

AutoTrees v1.1 and v2 are closely intertwined, with v2 essentially being v1.1 with a driver script; as such, this overview will address both versions and indicate where there are differences to be considered.

1. Project Goals

The purpose of AutoTrees is to take a .db file of tree data and automate the process of creating an animation of the trees growing over time, using Blender. R is used to unpack the .db file into a .txt file, to be read by a Python script run in Blender. V2 runs HandBrake to convert the Blender .avi into a .mp4 file, to conserve disk space and ensure a playable animation (AutoTrees tends to create large .avis, which may not be playable without conversion). V1.1 recommends HandBrake but it is not a requirement.

The R script takes the .db file and queries it for specific headings, splicing together the resulting matrices into a dataframe to preserve typing and then outputting it into a .txt file.

The Python script for Blender reads the .txt file, assigns colors to each species found in the data, and then creates and edits tree models year by year, places the models using a grid coordinate system, moves the camera on an orbit around the grid, establishes a basic environment (lighting, sky, and ground) and keyframes as necessary throughout, before exporting both an .avi file and a .txt file with the RGB values used for each species. V1.1 does not automatically scale the number of grid coordinates with the amount of models needed, if more spaces are needed the user must adjust variables accordingly. V2 automatically scales and no adjustments should be necessary to accommodate large volumes of models. Both versions scale model size based on the total number of grid coordinates.

The Python driver script for v2 is run in command line and uses Python 3. It queries the user for the name of their .db file and checks for the file before continuing (requiring that the file is in the same folder). It then uses the subcommand library to perform all necessary steps in the command line, including running HandBrake, and then issues a command to delete the Blender .avi as clean-up. (.avis created with this program grow large quickly, both taking up space and frequently being unplayable without conversion; therefore after conversion the .avi is both unnecessary and potentially taking up gigabytes of storage space, so removal is optimal.)

Workflow in v1.1 requires the user to run an R script, then open Blender and run a Python script, and then optionally run HandBrake on the resulting .avi. The user must tweak variables within the R and Python scripts before running to ensure correct paths.

Workflow in v2 requires the user to run a Python script that acts as a driver, running the R script, then using Blender to run another Python script, and finally running HandBrake on the .avi output by Blender. The average user should not need to change anything in the scripts, unless they are on Windows and not using R version 3.6.0.

2. Testing Environments

AutoTrees v1 and v2 were created tested on the following environments:

Operating System: Mac

OS version: macOS High Sierra 10.13.6

RAM: 16.0 GB

CPU: 2.2 GHz Intel Core i7

Python version: Python 3.7.3

Python libraries:

- v1: random, math; bpy via Blender version 2.79b
- v2: os, sys, random, math, functools, pathlib, subprocess; bpy from Blender version 2.79b

R version: R 3.6.0

R libraries: DBI, RSQLite

Blender version: 2.79b

HandBrake version: 1.2.2

File names (v1): animation_v1.1.py (python), unpack_v1.1.R (R), Trees.blend (Blender)

File names (v2): AutoTrees_v2.py (python), animation_v2.py (python), unpack_v2.R (R), Trees.blend (Blender)

Directory setup: All files in the same directory. Explicitly necessary for AutoTrees v2.

Dataset 1: 0-25 years, 25 year increment,

- Runs in approx 10 minutes, Blender outputs a 225 MB .avi

Dataset 2: 0-200 years, 25 year increment, maximum 4646 trees per year

- Runs in approx 1.5 hours, Blender outputs a 963 MB .avi

Operating System: Windows

OS version: Windows 8.1

RAM: 8.0 GB

CPU: AMD FX-6300 Six-Core Processor @ 3.50 GHz

Python version: Python 3.7.3

Python libraries:

- v1: random, math; bpy via Blender version 2.79b
- v2: os, sys, random, math, functools, pathlib, subprocess; bpy from Blender version 2.79b

R version: R 3.6.0

R libraries: DBI, RSQLite

Blender version: 2.79b

HandBrake version: 1.2.2

File names (v1): animation_v1.1.py (python), unpack_v1.1.R (R), Trees.blend (Blender)

File names (v2): AutoTrees_v2.py (python), animation_v2.py (python), unpack_v2.R (R), Trees.blend (Blender)

Directory setup: All files in the same directory. Explicitly necessary for AutoTrees v2.

Dataset 1: 0-200 years, 25 year increment, maximum 4646 trees per year

- Runs in approx 2 hours, Blender outputs a 963 MB .avi

Operating System: Windows

OS version: Windows 10 Enterprise, version 1703, build 15063.1868

RAM: 16.0 GB

CPU: Intel Core i7-6700 CPU @ 3.40GHz

Python version: Python 3.7.3

Python libraries:

- v1: random, math; bpy via Blender version 2.79b
- v2: os, sys, random, math, functools, pathlib, subprocess; bpy from Blender version 2.79b

R version: R 3.4.3

R libraries: DBI, RSQLite

Blender version: 2.79b

HandBrake version: 1.2.2

File names (v1): animation_v1.1.py (python), unpack_v1.1.R (R), Trees.blend (Blender)

File names (v2): AutoTrees_v2.py (python), animation_v2.py (python), unpack_v2.R (R), Trees.blend (Blender)

Directory setup: All files in the same directory. Explicitly necessary for AutoTrees v2.

