Documentation for the Canopy Constructor algorithm, v.1.0 (30/06/2020)

Main functions of the Canopy Constructor algorithm:

The Canopy Constructor is an algorithm to create individual-based 3D representations of empirical canopies from field inventories and canopy height models (CHMs). A detailed description will be available in a companion paper (currently in review).

At its core, the Canopy Constructor has two main steps:

- Step1: Creation of local best-fit forest canopies and summary statistics of forest structure (local inference/calibration step) based on allometric relationships between tree dimensions
- Step 2: Utilization of local crown packing densities together with canopy height models to infer tree positions and sizes in non-field measured plots (extrapolation step)

In addition, there are several, currently experimental, options:

- Step 0: Creation of synthetic canopies, either from actual field inventories or from more general diameter distributions plus allometric equations
- addition of physiological constraints (i.e. viability of the individual trees based on carbon balance)

Compilation of C++ code:

Requirements

- GNU scientific library (gsl) v. 1.7 or higher. This manual has been produced with gsl v.2.6. On OSX, for example, install with `brew install gsl`

Command line compilation:

- `g++/clang/icpc main.cpp –o CanopyConstructor.out –O3 –lgsl –lgslcblas`

Xcode IDE (v.11.3.1):

- add linker flags, as in Figure 1
- add optimization level: Fastest [-O3]

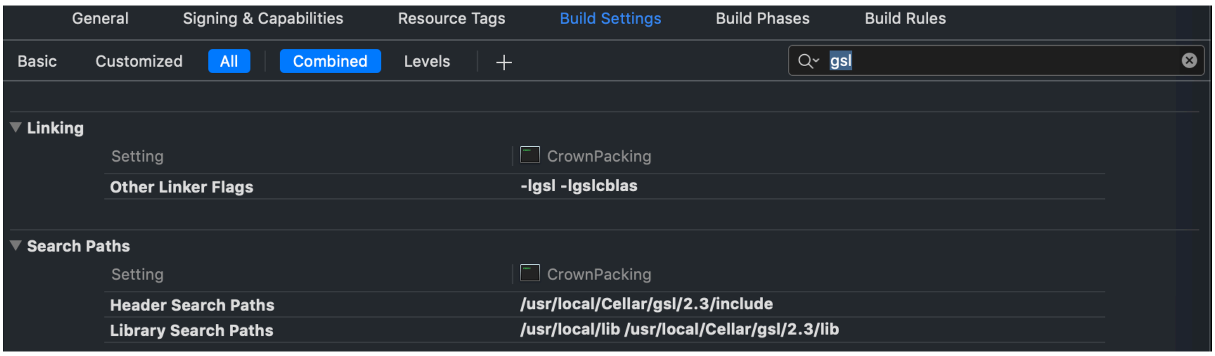

Figure 1: Example of build settings in Xcode 11.3.1

Execution

Command line:

- basic requirement: `./CanopyConstructor.out –o/'CC'`
- the *-o* option determines output file names and is always required
- additional options for input streams cf. Table 1

Flag	Input stream	Description	Required	Throws error
-g	general/ global parameters	The most basic parameters to regulate the Canopy Constructor algorithm and that generally won't need calibration. E.g., maximum spatial extent of the canopy and fitting/output options.	No. If not provided or only partially, default parameters are assumed. Attention: Impossible parameterizations are overwritten.	If file is provided, but cannot be read
-d	detailed parameters	Parameters that determine the shape of the reconstruction and may be explicitly calibrated. E.g., diameter distribution, allometric parameters. Also accepts several parameter sets to test sequentially.	No. If not provided or only partially, default parameters are assumed. Attention: Impossible parameterizations are overwritten.	If file is provided, but cannot be read.
-f	forest/field inventory	Provides a local set of tree diameter measurements (dbh) with x/y positions and wood density values, with optional information about other tree traits.	No. If not provided, trees will be drawn from distributions given in detailed parameters.	If file is provided, but cannot be read. If dbh is not provided or outside of allowed range (> dbh_limittop or < dbh_limitbottom, given in detailed parameters)
-c	chm	The local CHM with the same x/y coordinates as the inventory	No. If not provided, the algorithm defaults to creation of a random canopy (no spatial fitting, Step 0)	If file is provided, but cannot be read. If x-y coordinates or height is outside of range given in general parameters.
-v	crown packing density/volume	The crown packing density matrix, i.e. average crown volume per within-canopy height layer, per maximum canopy height.	No. If not provided, the algorithm defaults to the creation of either a random canopy (Step 0) or a spatial fit (Step 1), but does not allow for Step 2 inference	If file is provided, but cannot be read. If packing densities and heights are outside the potential range.
-m	meteorological/climate input	(experimental) The climate input file, as provided to the TROLL model	No. If not provided, the algorithm overwrites any options that would optimize the trees' carbon balance	If file is provided, but cannot be read.

