ARTHROPOD DIVERSITY INVENTORY WEBSITE
USER MANUAL
PYOINV 1.0

Uriel Garcilazo-Cruz & Fernando Álvarez-Padilla

This program represents a python-based version of the software BIOINV (Álvarez-Padilla et al. 2020). PYOINV 1.0 consolidates all the individual steps that the user had to do in the previous versions of BIOINV. It is written in Python and consists of a single platform that the program runs directly from the command line of Windows or other OS. If you have comments or need more information, please contact the second author at: fap@ciencias.unam.mx.

PYTHON SOFTWARE

PYOINV 1.0 https://github.com/UGarCil/PYOINV.git was created using python 3.7 and Anaconda 4.8.3. We strongly recommend using Anaconda to run PYOINV 1.0 as the newest versions install all the dependencies and libraries needed to run the program. To execute main.py you will need to have installed the following libraries:

- pandas
- PIL (also known as pillow)
- shutil
- logging
- random

We suggest running PYOINV 1.0 from your OS command terminal or directly executing the main.py module. To do this Anaconda will need to be added into your PATH variables. This option is offered to you ONLY if Anaconda is installed as Administrator.

Further instructions in how to install Anaconda and add the path are described in this web page and video: https://www.kdnuggets.com/2020/02/install-python-anaconda-windows.html.

You can find additional information in the README.md file of the github repository.

PYOINV 1.0 SOFTWARE

To run correctly PYOINV 1.0 must have the following folders, python modules and html files.

1) html_templates: this folder includes the html templates for the species and the menu pages (speciesTemplate.html and template.html respectively). The file styles.css controls the appearance of all pages regarding font style, font color, size of thumbnails, etc. Changes in these templates will be inherited to all pages.

2) Images: this folder includes the high-resolution images. The internal structure of the folder is arbitrary; however, we suggest three layers for family, genus and species names because it allows further additions to be included in a more systematic way. All images can also be stored directly inside the folder Images without any subdirectories, but this will be confusing for the users while appending and locating the images.
3) Python files: **main.py** coordinates all other python modules. **CloneDirectories.py** makes the thumbnails, **evalError.py** makes the CriticalErrors.txt file, and **populateHtmls.py** creates the individual menu and species pages. They can be modified, but the user needs to be experienced with the python language and IDLE.
4) Html files: index.html will call all required pages to run the website, blank_page.html will fill the empty space where the species menu will appear, and welcome.html is the presentation page for the inventory.
5) Excel files: **SPPDATA.xlsx** that contains the taxonomic information of each species provided by the user, and **structureNomenclature.xlsx** includes the names of the structures and their orientation for the standard views with the acronyms in the species images' names. The **SPPDATA.xlsx** or taxonomic table is organized in 13 columns summarized in Table 1.

Table 1. Taxonomic information included in the SPPDATA.xlsx database.

COLUMN NAME	DESCRIPTION
FAMILY	Necessary to create the main access menu for the website, but the taxonomic rank is optional.
OLD_SPP_NAME	Only necessary for the user book keeping.
UPDATED_SPP_NAME	Name of the species html and current taxonomic id, shown on website.
MSP_CODE	Most important relational element that links all database.
WSP_NUM	This catalog number links the current taxonomic id with its nomenclatural changes.
SP_AUTHOR	This column contains the species authorship only for described species.
ID_AUTHOR	Person in the inventory whom id the specimens to their current taxonomic rank.
FEMNUM	Number of female specimens
MALNUM	Number of male specimens
TAXON NOTES	Field for optional taxonomic information.
SPPIMAUT	Person in the inventory that photographed the voucher specimens
LOCALITY	This field includes the locality and collecting information (date, method, etc.) only for the voucher specimens. The data for the other material is included in the inventories' main database.
VOUCOD	Optional column containing the specimens' unique image codes.

PYIONV 1.0 creates the following files and folders:

1) Thumbnails: this folder is a clone of the Images folder structure and contains the low-resolution images. This step was implemented to maximize computational resources when loading the page.