Dataset 1: 0-25 years, 25 year increment,

- Runs in approx 10 minutes, Blender outputs a 225 MB .avi

Dataset 2: 0-200 years, 25 year increment, maximum 4646 trees per year

- Runs in approx 2 hours, Blender outputs a 963 MB .avi

3. Possible Points of Stress or Failure

- R script
 - .db files must have a table named cohorts with column headings cohortID, year, species, trees, diameter, and height, unless R script is edited to accommodate.
 - Blender script expects the order of data columns to remain the same, i.e. if you use different column headings, all data must still be represented, and must be pulled by the R script's queries in the order of cohortID, year, species, trees, diameter, and height.
 - V2 requires that the R script be fed two arguments, as specified in AutoTrees_v2.py's comments. If not enough arguments are given, or they are in the wrong order or an incorrect path/name, the script will fail.
- Blender Python script

- V2 requires that there exists a .txt file named “matrix_file_nr.txt” in the same folder as the project. It will fail if there is not when the line `db_file = open(os.path.dirname(__file__)+'/matrix_file_nr.txt', 'r')` is called .
 - The v2 R script will name its output this by default, but if the script fails or is changed, it is possible that no file will be created or it will be under a different name.
 - V1.1 requires that the path to the .txt file created by the R script be manually entered, and if it is not, it will fail on a similar line: `db_file = open(path, 'r')`.
- V1.1 requires that the variable `maxcoord` is set such that `maxcoord2*4` is greater than the maximum number of trees represented within a single year interval. Otherwise, it will fail at the line `rand = random.randrange(0, (len(coordlist)))` when it tries to get a coordinate for the next tree after all coordinates are used.
 - Default `maxcoord` is set to accommodate up to 10,000 trees at one time.
 - V2 should not have this problem, as it finds the maximum number of coordinates needed and automatically adjusts `maxcoord` accordingly.
- V1.1 requires the user sets `yearjump` manually. If it is set to the wrong number, a failure may occur when `hold['year'][now]` is called for comparisons within the main loop; there are three possible lines that would call it, depending on other comparisons.
 - Note that an incorrect value for `yearjump` should throw an error, but if the incorrect value is a multiple of the correct value, the program may have a small chance to run without errors but produce an incorrect animation. Please always ensure that v1.1's `yearjump` is set correctly each time `animation_v1.1.py` is run.
- Both versions have a maximum of 24 colors available, and the default versions will fail if there are more than 24 species in a dataset when the line `c = mydict['colors'].pop()` is called.
 - The User Manuals for both versions describe how to get around this issue; more colors must be added to the `colors` list.
- A failure at `bpy.ops.wm.append(filepath=filepath, filename=filename, directory=directory)` indicates that there is an issue with the model variables `blendfile`, `section`, and/or `obj` in the Global Variables heading.
 - If you have not changed `blendfile` to reflect a custom .blend file for your model, check that the path to `Trees.blend` is correct, and that `Trees.blend` is present (particularly with v2, which requires that it is in the same folder).
 - If you have changed `blendfile` to reflect a custom .blend file for your model, please see the [v1.1 User Manual](#) Addendum for a walkthrough of how to check the necessary settings for `section` and `obj`.
- A failure at `bpy.data.scenes['Scene'].render.filepath = renderpath` indicates that there is an issue with the variable `renderpath` under Global

Variables. Check that the path is correct and points to a directory that you have permissions to write to.

- V2 will point to a file name, not a directory. By default it should be a path to a not-yet created file within the directory that the project files are located in, i.e. it will attempt to create the Blender .avi in the same directory as the AutoTrees files. Make sure that this directory can be written to.
- Driver script for v2
 - The driver expects R version 3.6.0 and will need to be edited if you are running Windows and using a different version; the script will fail when it attempts to call RScript.
 - Windows: The driver expects R and Blender to be installed in the Program Files and HandBrakeCLI to be moved into the Program Files after download. The script will fail when called if they are not in those places.
 - Mac: The driver expects Blender to be in the Applications folder, and may have path issues for older versions of Blender (which include version in their file names)
 - Mac: The driver expects RScript and HandBrakeCLI to be recognized as commands. This should be default when R and HandBrake are installed via Homebrew (instructions are provided in the [v2 User Manual](#)), it is unknown if they are default when installed by hand. Installation via Homebrew is strongly recommended.
 - The driver expects all program files and all created files to be within the same directory as it. Paths will throw errors and the script will fail if any of the expected files are not within the same directory. Additionally, there is a chance of the R and Blender Python scripts failing if the paths are edited or files are moved.
- HandBrake has been observed to throw an “Encode failed <error 3>” on Windows in some instances of final.mp4 existing within the directory before HandBrake is called to create it; when this error is thrown, final.mp4 is not overwritten with the new animation. The reason for this is unknown, as the issue seems to occur inconsistently on Windows and has not been observed on Mac. For safety, the User Manuals instruct users on both operating systems to move final.mp4 to another directory (or delete the file) before re-running AutoTrees_v2.py. If you run into this error, please try moving the existing final.mp4 out of the AutoTrees directory.