Table 1: Additional options for the Canopy Constructor algorithm.

Example runs with simulated data:

In the following, we generate a synthetic data set with known tree dimensions (Step 0). We then discard the information on tree heights and crown dimensions and use it as the empirical data set, for which tree dimensions need to be inferred (Step 1). We then use the packing density matrix of the fitted forest to reinfer the underlying stem diameter distribution (Step 2). All plotting was done with the ggplot package (Wickham, 2011).

Step 0: Simulated data

First, we compile the code:

```
g++ main.cpp -O3 -o CanopyConstructor.out -lgsl -lgslcblas -Wall
```

and execute the following lines:

```
./CanopyConstructor.out -o'CC0'
```

This will create a synthetic data set over 16ha with automatically generated input files for the algorithm. The files `CC0_input_general.txt` and `CC0_input_detailed.txt` contain the default parameters. The file `CC0_input_inventorycutoff.txt` contains all tree positions and diameter measurements >=10cm dbh. `CC0_input_CHM.txt` contains the generated canopy height model which should be similar to Figure 2.

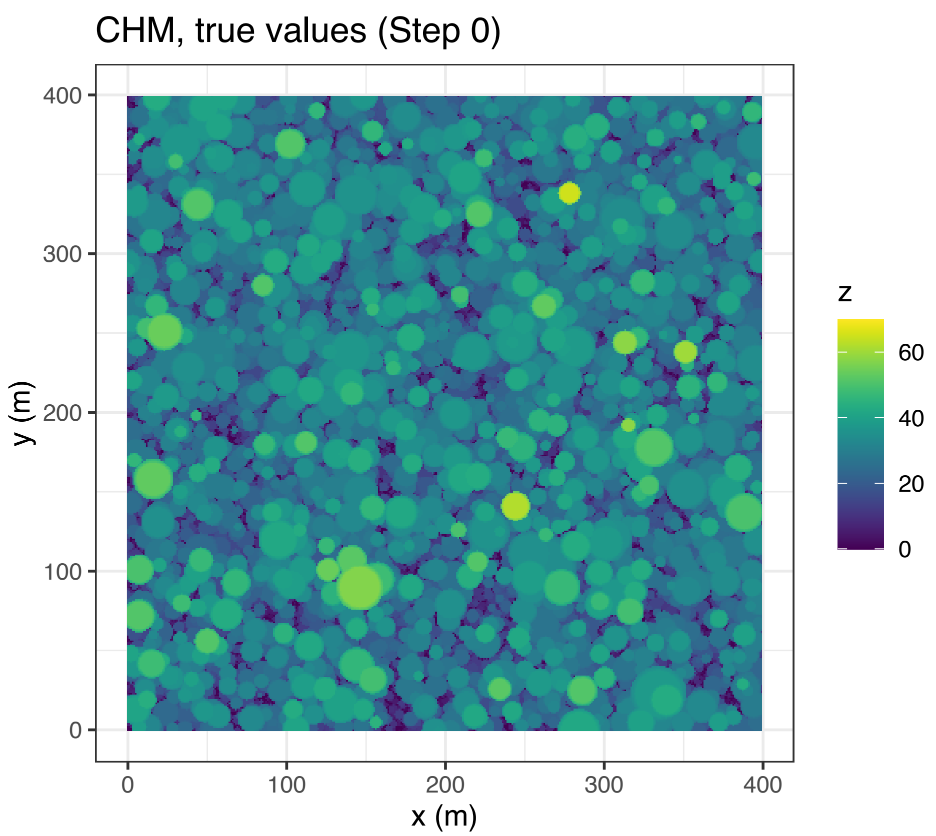

Figure 2: Generated canopy height model from random tree positioning.

