the allometric equation corresponding to the most similar species name.

Let’s look at an example of this.

```
library(tibble)

tree_df <- tribble(
  ~spec, ~diam, ~ht,
  'Pine, Jack', 21, 15.2,
  'Aspen, Trembling', 32.1, 28,
  'Spruce, Black', 31, 22.6,
  'Spruce, Black', 21, 20.8,
```

```

    'Cedar', 14, 10.2
)

tree_df
#> # A tibble: 5 x 3
#>   spec      diam    ht
#>   <chr>    <dbl> <dbl>
#> 1 Pine, Jack      21  15.2
#> 2 Aspen, Trembling 32.1  28
#> 3 Spruce, Black    31  22.6
#> 4 Spruce, Black    21  20.8
#> 5 Cedar          14  10.2

```

If we simply try to pass this dataset to `compute_biomass`, we will only get NAs.

```

compute_biomass(tree_df, species = 'spec', diameter = 'diam',
                height = 'ht')
#> # A tibble: 5 x 8
#>   spec      diam    ht  Bark Branches Foliage  Wood Total
#>   <chr>    <dbl> <dbl> <dbl>    <dbl>    <dbl> <dbl> <dbl>
#> 1 Pine, Jack      21  15.2    NA      NA      NA      NA    NA
#> 2 Aspen, Trembling 32.1  28      NA      NA      NA      NA    NA
#> 3 Spruce, Black    31  22.6    NA      NA      NA      NA    NA
#> 4 Spruce, Black    21  20.8    NA      NA      NA      NA    NA
#> 5 Cedar          14  10.2    NA      NA      NA      NA    NA

```

Therefore, we first need to identify the appropriate matches. To do this, we can use the function `match_species_name`:

```

match_species_name(tree_df, species = 'spec')
#>           spec      biomasscan
#> 1 Pine, Jack      Jack pine
#> 2 Aspen, Trembling Trembling aspen
#> 3 Spruce, Black    Black spruce
#> 4 Cedar          Cedar Western red cedar

```

We can see that there are actually two types of cedar in our database, and the one selected by `match_species_name` is Western Red Cedar. If we prefer the Eastern Cedar, we can remove any ambiguity by renaming Cedar in the `tree_df`.

```

library(dplyr)

tree_df <- mutate(tree_df,
                  spec = if_else(spec == 'Cedar',
                                'Cedar, Eastern',
                                spec))

match_species_name(tree_df, species = 'spec')
#>           spec      biomasscan
#> 1 Pine, Jack      Jack pine
#> 2 Aspen, Trembling Trembling aspen
#> 3 Spruce, Black    Black spruce
#> 4 Cedar, Eastern Eastern white cedar

```

Once we are satisfied that the function `match_species_name` gives us the correct match, we can use the output to augment our dataset and pass it to `compute_biomass`.

```

matches_df <- match_species_name(tree_df, species = 'spec')

tree_df |>
  left_join(matches_df, by = 'spec') |>
  compute_biomass(species = 'biomasscan', diameter = 'diam',
                  height = 'ht')

#> # A tibble: 5 x 9
#>   spec          diam    ht biomasscan    Bark Branches Foliage  Wood Total
#>   <chr>          <dbl> <dbl> <chr>          <dbl>    <dbl>    <dbl> <dbl> <dbl>
#> 1 Pine, Jack      21    15.2 Jack pine      9.29     14.2     7.51 101. 132.
#> 2 Aspen, Trembling 32.1    28 Trembling asp~ 78.1     43.9     7.36 402. 531.
#> 3 Spruce, Black    31    22.6 Black spruce  34.3     32.8     20.0 282. 369.
#> 4 Spruce, Black    21    20.8 Black spruce  16.5     10.6     8.29 132. 167.
#> 5 Cedar, Eastern   14    10.2 Eastern white~ 3.52     5.00     4.52 23.6 36.6

```

One important note: the function `match_species_name` will *always* find a match, so it is important to filter out the trees that simply do not appear in the database of allometric equations:

```

tribble(
  ~spec, ~diam, ~ht,
  'Jack pine', 21, 15.2,
  'Trembling aspen', 32.1, 28,
  'Black spruce', 31, 22.6,
  'Black spruce', 21, 20.8,
  'Eucalyptus', 14, 10.2
) |>
  match_species_name(species = 'spec')

#>           spec      biomasscan
#> 1 Jack pine      Jack pine
#> 2 Trembling aspen Trembling aspen
#> 3 Black spruce    Black spruce
#> 4 Eucalyptus      Black spruce

```