2) HTML_files. This folder includes all the access menus and species pages. The Species sub folder includes the species pages that incorporate the taxonomic data and the images from the thumbnails with hyperlinks to their original high-resolution versions. The Menus sub folder includes the Family and species access menus.
3) __pycache__: This folder includes the cache files that python uses. It can be erased after the program finishes running and shouldn't be uploaded into the server.
4) Esquema.xlsx: This file is an Excel table that summarizes the relationship between each of your species' images and their taxonomic information. The program uses the two Excel files mentioned above.
5) Summary_images_processed.txt. This file presents a list of all the images processed, the species that they belong, their respective standard views footnotes and the addresses for the high-resolution images.
6) CriticalErrors.txt. This file present to the user the file names with errors or discrepancies between the taxonomic and nomenclature tables. It indicates the error found and the corresponding image.
7) Creates a log.txt. This file presents all the individual steps that the program main.py does.

EXAMPLE WEBSITE

The example data set has the same rules mentioned above and it applies the program PYOINV 1.0 to 67 standard views for six species, one symphytognathid and five theridiosomathids. We recommend for the first time using this example to run the program. It should run without any problems given that all the image names and their corresponding information in the taxonomic and nomenclature tables are correct.

The first time main.py is executed it takes some time to react on the command interface, but give it a minute and you will see the progress lines while creating the thumbnails and at the end the html pages. After the program has correctly run include some errors either in the image names codes or the two data tables to see how the Critical errors and log files behave.

IMAGE NAMING RULES

Pyoinv runs two main processes. The first process asserts the quality of your image dataset names and its correspondence with the parameters added into the tables. Any error in the syntaxes will create a mismatch and raise a critical error. Common errors like missing letters in the image file names or missing structures or views will make the program halt its execution and refer the user to the log file CriticalErrors.txt. Once all errors have been fixed and the program is run again it will make the thumbnails images and the html pages.

Naming Rules and Image organization: These rules use the number and relative position of the characters inside the image file name to code the standard views metadata. All characters inside the image names are case sensitive. These data include:

1) A unique morphospecies or species code (10 characters maximum).
2) Specimen sex (**f**, **m** or **j**).
3) Type of image produced (**d** = drawing, **p** = light microscopy image, **s** = scanning electron microscope image, **c** = cleared structure view through light microscopy).
4) A three-letter code for the body part image. These codes must be equal as the ones included in the structureNomenclature.xlsx.
5) The orientation of the standard view, i.e., **d** = dorsal, **T** = posterior, etc. These codes must be the same as in the structureNomenclature.xlsx file.
6) A six-character code for the voucher specimen (TUXV058)
7) The magnification of the microscope used to begin with an underscore (_70X). This must be included if the digital camera does not automatically print the corresponding scale bar inside the image to later retrieve the information. If the camera does it, include a dummy code of three characters preceded by an underscore for all images (i.e. _00X).
8) A three characters code for the individual machine used to acquire the image preceded by an underscore (_Z10).
9) Images can have any of the following extensions: .jpg, .jpeg, .tiff, .png or .gif.

INVTUXV212fphablTXV058_70X_Z10.jpg

This name encodes the following metadata: (INVTUXV212) corresponding to the species *Globignatha sedgwicki* and all its associated data inside the SPPDATA.xlsx, (m) male specimen, (d) digital picture, of the complete body (hab) in lateral view (l) from the specimen voucher number TXV058 at 70X, taken with the workstation Z10 and it’s a jpg.

INSTRUCTIONS FOR WEBSITE CREATION

1) Run the main.py file. This will call the other python modules and guide you through the website creation process. The first time you run the program making the thumbnails can take

several minutes for datasets with thousands of images. To rename species or update other text data skip this process selecting no (n) in the command interface.

2) The program will stop if it finds inconsistencies between the image names and the data tables. Go to the CriticalErrors.txt to find the image names with errors and fix them. The log.txt file will include the errors found if the user modifies the python modules. Unexpected errors will be reported only through de command line interface of your OS.
3) The species webpages are identified by the species name. To rename misidentified species just change the names inside the SPPDATA.xlsx table. Run the program and look for the new name inside the HTML_files/Species folder, finally replace the corresponding family html page inside the Menus folder. If you added images or change their location the program must run again for the complete dataset to make the new thumbnails.
4) The structure of the species and menu pages can be modified in the respective html templates inside the *html_templates* folder. The inventory presentation page is the only that the user will make manually. This can be done on any program that edits html. An empty template is provided in the file welcome.html, however you can add any page and update its name inside the index file. The style of all pages is controlled by the file named styles.css inside the html_templates folder.
5) Transfer all the items inside the master folder into your hosting provider cpanel's File manager (inside your public_html). The website is called by the index.html file. If you need to change the name of the index file modify line 223 inside populateHtmls.py.

Thanks for using our protocol for Biodiversity Inventory Website construction!