Step 1: Local inference of forest structure

We now use the generated files as input to test step 1 of the Canopy Constructor.

```
    ./CanopyConstructor.out -o'CC1' -g'CC0_input_global.txt' -
d'CC0_input_detailed.txt' -f'CC0_input_inventorycutoff.txt' -
c'CC0_input_CHM.txt'
```

This creates a canopy with all tree dimensions inferred only from the diameters >= 10cm and the canopy height model. Trees < 10cm are supplemented. Again, we obtain the four output files (should be identical to the previous output files). For the next step, we will also require the newly created `CC1_input_cp.txt`, which summarizes crown packing densities. Please note that we output to CC1, not CC0. The file `CC1_input_CHM.txt` contains the newly fitted canopy height model and should be similar to Figure 3 and also similar to Figure 2, since it is fitted to it.

Note: Since we did not impose any restrictions on crown overlap in the creation of the synthetic data set, the option `flag_PreventCrownpiercing` is deactivated. For real forest stands, it is typically activated to prevent oversegmentation.

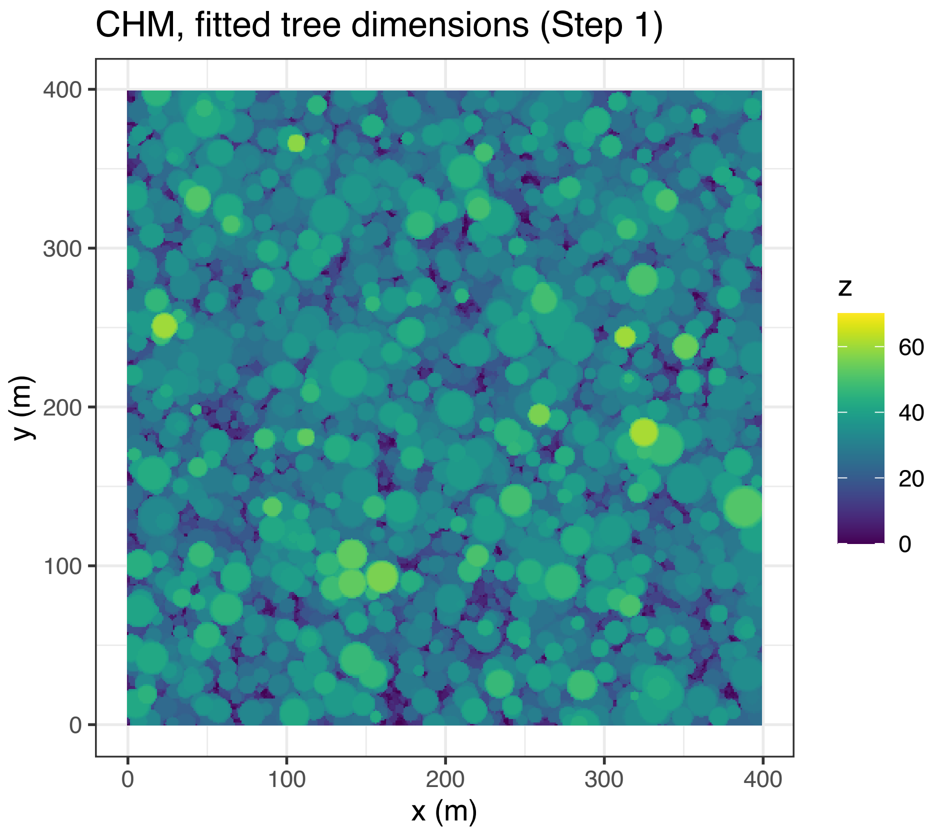

Figure 3: Generated canopy height model, from fitting simulated inventory into simulated canopy height model.

