AvatolCV v1.0

Contents

[Introduction 1](#_Toc451184198)

[Dependencies 1](#_Toc451184199)

[Configuring Python 2](#_Toc451184200)

[Getting the Installer 2](#_Toc451184201)

[Installing AvatolCV 3](#_Toc451184202)

[Updating AvatolCV 3](#_Toc451184203)

[Running AvatolCV 3](#_Toc451184204)

[An AvatolCV Session 4](#_Toc451184205)

[Data Sources 4](#_Toc451184206)

[Log in to Data Source 4](#_Toc451184207)

[What the Installer Does In Detail 4](#_Toc451184208)

# Introduction

AvatolCV is a system that supports running computer vision and machine learning algorithms on images of biological specimens. The goal is to automate scoring biological characters, leveraging training data provided by the biologist.

AvatolCV is integrated via web services with two web-based biological image repositories:

Morphobank (morphobank.org) – a taxon vs character matrix based system

BisQue (<http://bisque.iplantcollaborative.org/client_service>) – an image centric repository for plant images

# Dependencies

64 bit Windows / 64 bit OSX (Mac)

python 2.7.x

Java 8 64 bit

MATLAB R2015b

8G RAM

# Configuring Python

Your system may come with python 2.7.x pre-installed.

To check on Mac, open up a terminal (Finder->Applications->Utilities->Terminal) and type "python --version". If present you will see something like : ($ is the Mac prompt)

$python --version

Python 2.7.10

If not present, install it from https://www.python.org/downloads/release/python-2711/. If "python --version" does not yield an answer, you will need to set the PATH directory to contain the directory with the python executable using this command:

$export PATH=$PATH:<location where python was installed>

for example

$export PATH=$PATH:/usr/bin/python

To check on a Windows machine, type "python --version"

C:\>python --version

Python 2.7

If not present, install it from https://www.python.org/downloads/release/python-2711/. If "python --version" does not yield an answer, you will need to set the PATH directory to contain the directory with the python executable. (Control Panel -> System and Security -> System -> Advanced System Settings -> Environment Variables. In the System Variables window, select Path and click edit. Click once in the Variable Value field to ensure it is no longer highlighted. Right arrow to the end and type a semicolon followed by the path to where python was installed, for example if python was installed at C:\, you would append this to the path: ";C:\Python27". Then click OK, click OK, click OK. Now open up a new command shell and "python --version" should yield "Python 2.7)

# Getting the Installer

1. Got to https://github.com/AVATOL/AvatolCVInstaller

2. Click the button that says "Download ZIP"

3. unzip the file and then proceed with the instructions in the next section

# Installing AvatolCV

To run it, open up a command shell on Mac and type:

$cd whereTheFileWasUnzippedTo

$python updateAvatolCV.py install\_root

or on Windows

C:>cd whereTheFileWasUnzippedTo

C:\whereTheFileWasUnzippedTo>python updateAvatolCV.py install\_root

...where install\_root is replaced by your choice of what directory you want to install into. The installer will create a subdirectory called 'avatol\_cv' and place the system under that. So, for example, you specify

C:\whereTheFileWasUnzippedTo>python updateAvatolCV.py C:\someDir

...then c:\someDir\avatol\_cv will contain all the files for AvatolCV.

# Updating AvatolCV

If there is a bug fix and you need to pull in the updated system, merely run the same command that you used to install it initially.

python updateAvatolCV.py install\_root

# Running AvatolCV

Once the files are in place, you would do:

cd C:\someDir\avatol\_cv\java\lib

java -jar avatol\_cv.jar

...to start the system

or on Mac

cd /someDir/avatol\_cv/java/lib

java -jar avatol\_cv.jar

# An AvatolCV Session

## Data Sources

## Log in to Data Source

# What the Installer Does In Detail

AvatolCV is broken up into the following bundles:

docs

modules\_3rdParty

modules\_osu

java

usage: python updateAvatolCV.py <installRoot>

Here is the process updateAvatolCV.py uses:

1. ensures specified installation root dir exists

2. creates an avatol\_cv dir under that installation root dir

3. determines a platform code - what system we are running on : win | mac | unsupported

4. generates the name of the downloadManifest file that it needs to download (from http://web.engr.oregonstate.edu/~irvineje/AvatolCV/) using:

downloadManifest\_<platform\_code>.txt

5. generates a temporary name that it will save that file under

downloadManifest\_<platform\_code>\_new.txt

6. downloads that file and saves as that name

new\_manifest\_pathname = downloadFile(avatolcv\_root, new\_manifest\_filename, manifest\_truename)

7. if this is not the first installation maneuver, the prior maneuver would have saved the file as...

downloadManifest\_<platform\_code>\_old.txt

...so we look to see if that file exists

8. If that old file exists we dump it to the console, then dump the new one to the console for sanity comparison by user

9. We determine which bundle names need downloading by comparing old and new.

- If no old file exists, everything in new will be pulled.

- If any bundle name represented in new is not represented in old, it is pulled

- otherwise, we compare the datestamp (i.e. the version) associated with new to see if it is different than old. If so, we download.

Note that we don't check for a more recent version, just difference with prior file. This will simplify regressing back to prior version if need be.

If no bundles are needed, we declare we are up to date and exit, otherwise we continue

10. foreach bundle we need to download, we first uninstall the prior-installed version of that bundle using the

allFiles\_<bundle\_name>.txt

file that came with each bundle as a guide for which files to delete.

11. we delete the old manifest file and rename the new one to

downloadManifest\_<platform\_code>\_old.txt

...so it can be cheacked at the later download

12. now we install each bundle by looking up the bundle filename from the downloadManifest and pulling from the distro site.

This yields a .tgz bundle that we then unwrap underneath the avatolcv dir