Step 2: Extrapolation model

We now use the input files from step 0 as well as the crown packing file from Step 1 as input to test the Canopy Constructor procedure's second step:

```
      ./CanopyConstructor.out -o'CC2' -g'CC0_input_global.txt' -
d'CC0_input_detailed.txt' -f'CC0_input_inventorycutoff.txt' -
c'CC0_input_CHM.txt' -v'CC1_input_cpcutoff.txt'
```

Again, this should result in a similar canopy height model. Cf. Figure 4.

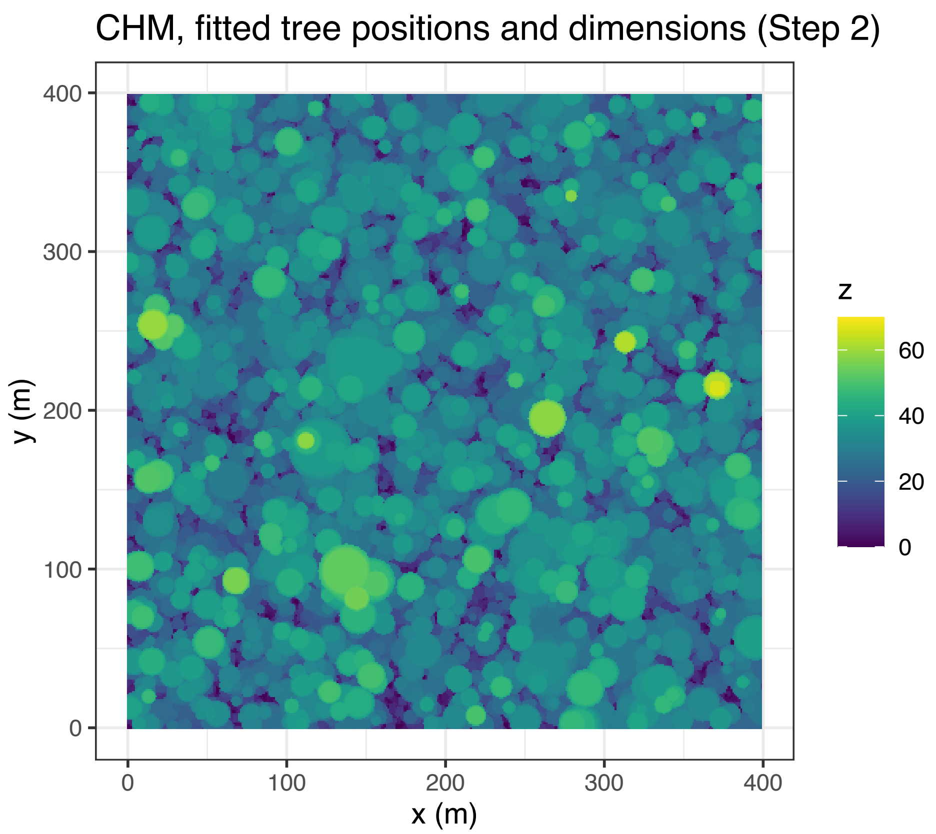

Figure 4: Canopy height model from Step 2. Based on inferring the underlying diameter distribution of the simulated canopy from crown packing densities.

Overall approach:

The approach can, for example be used for the inference of allometric parameters. Different allometries could be used in Step 1, and then the capability of the Canopy Constructor to reproduce the original, "real" allometry from Step 0 could be assessed. The uncertainty could also be propagated to simulations in Step 1. [The manual will be updated in the future to describe this in more detail].

Parameters

Global:

- **cols:** extent in x direction
- **rows**: extent in y direction
- **height_max:** the maximum height extent of the simulated scene
- **seed_rng**: if >= 0, the seed for the random number generator, otherwise randomly chosen
- **dbh_threshold_stop**: (currently not used) the tree diameter for which successful fitting is observed
- **threshold_stop**: (currently not used) the threshold when to stop fitting across all steps; fraction of successful fits for trees >= dbh_threshold_stop
- **nbsteps_dissimilarity**: a fixed number of steps for reducing canopy dissimilarity; overrides threshold_stop
- **nbsteps_mae:** a fixed number of steps for reducing mean absolute error; overrides threshold_stop
- **nbsteps_physiology:** a fixed number of steps for minimizing the number of trees with negative carbon balance; overrides threshold_stop
- **nbsteps_combined:** final number of steps for a combined metric; overrides threshold_stop
- **dbh_limitbottom:** minimum tree diameter; if a field inventory is provided with a cutoff above dbh_limitbottom, then trees are filled up
- **dbh_limittop:** the maximum tree diameter
- **dbhlog_binsize:** binsize of trees on logscales
- **crowndisplacement_factor:** distance a crown center can be displaced as fraction of tree height, will automatically be limited between 0 and the crown diameter; defaults to zero
- **flag_ApplyMedianfilter:** only if CHM is provided: if == 1, applies a median filter to the CHM to remove small canopy gaps
- **flag_Prefitting:** if == 1, introduces a prefitting procedure to improve convergence for step 2 of the Canopy Constructor algorithm
- **flag_PreventCrownpiercing:** if == 1, introduces condition to prevent slim tree crowns from piercing wide tree crowns
- **flag_OutputReduced**: if == 1, minimizes output size, only outputs summary statistics, for example for ABC fitting
- **flag_OutputTROLL:** if == 1, an input sheet for the TROLL IBM is created, as well as an initial inventory to initialize TROLL from

Detail:

- **paramID:** this is the unique identifier of the current parameter combination
- **a_sdd:** either (log-log) slope of a power law (when negative) or the shape parameter of a 2-parameter Weibull distribution from which trees are drawn (sdd = "stem diameter distribution"); overridden by field inventory input, but used to fill up trees if inventory does not extend to "dbh_limitbottom"
- **b_sdd:** either the inverse of the rate of an exponential function (when negative) into which the power law transitions, or the scale parameter of the 2-parameter

Weibull distribution; overridden by field inventory input; the transition point between power law and exponential function is automatically determined by finding the point where the log-log slope of the exponential function equals the log-log slope of the power law. This is simply: a_sdd/rate

- **nbavg_sdd:** the maximum number of trees to draw from the distribution; overridden by field inventory input,
- **a_height:** hmax parameter of the Michaelis-Menten height allometry
- **b_height:** ah parameter of the Michaelis-Menten height allometry
- **sigma_height**: variation around the allometric mean
- **hetscedas_height:** the fraction of variance at dbh_limitbottom (sigma_height) retained at dbh_limittop
- **a_CR:** log-intercept of the power law crown radius allometry
- **b_CR:** log-slope of the power law crown radius allometry
- **sigma_CR:** variation around the allometric mean
- **hetscedas_CR:** the fraction of variance at dbh_limitbottom (sigma_CR) retained at dbh_limittop
- **a_CD:** intercept of the linear relationship between tree height and crown depth
- **b_CD:** slope of the linear relationship between tree height and crown depth
- **sigma_CD:** variation around the allometric mean
- **hetscedas_CD:** the fraction of variance retained at the maximum height
- **LMA:** leaf mass per area (g/m^2)
- **sigma_LMA:** intraspecific variation in LMA (g/m^2)
- **Nmass:** leaf nitrogen on a per mass basis (g/g)
- **sigma_Nmass**: intraspecific variation in Nmass (g/g)
- **Pmass:** leaf phosphorus on a per mass basis (g/g)
- **sigma_Pmass:** intraspecific variation in Pmass (g/g)
- **wsg:** wood specific gravity (g/g)
- **sigma_wsg:** intraspecific variation in wsg (g/g)
- **corr_Nmass_Pmass:** correlation coefficient between Nmass and Pmass
- **corr_Nmass_LMA:** correlation coefficient between Nmass and LMA
- **corr_Pmass_LMA:** correlation coefficient between Pmass and LMA
- **shape_crown:** the shape parameter for crowns, defined as the percentage of radius at the crown top compared to the crown bottom; in the limit of 1.0 cylindris, in the limit of 0.0 cone
- **gapfraction_crown**: fraction of gaps in the average crown (only used when physiological module is activated)

Bibliography

Wickham, H., 2011. ggplot2. Wiley Interdiscip. Rev. Comput. Stat.

https://doi.org/10.1002/wics.147